Predictive Modeling for Sepsis: Leveraging SHAP Values

Gabriel Marin
Adviser: Mona Sighn

Abstract

This paper entails leveraging machine learning to aid in the issue of sepsis, the leading cause of death in hospitals. Due to its complexity, sepsis presents significant challenges in the medical field in terms of research, diagnosis, and treatment. This project utilizes a gradient boosting model through XGBoost and feature analysis through SHAP Values. SHAP Values let us visualize feature contribution in a model's prediction and can be further utilized to generate insights. These insights will inform medical professionals and researchers to allow for more informed decision-making and aid in future research.

1. Introduction

1.1. Motivation and Goal

If one had to guess the number one cause of death in hospitals, sepsis would not be the first thing that comes to mind. Sepsis is a life-threatening condition caused by the body's response to infection and it remains the number one cause of death in hospitals. When someone gets an infection the body's immune system releases chemicals to fight. Sepsis is the hijacking of the body's immune system resulting in inflammatory responses that can damage blood flow to organs. For the past three decades, significant research efforts have occurred, yet the sepsis fatality rate still ranges from 15%-30% per patient. Sepsis is problematic for several reasons:

1. The origin of Sepsis: Since sepsis can originate from any infection it becomes difficult to know when it will occur. The most common origins are pneumonia, urinary tract infections, or intra-abdominal infections such as appendicitis. [13]

2. Difficulty of early detection: There is no single test for sepsis as early symptoms can vary and make diagnosis difficult for healthcare workers.

3. Time-Sensitive Treatment: Treatment is most effective when detected early on and becomes less effective as time goes on. The longer that a patient goes undiagnosed, the more that the rate of fatality and length of treatment increases.

As mentioned, despite much progress made sepsis is still extremely dangerous. An example of this is the fact that the rate of fatality for sepsis has remained about the same since the year 2000 for adults aged 65 and over as shown in Figure 1.

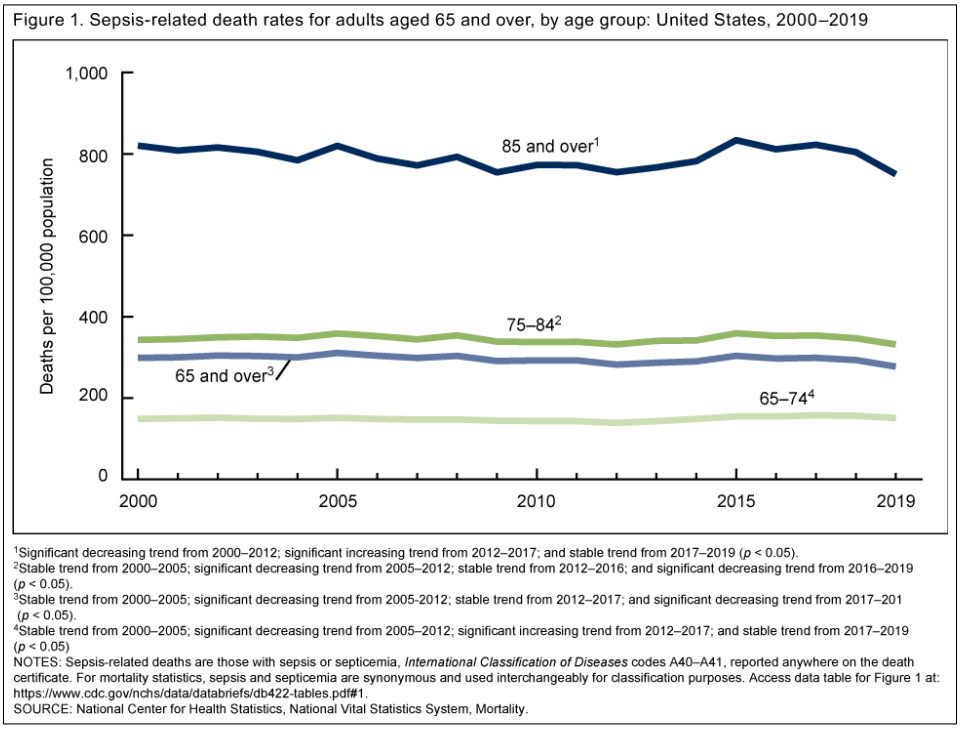

Figure 1: Sepsis-related death rates for adults aged 65 and over, by age group: United States, 2000–2019. (CDC 2019) [5]

With this in mind, this project seeks to leverage machine learning to deepen the understand of how features like vital signs, laboratory values, and demographics contribute to sepsis prediction models.

2. Background and Related Works

The severity of sepsis has sparked interest in developing predictive models to aid in early sepsis diagnosis and treatment. Various models have been utilized employing numerous algorithms

which each attempt to improve on predictive modeling for sepsis. In 2019, the Laboratory of MIT for Computational Physiology released the 2019 Physionet Challenge dataset [11]. This dataset provided large amounts of ICU patient records and encouraged participants to create models that could give early accurate predictions. There were many interesting works but the most notable of these is the work done by the winners of the challenge James Morrill and his team [1]. The route that James Morrill and his team took to win the competition was combining a gradient-boosting model with feature engineering. This feature engineering entailed creating features from signatures of paths of the time-series data in the dataset to supplement the model along with the original features. This innovative approach won the Physionet 2019 Challenge with a utility score of 0.360 and an AUROC score of 0.868.

One of the golden standards in sepsis predictive modeling is the InSight algorithm. This algorithm is similar to screening tests like SIRS, SOFA, and MEWS, which are all algorithms based on a small amount of information from a patient. These are all scoring systems that utilize no more than 9 vital signs and allow for some information when looking to diagnose sepsis. The Insight algorithm has been shown to outperform SIRS, SOFA, and MEWS by a significant amount [3]. It was developed by Jacob S. Calvert and his team from Dascena in 2016 and only makes use of nine commonly available vital signs [2]. The focus of the Insight algorithm is to maximize the performance of predictions made three hours before a patient's first SIRS episode, which is usually five hours in length. The performance outlined in the paper is amazing however there are a couple of issues that Scherpf Matthiue and his team elaborated on in their work. Matthieu claimed that the reclassification of sepsis in 2016 could "lead to delayed identification of health deterioration as the new definition of the term sepsis defines a more critical physiological status than before."[4]. In his lens, prior definitions of sepsis are more challenging to tackle in predictive modeling. This gap allows for improvement upon the Dascena Insight algorithm. He worked with the MIMIC III database with the goal of comparing performance to the insight algorithm [4]. Matthieu ended up outperforming the Dascena Insight algorithm by taking temporal developments into account with his recurrent neural network model. Beyond performance and metrics, this goes to show that

predictive modeling for sepsis is an ever-changing field due to the complexity of the disease. This project builds on these developments by focusing not only on performance but also on generating insights from the features themselves and the contribution they make towards a positive or negative sepsis prediction.

