# EARLY PREDICTION OF SEPSIS FROM CLINICAL DATA

Ofri Hefetz 209028067, Shai Shani 206165318

**Repository Link: https://github.com/OfriHefetz/Sepsis_Prediction**

## 1. EXECUTIVE SUMMARY


- For better patient outcomes, early diagnosis of sepsis in intensive care units is required. We constructed a machine learning model in this study to predict whether a patient in intensive care will develop sepsis around 6 hours before being diagnosed with sepsis.
- We analyzed the dataset using exploratory data analysis, which examined the distribution of attributes based on whether a patient had sepsis. It distinguishes between sepsis and non-sepsis patients based on vital signs, laboratory data, and specific demographic characteristics.
- The treatment of missing data is reviewed, emphasizing that most features, particularly in laboratory measurements, include missing values. Missing numbers suggest some tests were not performed on certain patients. It is essential to handle missing data effectively since it can affect the prediction model's performance and accuracy. We suggest a binary column indicating if a value is missing to manage missing values during modeling since imputation methods are seen as insufficient for missing data.
- Our analysis discovered that one technique of the 'aggregate' dataset performed better, therefore we focused our efforts on that dataset. To find the best threshold point, we examined several levels. Our final model will be used to predict sepsis in intensive care unit patients.
- Our findings demonstrate that EDA, feature engineering, and prediction algorithms are effective at comprehending complex medical datasets and producing accurate predictions to enhance patient outcomes. This study provides evidence for sepsis detection in intensive care units and has the potential to enhance patient treatment and outcomes.


**Keywords**: Sepsis prediction, Intensive care, Exploratory data analysis (EDA), Feature engineering, Hyperparameter optimization, XGBoost.

## 2. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a crucial step in any project. In this section, we will explore the various features available in the dataset and describe them in detail. We will also conduct a comparative analysis between the features, inspect their distribution using plots and perform hypothesis testing to determine if any relationships or correlations exist. Furthermore, we will discuss how to handle missing data to ensure that our analysis is accurate and robust.

### 2.1 Features Analysis

The dataset contains information about patients in intensive care units at two hospitals in the US and their vital signs, laboratory values, demographics, and whether or not they have sepsis.

Each row represents data collected for one hour, sorted according to the hours. The first row represents the first hour of the patient's arrival, and the last row represents when the patient left the intensive care unit.

The data consists of 754817 records and 42 features (20000 patients). The first step in the exploration of

our data we examine the structure and quality of the data. Most of the data types of the features are float64, which means they are continuous variables, except for the features of gender, Unit1, Unit2, and the target isSepsis. It is important to note that no duplicates were found in the data and that each row is unique.

Furthermore, the dataset can be divided into 4 main subjects, including the following features:

1. Vital signs (columns 1-8)
2. Laboratory values (columns 9-34)
3. Demographics (columns 35-40)
4. Outcome (column 41)

The exploratory data analysis was performed solely on the training set.

It is important to remember that in the training dataset, we have 20,000 patients, while in the testing dataset, we have 10,000 patients.

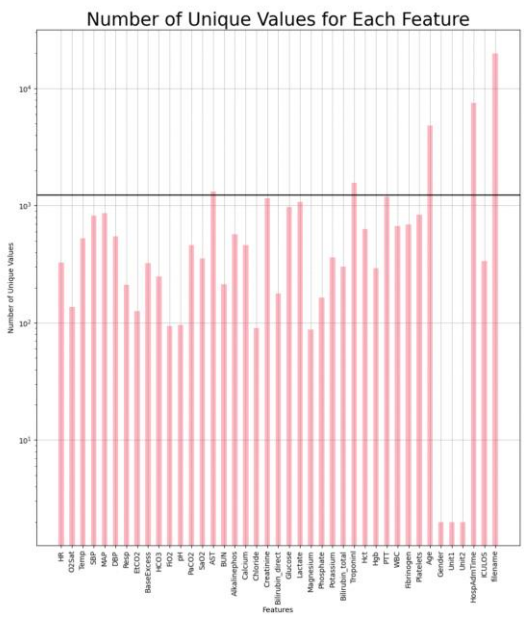



Figure 1: Number of Unique Values for Each Feature

Furthermore, as shown in Figure [1], the HR feature has 326 distinct values, while the O2Sat feature has 137. This shows that there may be a range of these characteristics.