2.1. Approach

The objective set forth above will be completed through the utilization of SHAP Values, extreme gradient boosting, and a windowing method. All of these will be further elaborated in this work. The utilization of extreme gradient boosting models for classification tasks is nothing new, however in this case the results will be utilized for further feature analysis. SHAP Values allow us to visualize the contribution of each feature in a model's output and will be further analyzed to generate insights. Furthermore, a windowing method will be utilized to manipulate the dataset and observe the development of SHAP Values in different temporal settings. Overall, this approach seeks to unveil critical insights into how and why certain features influence the risk of sepsis that could have been missed in other works. Success in this project will be evaluated through performance of the model and alignment of SHAP Value analysis with current known medical research.

3. Implementation

3.1. Dataset

The dataset for this project comes from the PhysioNet/Computing in Cardiology Challenge 2019. The dataset holds information on 40,336 patients. The structure of this data set is hourly entries for each patient's entire stay at the ICU. A single row is a single hourly entry and it contains 41 columns that correspond to the features. There are three different feature types such as demographics, vital signs, and laboratory values. Table 1 displays some important features and what they mean, a full list of features can be found in Table 3 in the Appendix. 3

Table 1: Features and Names/Metric

Feature Symbol	Feature Name/Metric
HR	Heart rate (beats per minute)
Resp	Respiratory rate (breaths per minute)
Age	Years (100 for patients 90 or above) %
ICULOS	ICU length-of-stay (hours since ICU admission)
SepsisLabel	Sepsis Classification (1 for positive) [1ex]

The SepsisLabel feature is the feature used for classification. It is a 1 if a patient has sepsis and 0 if a patient does not have sepsis. However, there is something of importance to note with this feature. The hour in which a sepsis patient's SepsisLabel turns to a 1 is actually 6 hours before the true onset of sepsis. For example, if a patient's SepsisLabel turns to 1 at hour 6, that means that true sepsis onset occurs at hour 12. This was done by the organizers of the challenge dataset to encourage early predictions.

There are a total of 1,552,210 rows or hourly entries across all patients. Also, there is a large amount of missing data within this dataset. This is due to the fact that the dataset was created with data from three different hospital systems each with different equipment. Table 2 displays a few features and the percentages of total hourly entries in which they are missing. A full table can be found in Table 4 in the Appendix. 4

Table 2: Features and Percentage of Entries in which they are missing

Feature	Percentage of Entries Missing
pH	93%
Calcium	94.11%
Chloride	95.46%
Glucose	82.8%
Potassium	90.6%
Phosphate	95.9%
Magnesium	93.6%

The majority of the missing data are laboratory values and this aligns with the differing equipment at each hospital system. In total, there are 40,336 patients but only about 5.6% have sepsis. This means there is a pretty large imbalance in patients who have sepsis versus patients who do not have sepsis.

3.2. Data Preprocessing

The dataset takes the form of two folders in which each .psv file corresponds to a patient's stay in the ICU. Each row in a patient's .psv files corresponds to an hourly entry. These hourly entries contain information that was gathered at the hour by medical professionals. An example of this format can be seen in Figure 2:

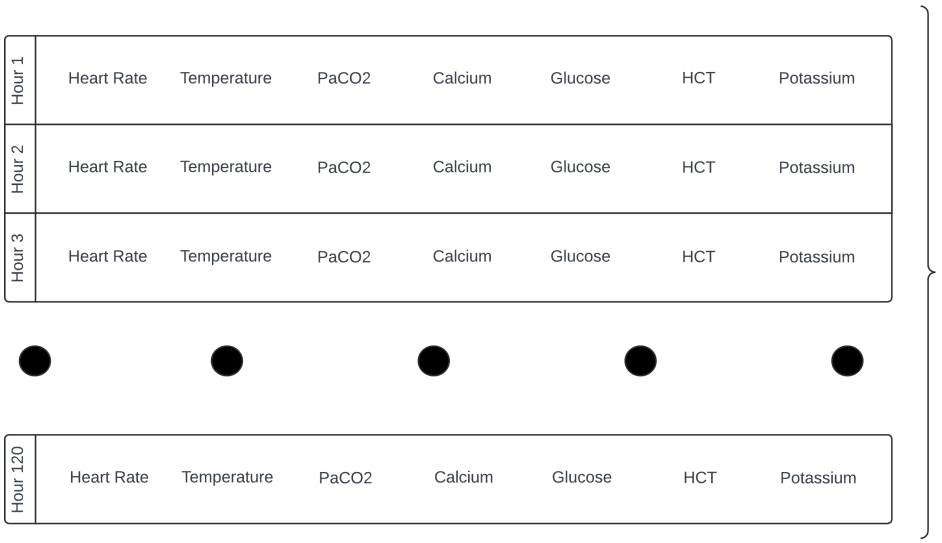

Figure 2: Example File for a Patient's stay at the ICU filled with hourly entries

Using the Pandas Python library I read each patient's file and appended each hourly entry into one Pandas data frame. The Pandas Python library is a library that allows for data analysis and manipulation in the Python language. A Pandas data frame is simply a two-dimensional structure of tabular data with labeled axes. This allows for easy use of the data in model construction.

3.2.1. Windowing Method In my analysis of feature importance, I wanted to explore how feature importance develops through temporal changes. More specifically, I wanted to explore how SHAP Values develop when the model has less contextual information. In this case, the contextual information is the hourly entries surrounding the hour at which SepsisLabel turns positive. Thus, I created a windowing method that allows me to limit the amount of data that the model receives. Through this method, I can analyze how changes in window size influence feature importance over time. Figure 3 displays the utilization of this windowing method on one patient.

Patient

Labels	Heart Rate	Temperature	PaCO2	Calcium	Glucose	HCT	Sepsis Label
Hour 1	97	36.22	35	8.6	70	41	0
Hour 2	106	36	36	9.1	80	46	0
Hour 3	102	36	36	10	85	41	0
Hour 4	100	36	36	10.1	100	41	0
Hour 5	99	36	35	11	96	41	0
Hour 6	97	36	35	9.6	87	46	1
Hour 7	105	36	36	9.6	77	47	1
Hour 8	101	36	37	11.1	98	49	1
Hour 9	100	36	38	8.7	86	477	1

Window Size: 4

Figure 3: Example of windowing method utilization with a window size of 4

In this example, we have a patient with a positive SepsisLabel at hour 6 and a window size of 4. The red row represents the hour at which SepsisLabel turns to 1. Through the windowing method, this patient's data will be cut down to the rows highlighted in red before being appended to the Pandas data frame. This means that the model will get 2 hours into the past and future in relation to the hour of onset. As a counter-example, with a window size of 12, the model would receive information 6 hours before and 6 hours after the hour of sepsis onset. The choice of the window size thus affects the construction of the model in terms of what features are deemed more important due to the varying amounts of data available to the model.