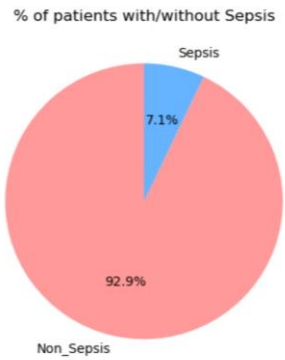



Figure 2: Percentage of Patients with/ without Sepsis

The analysis found that the dataset has an imbalance regarding the number of sepsis patients, reflecting sepsis's incidence in real-world ICU settings. Sepsis is a prevalent and dangerous illness affecting many intensive care patients. It is a considerable proportion of intensive care patients, and it is the top cause of death in ICU patients in the United States who do not have coronary artery thrombosis. Sepsis affects 1-2% of all hospitalizations and accounts for around 25% of ICU beds. [1] (Source: Wikipedia, https://en.wikipedia.org/wiki/Sepsis)

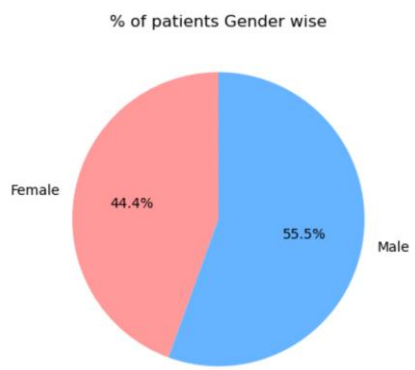



Figure 3: Percentage of Patients Gender Wise

The results also show that male patients are likelier than female patients to develop and die from sepsis. 55.5% of male and 44.4% of female patients were diagnosed with sepsis during an ICU stay.

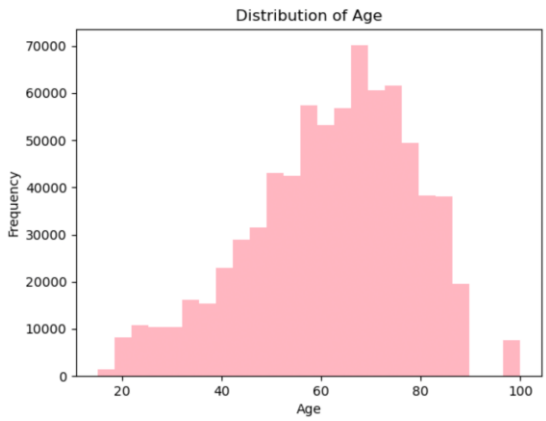



Figure 4: Distribution of Age

The patient's age distribution looks typical when the peak occurs between 60 and 80. Noticeably, there is a shortage of data for patients aged 90 to 100.

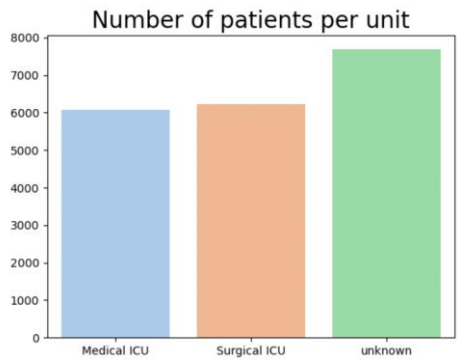



Figure 5: Number of Patients per Unit

Furthermore, the graph shows no statistically significant difference in sepsis patients between departments. The differences between them are insignificant. Thus, we will not discuss them further.

In addition, it can be seen from the graph that there is no significant difference between the patients with sepsis in the different departments. That is to say, the differences between them are negligible, so we will not refer to them later.

### 2.2 Features Comparative Analysis

We will now look at each feature's distribution given the distribution of whether a patient has sepsis (Appendix A). To better understand the data, we will check the distribution according to the groups we presented earlier:

1. <u>Vital signs features:</u>
   a. HR, Temp, Resp, and EtCO2 differ in whether a patient has sepsis.
   b. The rest of the attributes do not differ; therefore, we might consider them irrelevant to sepsis.
2. <u>Laboratory features:</u>
   a. Despite having a similar mean, the features BaseExcess, pH, BUN, HCO3, AST, Calcium, Bilirubin_direct, PTT, Chloride, WBC, and Creatinine appear to have a higher mean for septic patients.
   b. We should remember that despite Bilirubin_direct looking higher for septic patients, more than 96% of the patients are missing this attribute. In addition, although Bilirubin_total is higher, it is not worth a lot because it is defined as the sum of Bilirubin_direct and indirect bilirubin.
   c. Hct, Platelets, and Hgb have slightly lower means for septic patients.
3. <u>Demographics features:</u>
   a. HospAdmTime, ICULOS, and Age differ in whether a patient has sepsis.
   b. The rest of the attributes do not differ; therefore, we might consider them irrelevant to sepsis.

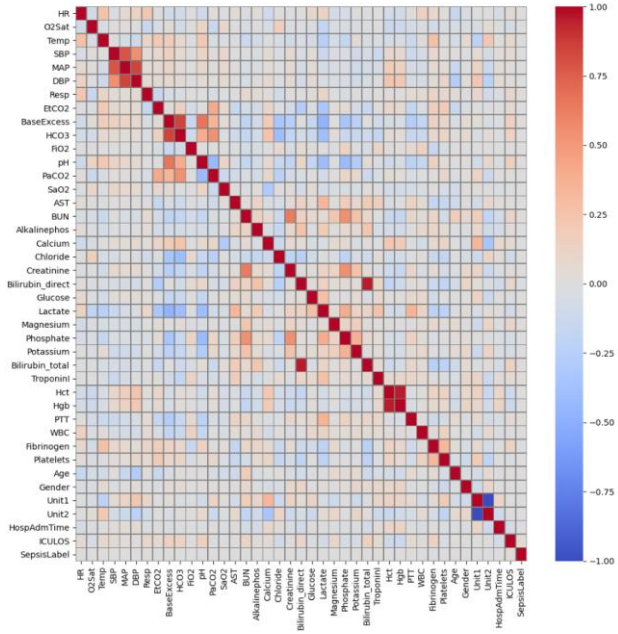



Figure 6: Features Correlation Heat Map

To better understand the relationships between the features, we used a visualization of a correlation heat map between each feature and feature category, as shown in figure [6]. The correlation heat map shows the correlation coefficients between all pairs of features. For example, the correlation heat map matrix shows that the SBP, MAP, and DBP features are highly correlated, which is expected as they are all related to blood pressure. Likewise, the HCO3, pH, and PaCO2 positively correlate with the BaseExcess feature, which is expected as they are both related to acid-base balance. Also, we can see that the Bilirubin_direct and Bilirubin_total are, as expected, positively correlated with 0.956440, as one indicated the total of the other.

Hgb and the Hct also have a high correlation with 0.953687, which is expected as they are related to red blood cells.

Here are some of our findings of the feature correlation based on each category:

1. <u>Vital Signs Features:</u>

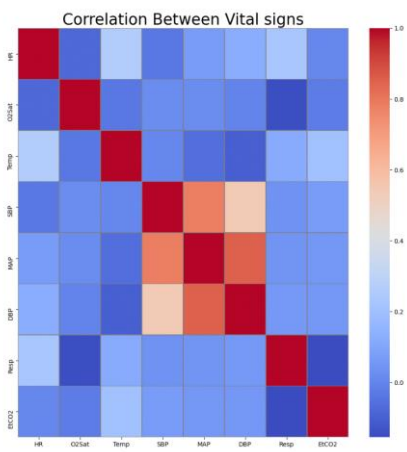



Figure 7: Vital Signs Features Correlation Heat Map

We observed that most of the features in the category are negatively correlated except for the center of the heat map, which contains the SBP, MAP, and DBP we mentioned earlier. This is unexpected and may warrant further investigation.