3.3. Model Selection

As shown previously, there is a lot of missing data in this dataset. At first, I wanted to work with an LSTM model as that made the most intuitive sense. An LSTM model, Long Short Term Memory,

is a model utilized often for time series datasets. However, after careful research through related works, it became clear that developing solely an LSTM model would be more difficult and would tend to perform the same if not worse than a gradient boosting model. The issue with an LSTM model is that missing data needs to be dealt with through interpolation or imputation, however with a large amount of missing data this can severely affect the model. With this in mind, I selected to use a gradient boosting model instead, specifically XGboost, because of the way that it handles missing data. A gradient-boosting model is an ensemble of weak prediction models called decision trees, in which each decision tree is a sequence of branches in which each branch aims to reduce uncertainty about the target variable. In XGBoost, a boosting round is a single iteration or a new tree added to the ensemble. XGBoost handles missing data differently than a usual gradient-boosting model. In other gradient boosting models missing data would be dealt with by creating a branch of missing and non-missing. XGBoost instead assigns a default direction of branches as it trains and when encountered with missing data simply assigns the default direction. This default direction is learned and improved upon when training, so for XGBoost missing data ends up being easy to handle. Figure 4 shows an example of how this functions.

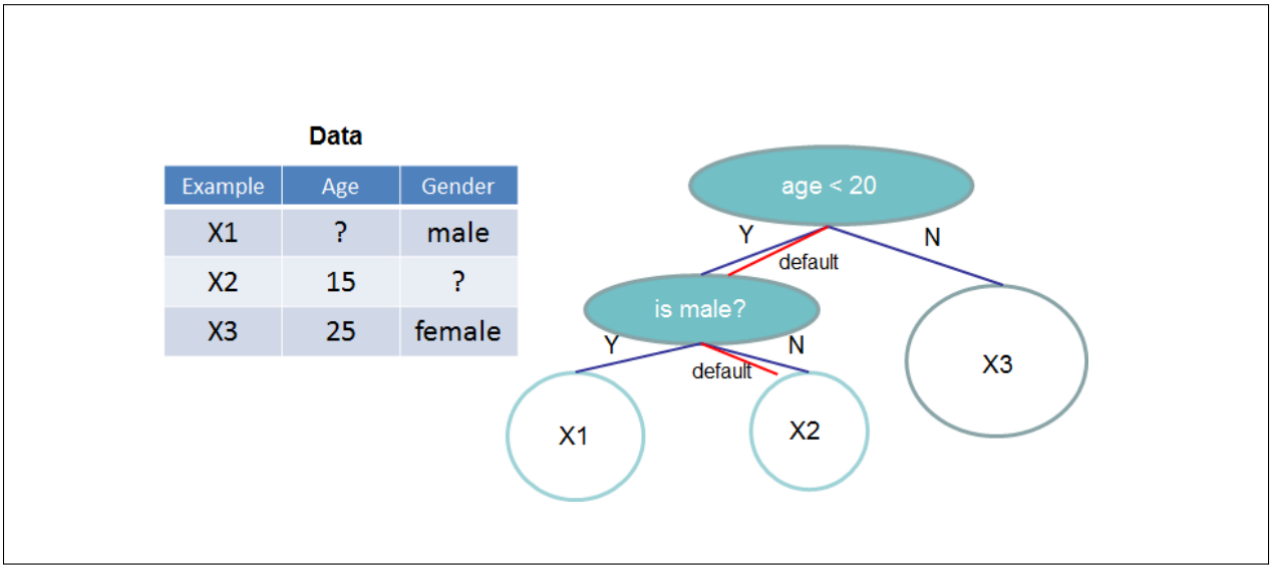

Figure 4: Exemplary use of default branch directions for XGBoost. An input with missing data will be classified into the default direction [12]

Thus, additional data processing in the form of interpolation or imputation was not necessary.

3.4. Framework

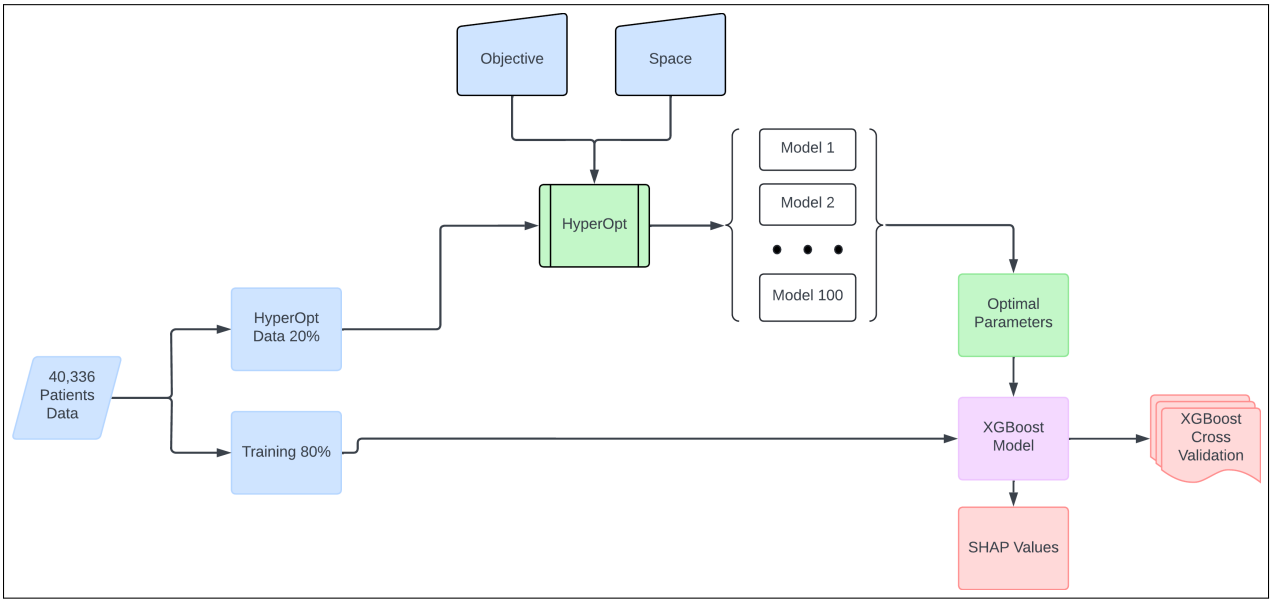

Figure 5: Architecture

Presented above in Figure 5 is an overview of the architecture for this entire process. At first, the data will be separated with a 20%/80% split for model training and parameter optimization. After the optimal parameters are found they will be used to build the model, and then the model will be trained over 500 boosting rounds with the 80% training data split. At the end, five-fold cross-validation will be performed to examine the performance of the model and SHAP Values will be generated to allow for feature analysis.

3.5. Parameter Optimization

In my project, the first step was to optimize the parameters for XGboost. There are three different kinds of parameters in XGBoost: task parameters, booster parameters, and general parameters. The focus of parameter optimization in this project was on booster parameters as these were the ones that could affect the performance of the gradient-boosting model. There are two other types of parameters like general parameters which select which kind of algorithm to use and task parameters which are tied to things like metrics. I utilized the HyperOpt Python library for optimizing these parameters. HyperOpt is a library that is designed for optimizing hyperparameters when the search

space is complex. It functions by employing random search, tree of parzen estimators, and adaptive TPE. The specifics of these algorithms are further elaborated in HyperOpt documentation[14]. HyperOpt requires a search space definition which essentially outlines which hyperparameters should be optimized and for what ranges. Beyond this, it requires an objective function that lets HyperOpt know how well a set of hyperparameters performs. In this case, this was performed with 20% of the data that was separated at the beginning of the process as shown in Figure 5.