2. <u>Laboratory Features:</u>

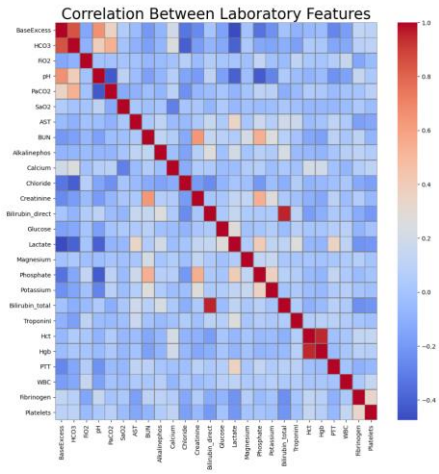



Figure 8: Laboratory Features Correlation Heat Map

Most of the category's features are not negatively or positively correlated. However, a few features in that category correlate to most others. These are BaseExcess, HCO3, pH, BUN, PaCO2, Lactate, and Phosphate.

3. <u>Demographics Features:</u>

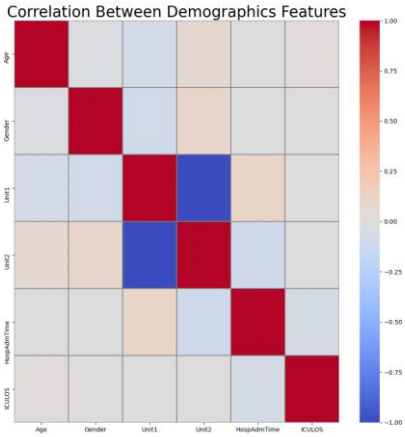



Figure 9: Demographics Features Correlation Heat Map

Except for the negative correlation between Unit 1 and Unit 2, no other features correlate to others.

### 2.3 Handling Missing Data

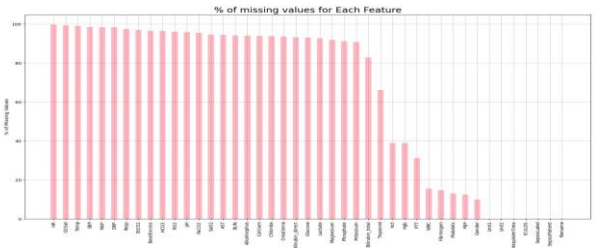



Figure 10: Proportion of missing values in each feature in training data

As shown in Figure [10], most of the features have many missing values, whereas most of the missing values were of lab measurements and had only a small proportion of recorded values. Notably, missing values for specific laboratory tests suggest that these tests were not conducted on some patients. Therefore, we concluded that the data missing is MNAR.

After many discussions with a doctoral student experienced in medical data analysis, Mrs. Shonit Agmon, under Dr. Kira Radinsky, director of diagnostic robotics, we determined several concerns while handling missing data in the dataset:

1. Medical data is very noisy. Trying to complete it is inaccurate because patients only sometimes.
2. go through all the tests. Therefore, it does not simulate reality and adds more noise.
3. We cannot know the factor that affects the disease, so we will leave out all the features.
4. If any feature is at the end irrelevant, the algorithm will give it less weight in calculating the result.

Furthermore, we noted that it is not customary to remove individuals with dry data such as sex and age since the model should still assess whether they will get sepsis.

## 3. FEATURE ENGINEERING

Feature Engineering is a crucial step in any machine-learning project. This section will discuss the features we have selected for our model, the transformations we will apply to them, and any data enrichment we may need to perform.

### 3.1 Features Selection

As we mentioned earlier, we will not delete any feature. This is because We cannot know the factor that affects the disease. Therefore, we do not want to guess and will leave out all the features. If any feature is at the end irrelevant, the algorithm will give it less weight in calculating the result. In the world of medicine, it is not known yet what is the cause of sepsis. Therefore, we decided that deleting any of the features would be irresponsible.