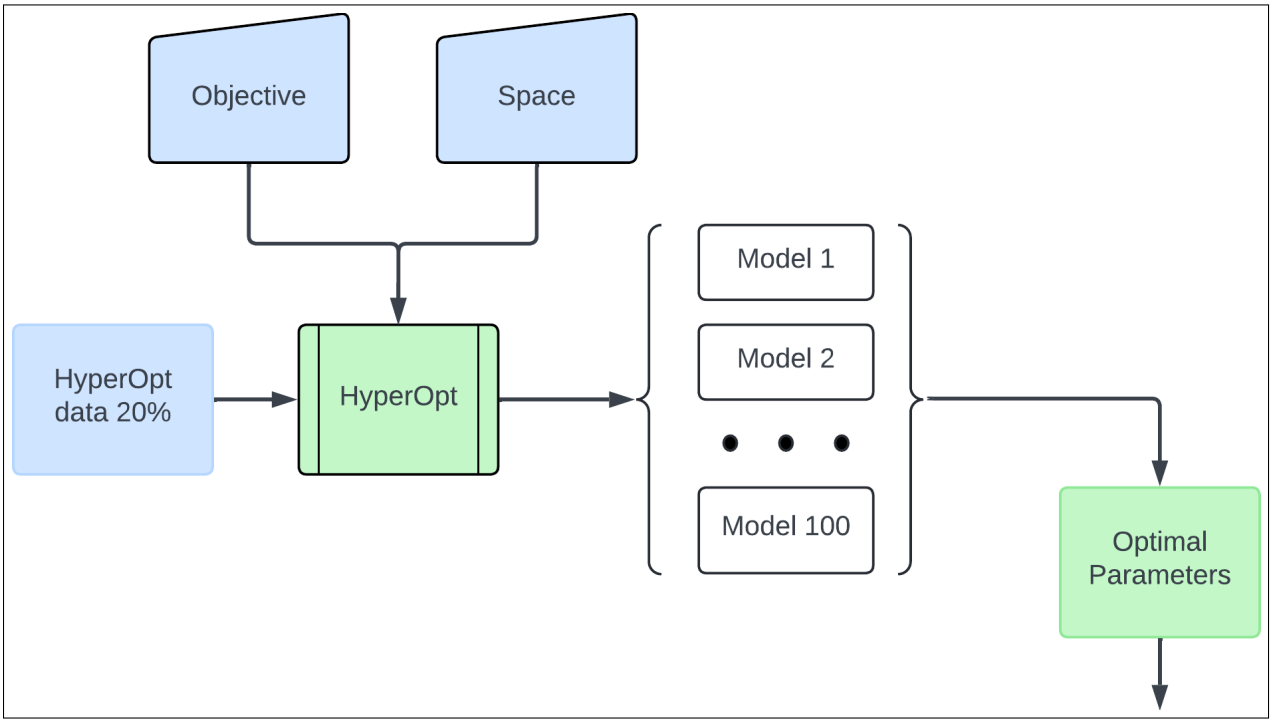

Figure 6: Steps and Components of Parameter Optimization with HyperOpt

As for the objective function, I returned the performance of the XGBoost model with 500 boosting rounds on 5 different folds of the HyperOpt data. Beyond this, the search function I created optimized the following hyperparameters: max depth, gamma, eta, min child weight, colsample by tree, colsample by level, scale pos weight, subsample, lambda, and alpha. The ranges for which these parameters were optimized can be found in Table 5 in the Appendix. 5 Within this set of parameters, scale pos weight is an important one to highlight as it handles imbalances in classes which is very prevalent in this dataset. In an ideal world, we could optimize all of the hyperparameters for large ranges, however, this takes lengthy amounts of time. With the parameters

that are currently outlined and making use of an RTX 2060 Nvidia GPU through Nvidia's Cuda toolkit, this takes approximately 30 minutes.

3.6. Cross Fold Validation

The next step in my implementation was to use cross-fold validation for the evaluation of the optimized parameters. I utilized the XGBoost cross-validation function which executes cross-fold validation with 500 boosting rounds across 5 folds. Here we can see the results of the cross-fold validation and this was done without utilization of the windowing method. The way to read this is that the x-axis is the boosting round that we are at and the y-axis is the average AUROC score across the five different folds. So at boosting round 200, the y-value gives us the average AUROC for all 5 folds. We look towards the end to see the average AUROC score across the 5 folds once we are at the final boosting round.

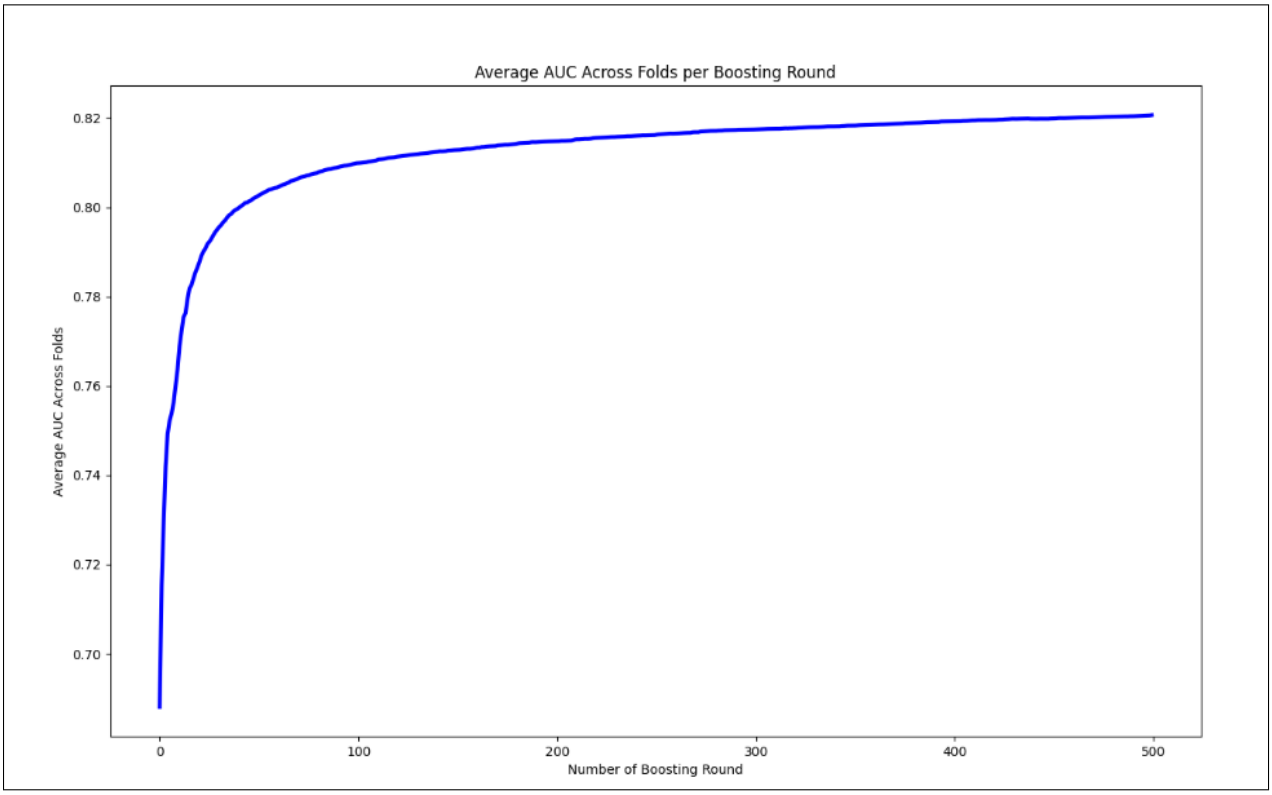

Figure 7: Average AUROC Scores over 5 folds and 500 boosting rounds. The X-axis is the number of boosting round. Y-axis is the average AUROC score over 5 folds.

3.7. SHAP Values

I utilized the SHAP library for feature analysis within this project. The SHAP Library is a Python library that allows for the interpretation of feature contributions through a game-theoretic approach. The library is based on the concept of SHAP (SHapley Additive exPlanations) Values. For each prediction made for a model, each feature acts like a player in a game where the prediction is the payout. SHAP Values essentially describe how to distribute this payout among the features. Each SHAP Value is calculated by examining the change that occurs in the prediction when a feature is extracted and then added back to the model. [15] The exact computation of these SHAP Values entails multiple algorithms to approximate these values. A SHAP Value chart is shown in Figure 8 which was generated through the process described in this section.

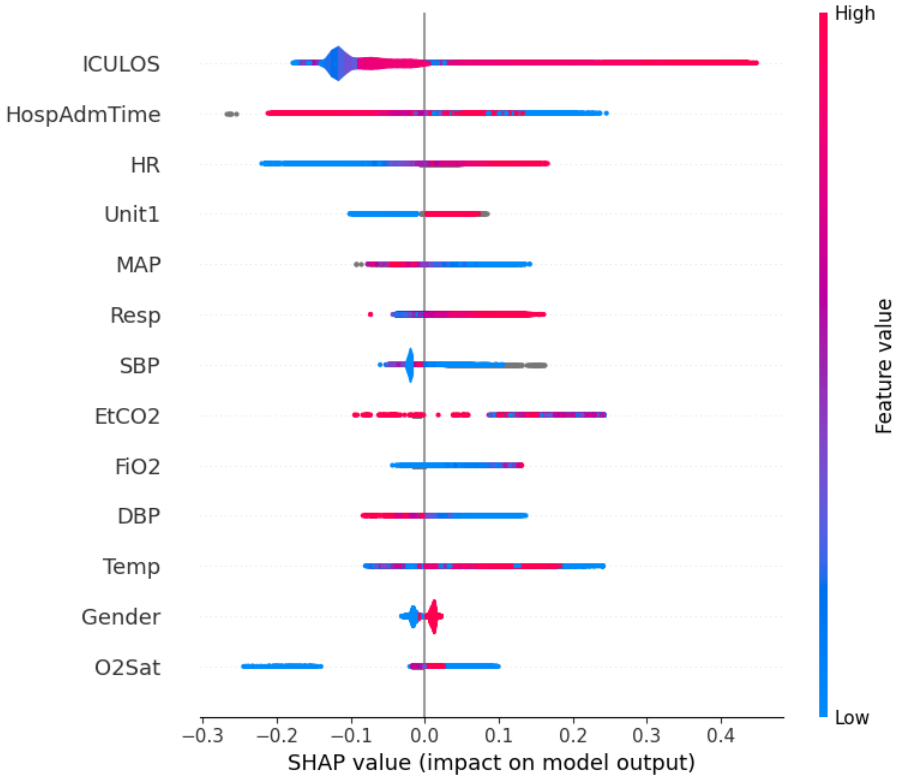

Figure 8: SHAP Value Summary Chart ordered highest importance at the top determined by the absolute sum of SHAP Values

All the SHAP Values are organized from order of highest importance to lowest importance. In this case, importance does not mean a positive or negative contribution instead it is a sum of absolute

values. A single SHAP value or dot on the chart represents an instance, or hourly entry, in which that feature had a contribution. The range of color represents high or low feature values while the x-axis represents the impact of the feature on the prediction. In my case a prediction of sepsis is represented by the value 1 so SHAP Values on the positive side of the axis contribute towards a positive sepsis prediction and the opposite is true for SHAP Values on the negative side of the x-axis. On this chart, we see that ICU Length of Stay (ICULOS) is the feature with the most impact on the model. Another example is the MAP feature, mean arterial pressure. We see a cluster of red SHAP values near the negative side which means that the model associates a higher MAP Value with a decreased likelihood of sepsis. However, it's important to take into account that a patient could come in with high or low blood pressure and this could be present in the dataset.

4. Results

4.1. AUROC Score

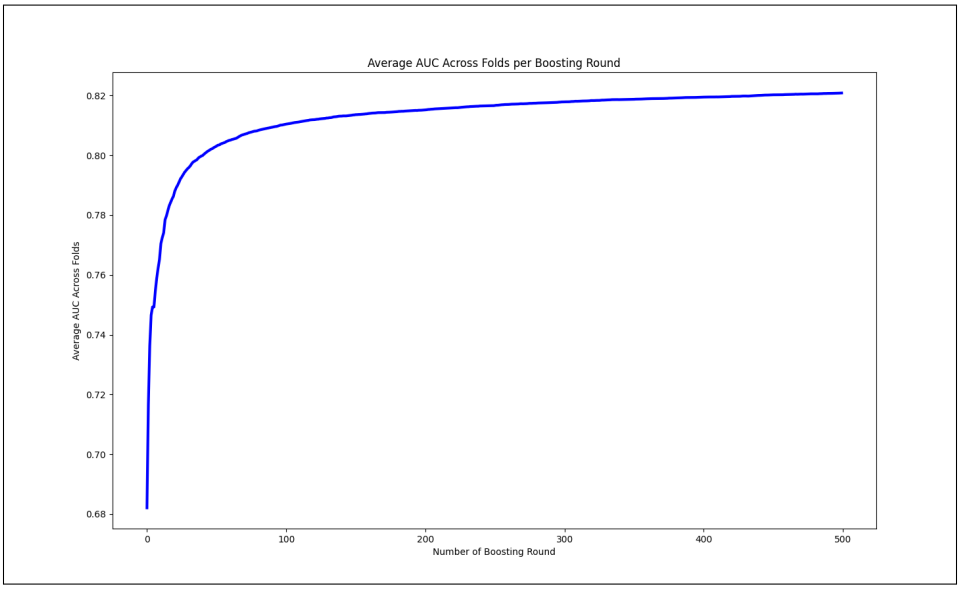

Figure 9: Average AUROC Scores over 5 folds and 500 boosting rounds. The X-axis is the number of boosting round. Y-axis is the average AUROC score over 5 folds.