### 3.2 Features Transformations

As we know, for MNAR, all imputation methods are invalid (https://www.sciencedirect.com/science/article/pii/S0895435622002189). In a review of articles that dealt with the prediction for medical processes, it was suggested that adding another binary column for each feature representing whether it is missing should help the model and indicate when data exists and when it is not. Therefore, we suggested a binary column for each feature indicating whether the value was missing. This method can be used with models that can handle missing values, preventing any bias or noise that may occur if data is imputed.
Regarding data enrichment, we decided only to use the data provided to us

## 4. PREDICTION

Prediction is the final stage of a machine-learning project. In this section, we will use four different algorithms to make predictions and report on their performance and interpretability.

### 4.1 Algorithm Selection

During our work, we performed experiments to predict sepsis using various methods. To this end, we treated the data in four ways:

1. We divided the data by accounting for the time component and patient history by aggregating the data for each patient versus neglecting the time component and accounting for each row independently. We did this to check if patient history is essential for predicting sepsis or if one line of data is enough to predict the subject.
2. We investigated whether our binary column for dealing with missing values for each feature would improve the performance of predicting sepsis.

To create the four different datasets for each combination, we connected the dataset of all the patients into one dataset by performing an appending for each independent file of a patient. Then, we saved this file in CSV.

We tested four machine learning algorithms to predict sepsis in patients based on their vital signs and laboratory values. These algorithms were *Naive Bayes*, *Decision Tree*, *Logistic Regression*, and *XGBoost*.

For the first three algorithms, Naive Bayes, Decision Tree, and Logistic Regression, we converted null values to (-1) because these algorithms do not know how to handle missing values.

1. *Naive Bayes* is a probabilistic algorithm that assumes the features are conditionally independent given the target variable. Therefore, we did not tune any hyperparameters for this algorithm.
2. *Decision Tree* is a decision-making algorithm that recursively splits the dataset into subsets based on the value of a particular feature until a target variable is reached. In addition, we tuned the maximum depth and the minimum number of samples required to split an internal node as hyperparameters to prevent overfitting.
3. *Logistic Regression* is a linear regression model that uses a logistic function to model the probability of the target variable. In addition, we tuned the regularization strength parameter to prevent overfitting.
4. *XGBoost* is a boosting algorithm that combines multiple weak learners to improve accuracy. We tuned the learning rate, the number of estimators, and the maximum depth of each tree as hyperparameters. To prevent overfitting, we applied early stopping.

These methods were chosen for their ability to work well with the characteristics of our dataset, which included a large number of features and an uneven distribution of the target variable. We can better understand which algorithms work best for our dataset and might improve the performance of our final model by selecting a different number of algorithms.

We evaluated each algorithm on the four datasets we created from our main consolidated file using different approaches to handle missing values and time dependencies. We evaluated their performance on accuracy, precision, recall, and F1 score.

| | | Original data (all the rows) | Original data with binary col | Aggregated | Aggregated with binary col |
|---|---|---|---|---|---|
| **Decision Tree**<br><br>* we converted null values to (-1) | Accuracy | 0.9980 | 0.9980 | 0.9587 | 0.9587 |
| | F1 score | 0.0 | 0.0 | **0.6595** | **0. 6595** |
| | Confusion Matrix | [[378796 4]<br>[ 741 0]] | [[378800 0]<br>[ 741 0]] | [[9187 72]<br>[ 341 400]] | [[9187 72]<br>[ 341 400]] |
| **Logistic Regression**<br><br>* we converted null values to (-1) | Accuracy | 0.9980 | 0.9980 | 0.9404 | 0.9405 |
| | F1 score | 0.0 | 0.0 | 0.4051 | 0.4079 |
| | Confusion Matrix | [[378800 0]<br>[ 741 0]] | [[378800 0]<br>[ 741 0]] | [[9201 58]<br>[ 538 203]] | [[9200 59]<br>[ 536 205]] |
| **XGBoost** | Accuracy | 0.9980 | 0.9980 | 0.963 | 0.9623 |
| | F1 score | 0.0 | 0.0 | **0.7025** | **0.6976** |
| | Confusion Matrix | [[378782 18]<br>[ 741 0]] | [[378795 5]<br>[ 741 0]] | [[9193 66]<br>[ 304 437]] | [[9188 71]<br>[ 306 435]] |
| **Naive Bayes**<br><br>* we converted null values to (-1) | Accuracy | 0.8060 | 0.7653 | 0.9186 | 0.9186 |
| | F1 score | 0.0093 | 0.0080 | 0.3403 | 0.3403 |
| | Confusion Matrix | [[305598 73202]<br>[ 394 347]] | [[290137 88663]<br>[ 381 360]] | [[8976 283]<br>[ 531 210]] | [[8976 283]<br>[ 531 210]] |