Through this process, cross-fold validation was utilized over five folds on the remaining 80% of the data reserved after separating a portion for parameter optimization. An AUROC score of 0.5

corresponds to random guessing and any score about 0.8 is considered good. The learning curves stabilize as boosting rounds increase, converging to an average AUROC score of 0.82 at the final boosting round. This indicates good predictive ability as well as generalization of the model across different subsets of the data.

4.2. SHAP Values

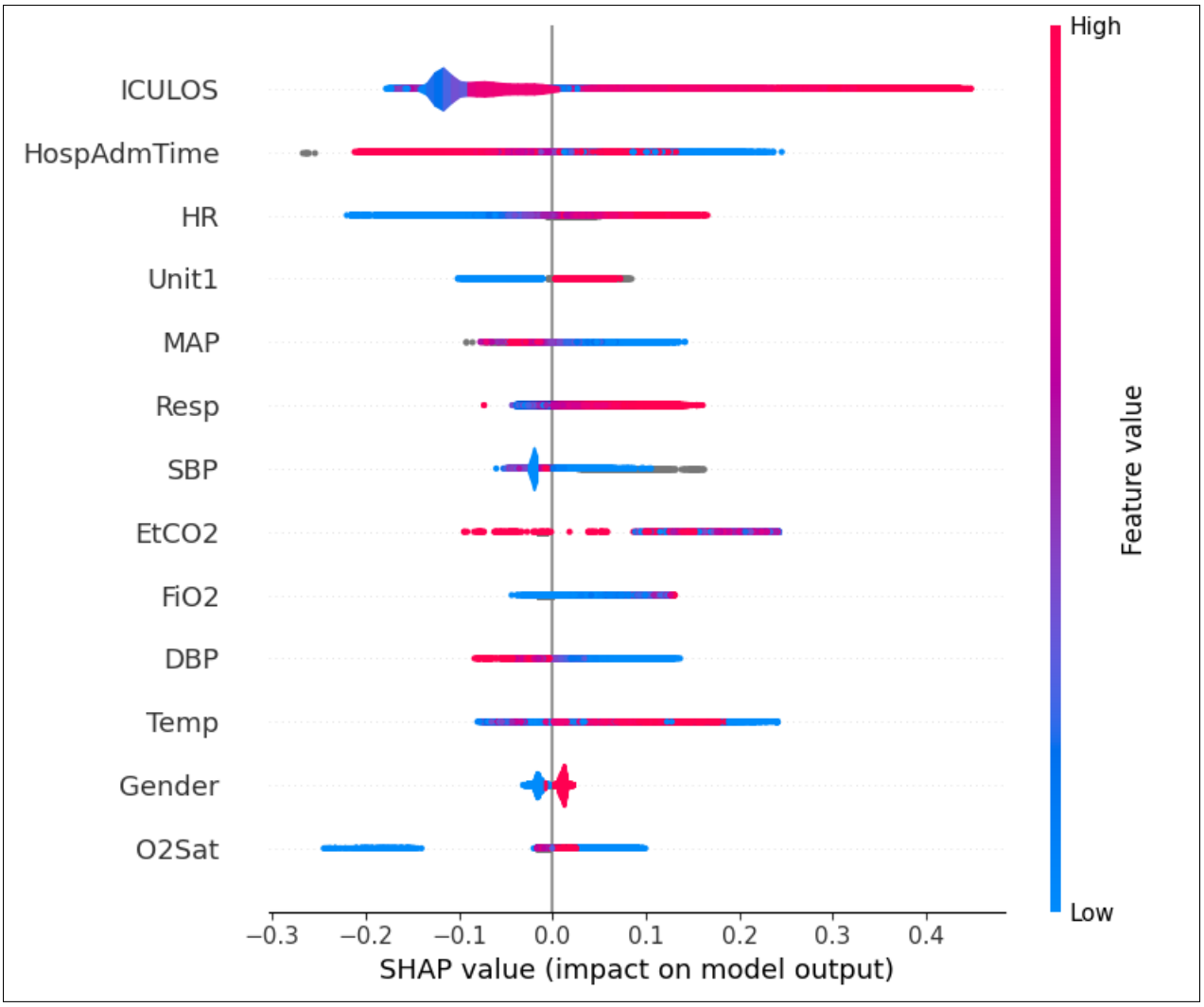

Figure 10: SHAP Value Summary Chart ordered highest importance at the top determined by the absolute sum of SHAP Values

The SHAP values for the model reveal insights into sepsis predictive modeling. ICU Length of Stay is the number one feature with the most contribution to the model along with HospAdmTime.

Notably, the ICULOS's SHAP value distribution includes a significant cluster of low (blue) values. This correlates with the clinical understanding that patients with sepsis often have prolonged ICU stays [6]. At the same time, a patient without sepsis is likely to be discharged at an earlier time, hence the cluster of shorter stays contributing towards a negative sepsis prediction. Furthermore, all vital signs are represented in the SHAP value chart, indicating they possess a greater contribution to the model than a lot of the laboratory values. Most distributions are narrow and exhibit a rightward shift, suggesting that they consistently contribute to a positive sepsis prediction. In terms of how these findings align with current sepsis medical research, several correlations can be observed. The SIRS (Systemic Inflammatory Response Syndrome) criteria, commonly used for sepsis identification and diagnosis, encompass variables like temperature, heart rate, respiratory rate, and white blood cell counts[8]. Most of these criteria are reflected in the SHAP values specifically through HR, Resp, and Temperature features.

The golden standard for sepsis management is the international guidelines for the management of sepsis and sepsis shock. The most recent guidelines from 2021 disregard qSOFA as a screening tool for sepsis since it is neither sensitive nor specific for sepsis[9]. qSOFA stands for quick SOFA score and is composed of altered mental state, respiratory rate, and systolic blood pressure. Despite qSOFA's limitation outlined by these guidelines, my SHAP value analysis indicates that two of these, respiratory rate and SBP (Systolic Blood Pressure), still have a great impact on the model's predictions.

SBP displays a complex distribution both in shape and feature values. This can be explained by SBP's relationship with sepsis. SBP can be influenced by many factors such as underlying conditions and medications which are not disclosed in the dataset. Furthermore, individuals respond differently to infection with some having a drop in SBP and some having a normal SBP. This just further underscores the complexity of sepsis and the challenges in its diagnosis and treatment.

I have generated three additional SHAP value graphs with varying window sizes to analyze how the SHAP Values evolve as the model receives more contextual information.

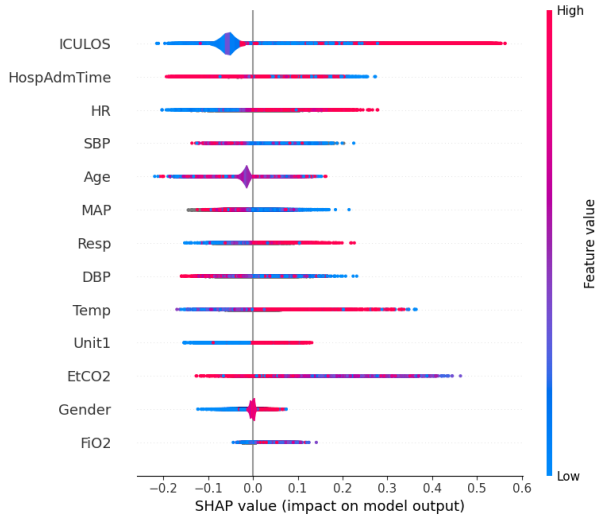

Figure 11: (A) SHAP Values with a window size of 4

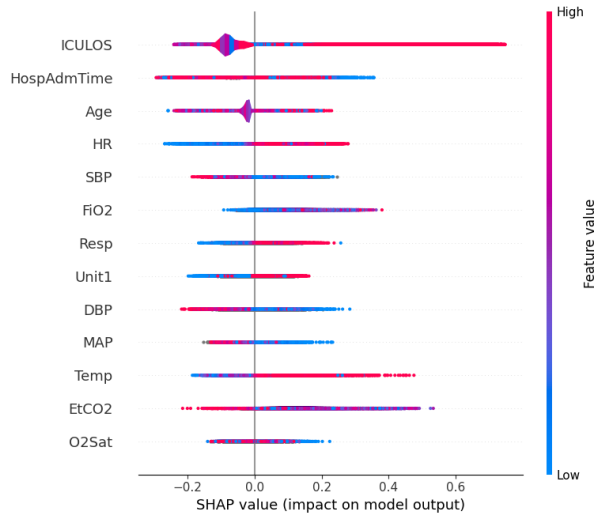

Figure 12: (B) SHAP Values with a window size of 8

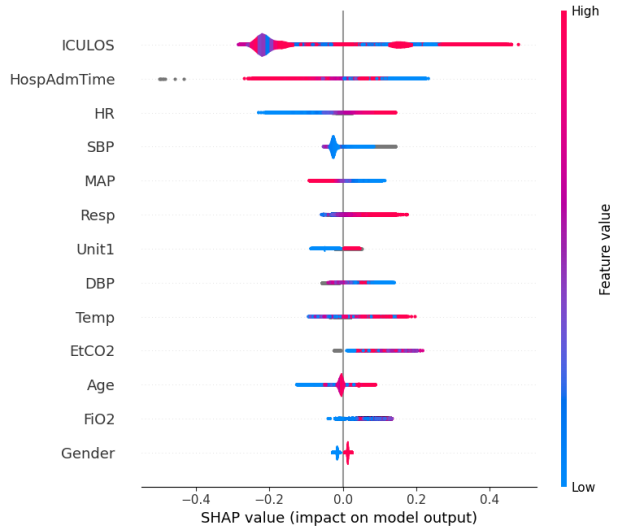

Figure 13: (C) SHAP Values with a window size of 16

These three different figures show us SHAP Values generated using multiple window sizes. Throughout all of the different window sizes, ICU Length of Stay and HopsAdmTime reign as the features with the most overall impact. As we can see in the change from Figure A to B to C, the distribution of ICU length of stay gets narrower as the window size gets larger. This points to a more uniform impact on the model as more contextual information is available. This is also a trend in which distributions tend to be more narrow or consistent with larger window sizes. Overall we can spot the trend of a rightward shift in the distributions as window size increases which means

that features contribute more to a negative sepsis prediction. Age, initially a strong predictor, sees its influence wane in larger windows reaffirming the complex age-transcending nature of sepsis progression. Initially, age can be more useful in predictive modeling as age differences in terms of sepsis severity have a more distinct difference, but as sepsis progresses, this distinction is less clear. There are a number of features that maintain a pretty even division in colors. For example, Gender and Temperature display a distinct division: lower values tend to reduce the likelihood of sepsis, whereas higher values increase it. As mentioned before, SIRS criteria are a vital part of sepsis predictive modeling and we can see that throughout different window sizes. Three of these criteria: heart rate, respiratory rate, and temperature remain about the same in terms of total contribution. Furthermore, we can see in the change of distribution for SBP from Figures A and B to C that it presents as a reliable early-stage indicator, yet its predictive reliability shifts as treatment and physiological adaptations come into play. Figure C brings Gender into sharper focus with a very narrow distribution. There is a clear distinction between males and females. Males are presented in the dataset as 1 and females as 0. This aligns with studies that have demonstrated that men are more at risk of sepsis than women [7]. From Figure B to C we can see that Fi02 (Fraction of inspired oxygen %) drops in contribution and this can be attributed to concerns about oxygen toxicity when patients receive high levels of oxygen for a prolonged period of time [10]. In Figure C we can see that the distribution for Unit1 becomes very narrow as opposed to shorter window frames. Unit1, which refers to a patient staying in the MICU, has its total contribution increase. The MICU is the Medical Intensive Care Unit which houses patients who are usually in need of stability but whose state is not as critical as patients in the ICU. It seems unintuitive that a patient who is at the MICU is more likely to have sepsis according to the model, however, it's important to take note that sepsis has different levels of severity. The patient care pathways and practices across different units require deeper understanding, especially for sepsis which can have a wide range of severities and affect patients with preexisting conditions differently.

Overall, analysis of SHAP value graphs across different window sizes reveals consistent trends. ICU Length of Stay remains the feature with the most contribution. As we increase time frames

distributions get narrower and shift to the right, displaying an increase in a positive sepsis prediction across all features. The impact of vital signs and respiratory markers is more pronounced in shorter window frames but tends to diminish as window sizes expand. Similarly, the influence of certain demographic features becomes less pronounced with larger window sizes aligning with existing medical research.

5. Conclusion

5.0.1. Strengths and Contributions The contributions of this project are multifaceted. The majority of the contributions come from the application of SHAP values to clarify the influence of various clinical features in the model. This is taken further through the exploration of SHAP values across different window sizes which provide deeper insights into feature importance through temporal variations. These insights are a stepping stone towards developing more early detection strategies for sepsis. The overall process developed in this project allows for easy predictive model training and parameter optimization with different window sizes. Each optimization is tailored to the selected window size ensuring that the model dynamically adjusts to the nuances introduced by varying amounts of contextual patient information. Within predictive modeling for medical purposes, this is a very valuable property as often medical datasets are far from ideal. This integration of model training, parameter optimization, windowing method, and SHAP value exploration summarizes the contributions of this project.

5.0.2. Limitations and Future Work Despite its strengths, there are some limitations to this project. The performance of the model created in this project is not as good compared to the winners of the Physionet 2019 Challenge. The nature of this dataset is also a limitation as there is an imbalance of patients with sepsis and without sepsis. While this is tackled in parameter optimization, it would be best to work with multiple datasets in future work. There are also many interactions between features that take tremendous amounts of time to fully understand and grasp.