As shown before, in our prediction, we used four different algorithms to make predictions on the dataset: Decision Tree, Logistic Regression, XGBoost, and Naive Bayes. We carefully selected the hyperparameters for each of the first three algorithms to optimize their performance.

1. *Decision Tree* algorithm: criterion="entropy", random_state=100, max_depth=35, min_samples_leaf=50.
2. For *Logistic Regression*, we used mode_random=0 and max_iter=10000.
3. Finally, for *XGBoost*, we used n_estimators=1000, booster='gbtree', objective='binary:logistic', eta=0.2, and colsample_bytree=0.7.

In our prediction part, we built and tested our models using a variety of datasets. However, we discovered that some datasets did not match the necessary assumptions for the models to perform effectively. As a result, we opted to restrict our study only to the 'aggregate' dataset, which produced encouraging findings given the assumptions. Furthermore, we noticed that the XGBoost algorithm performed best. Therefore, we will test and improve it on the particular dataset.

In the next part, we will continue to test and fine-tune our model until we reach the optimal threshold that balances sensitivity and specificity. This will allow us to properly anticipate sepsis in intensive care unit patients and assist healthcare givers in making timely interventions to enhance patient outcomes.

### 4.2 Fine-Tune

After reviewing the outcomes of the various models on different data sets, we focused on one model that could be fine-tuned and one type of dataset for which we conducted aggregation. We used three distinct per-patient aggregation approaches for our data: one standard, one with a binary column showing if the value is missing per feature, and one with a column reflecting the proportion of missing values per feature. We also experimented with various thresholds ranging from 0 to 1 with 10,000 steps in between. This will enable us to assess the performance of our model at different cutoff points and select the best threshold for the XGBoost.

To deeply understand the existing data with the aggregation, we first wanted to check the importance of the various features; for this, we performed the F1 test on all the features of the dataset with the binary columns; as a result, we discovered that the columns we added had importance, which will help us choose the desired model.

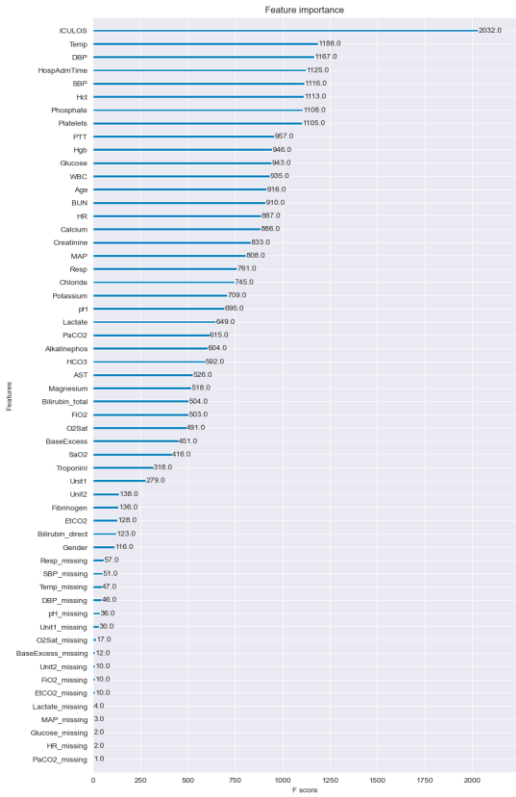



Figure 11: XGBoost Feature Importance on the dataset with the binary columns

We used several forms of aggregation (median, mean + median, mean) on our data while adjusting the model hyperparameters to get the highest possible F1 value. The F1 score is determined as the harmonic mean of the recall and precision values, and a higher F1 score indicates a higher-quality classifier.