As for future work, I would like to explore more SHAP Values that were created in this project. I was only able to highlight about thirteen features in this project as exploring forty-two features is too

lengthy for this paper. Beyond that, I would like to explore SHAP Values utilizing my windowing methods on different datasets and with different models. This could include integrating ensemble methods or advanced neural network models. Beyond exploring more individual datasets it would also be of interest to append many datasets together and examine SHAP Values there.

Furthermore, others could adopt the windowing method and process to investigate SHAP Values on their own. Medical researchers could also make sure of insights highlighted through this project and conduct further medical research/experimentation.

6. Acknowledgement

I would like to thank Professor Mona Singh and Professor Quang Minh Hoang for their help. This project would not have been possible without their guidance. Beyond this, I would like to thank my classmates as their feedback was very helpful in the creation of this project.

7. Honor Code

Honor Code This paper represents my own work in accordance with university regulations. /s/ Gabriel Marin

References

[1] Morrill, James H. MMath1,2; Kormilitzin, Andrey PhD1,3; Nevado-Holgado, Alejo J. PhD3; Swaminathan, Sumanth PhD2,4; Howison, Samuel D. DPhil1; Lyons, Terry J. DPhil1. Utilization of the Signature Method to Identify the Early Onset of Sepsis From Multivariate Physiological Time Series in Critical Care Monitoring. Critical Care Medicine 48(10):p e976-e981, October 2020. | DOI: 10.1097/CCM.0000000000004510

[2] Calvert, Jacob S., et al. "A computational approach to early sepsis detection." Computers in biology and medicine 74 (2016): 69-73.

[3] Mao, Q., et al. "Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and ICU. BMJ Open. 2018; 8 (1): e017833." Shock 44.5 (2015): 390-396.

[4] Scherpf, Matthieu, et al. "Predicting sepsis with a recurrent neural network using the MIMIC III database." Computers in biology and medicine 113 (2019): 103395.

[5] Kramarow, Ellen A. "Sepsis-related mortality among adults aged 65 and over: United States, 2019." (2021).

[6] Sakr, Yasser, et al. "Sepsis in intensive care unit patients: worldwide data from the intensive care over nations audit." Open forum infectious diseases. Vol. 5. No. 12. US: Oxford University Press, 2018.

[7] Lakbar, Ines, et al. "Interactions between gender and sepsis—implications for the future." Microorganisms 11.3 (2023): 746.

[8] Gauer, Robert, Damon Forbes, and Nathan Boyer. "Sepsis: diagnosis and management." American family physician 101.7 (2020): 409-418.

[9] Prescott, Hallie C., and Marlies Ostermann. "What is new and different in the 2021 Surviving Sepsis Campaign guidelines." Medizinische Klinik-Intensivmedizin und Notfallmedizin 118.Suppl 2 (2023): 75-79.

[10] Pannu, Sonal R., et al. "Development and validation of severe hypoxemia associated risk

prediction model in 1,000 mechanically ventilated patients." Critical care medicine 43.2 (2015): 308-317.

[11] Reyna, Matthew A. PhD1; Josef, Christopher S. MD1; Jeter, Russell PhD1; Shashikumar, Supreeth P. B.Tech2,3; Westover, M. Brandon MD, PhD4; Nemati, Shamim PhD1,3; Clifford, Gari D. DPhil1,2; Sharma, Ashish PhD1. Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. Critical Care Medicine 48(2):p 210-217, February 2020. | DOI: 10.1097/CCM.0000000000004145

[12] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[13]Prescott, Hallie. "What Are Septic Shock and Sepsis? The Facts behind These Deadly Conditions." Michigan Medicine, 7 June 2016, www.michiganmedicine.org/health-lab/what-are-septic-shock-and-sepsis-facts-behind-these-deadly-conditions.

[14]"Hyperopt: Distributed Asynchronous Hyper-Parameter Optimization." Hyperopt Documentation, hyperopt.github.io/hyperopt/#algorithms. Accessed 28 Apr. 2024.

[15] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Advances in neural information processing systems 30 (2017).

8. Appendix

Table 3: List of Features and Names/Metrics

Feature Symbol	Feature Name/Metric
HR	Heart rate (beats per minute)
O2Sat	Pulse oximetry (%)
Temp	Temperature (Deg C)
SBP	Systolic BP (mm Hg)
MAP	Mean arterial pressure (mm Hg)
DBP	Diastolic BP (mm Hg)
Resp	Respiration rate (breaths per minute)
EtCO2	End tidal carbon dioxide (mm Hg)
BaseExcess	Measure of excess bicarbonate (mmol/L)
HCO3	Bicarbonate (mmol/L)
FiO2	Fraction of inspired oxygen (%)
pH	N/A
PaCO2	Partial pressure of carbon dioxide from arterial blood (mm Hg)
SaO2	Oxygen saturation from arterial blood (%)
AST	Aspartate transaminase (IU/L)
BUN	Blood urea nitrogen (mg/dL)
Alkalinephos	Alkaline phosphatase (IU/L)
Calcium	(mg/dL)
Chloride	(mmol/L)
Creatinine	(mg/dL)
Bilirubin_direct	Bilirubin direct (mg/dL)
Glucose	Serum glucose (mg/dL)
Lactate	Lactic acid (mg/dL)
Magnesium	(mmol/dL)
Phosphate	(mg/dL)
Potassium	(mmol/L)
Bilirubin_total	Total bilirubin (mg/dL)
TroponinI	Troponin I (ng/mL)
Hct	Hematocrit (%)
Hgb	Hemoglobin (g/dL)
PTT	partial thromboplastin time (seconds)
WBC	Leukocyte count (count 10^3/μL)
Fibrinogen	(mg/dL)
Platelets	(count 10^3μL)

Table 4: Full List of Features and Percentages of entries missing

Feature	Percentage
pH	93%
Calcium	94.11%
Chloride	95.46%
Glucose	82.8%
Magnesium	93.6%
HR	9.8%
O2Sat	13%
Temp	66.16%
SBP	14.5%
MAP	12.45%
DBP	31.3%
Resp	15.3%
EtCO2	96.2%
BaseExcess	94.5%
HCO3	95.8%
FiO2	91.66%
PaCO2	94.4%
SaO2	96.5%
AST	98.3%
BUN	93.13%
Alkalinephos	98.39%
Creatinine	93.9%
Bilirubin_direct	99.8%
Lactate	97.3%
Phosphate	95.9%
Potassium	90.6%
Bilirubin_total	98.5%
TroponinI	99%
Hct	91.1%
Hgb	92.6%
PTT	97%
WBC	93.5%
Fibrinogen	99.3%
Platelets	94.0%
Age	0%
Gender	0%
Unit1	0.39%
Unit2	0.39%
HospAdmTime	0.00005%
ICULOS	0%
SepsisLabel	0%

Table 5: Full List of Features and Percentages of entries missing

Parameter	Range
max_depth	4-8
gamma	0-2
eta	0.1-0.35
min_child_weight	0-2
colsample_bytree	0.5-1
colsample_by_level	0.7-1
scale_post_weight	30-70
subsample	0.5- 1
lambda	1-3
alpha	0 - 1