We will next present a few of the findings produced during the execution of the chosen model with specific hyperparameters on data with various aggregations. While also investigating which threshold works best.

we used the *XGBoost*, with the selected hyperparameters: n_estimators=1000, booster=

'gbtree', objective='binary:logistic', eta=0.03,max_depth=10, colsample_bytree=0.6,subsample=0.95.

1. The modal on the Aggregated dataset with median aggregation with binary col:

   While using the threshold of 0.20552, we got the Max F1 score of 0.70145.

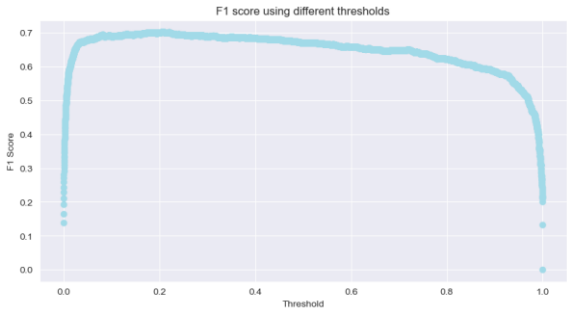



Figure 11: F1 score for the Aggregated dataset with median aggregation with binary col

2. The modal on the Aggregated dataset with median aggregation

   While using the threshold of 0.21532, we got the Max F1 score of 0.69491.

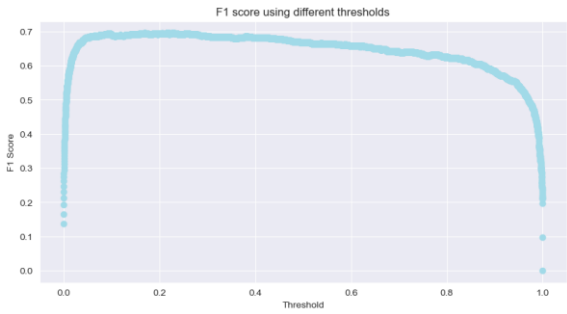



Figure 12: F1 score for the Aggregated dataset with median aggregation.

3. The modal on the Aggregated dataset with mean aggregation with binary col

   While using the threshold of 0.18951, we got the Max F1 score of 0.71094.

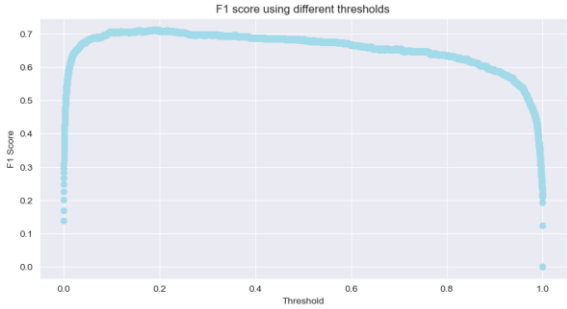



Figure 13: F1 score for the Aggregated dataset with median aggregation with binary col

4. The modal on the Aggregated dataset with median and mean aggregation

   While using the threshold of 0.19151, we got the Max F1 score of 0.71236.

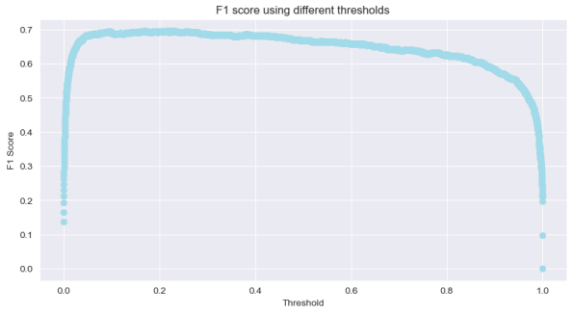



Figure 14: F1 score for the Aggregated dataset with median and mean aggregation.

Figure [11] and F1 findings indicate that the data we presented with the additional columns did not contribute to the accuracy and obtained a lower F1 score.

As a result, the selected data was only aggregated without additional columns. Furthermore, based on the tests we presented and the tests completed in the notebook, we chose to aggregate the patient rows using the mean aggregation, despite the fact that, as is well known, a mean is very sensitive to exceptions. This is due to the fact that the model performed best on this type of data aggregation.

Thus, using the aggregated dataset with mean aggregation, the selected model XGBoost with the following hyperparameters: n_estimators=400, booster='gbtree', objective='binary:logistic', eta=0.6,max_depth=2,

colsample_bytree=0.8,subsample=0.91,learning_rate=0.09,reg_alpha=0.01, reg_lambda=0.05 and the threshold of: 0.35553 we produced the following F1 result: **0.713**

## 5. SUMMARY AND DISCUSSION

Early diagnosis of sepsis in intensive care units is critical for improving patient outcomes. This study aimed to create a machine-learning model capable of predicting sepsis incidence around 6 hours before its diagnosis. To achieve this aim, we used multiple steps, including exploratory data analysis, missing data management, feature engineering, and model selection.

We investigated the dataset and acquired insights into its properties using exploratory data analysis. We afterward addressed missing data, assuring the dataset's integrity and completeness. Feature engineering was used to find and choose the most significant characteristics for sepsis prediction.

We investigated four distinct algorithms to construct our sepsis prediction model, carefully selecting optimal hyperparameters for each one. Furthermore, we evaluated the models' interpretability and performance on several subgroups. Our investigation found that one method, which made use of the aggregated dataset, had the best results. As a result, we concentrated our following study on this dataset.

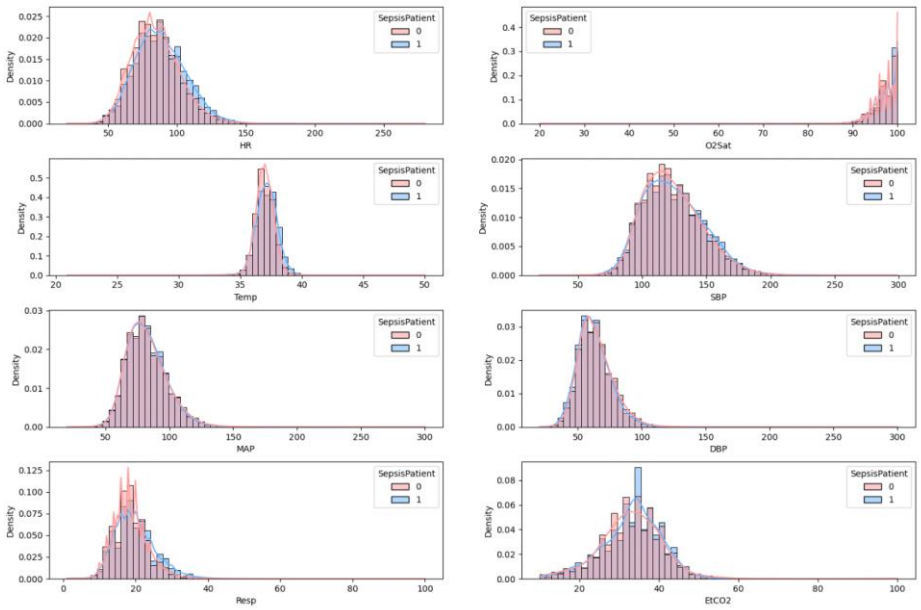
We tried with numerous thresholds inside the aggregated dataset to find the best cutoff point for prediction. Finally, our final model, which used the *XGBoost* method, outperformed the others. Consequently, the F1 score is 0.713.

The outcomes of our experiment emphasize the application of exploratory data analysis, feature engineering, and prediction algorithms in comprehending complex medical information and producing accurate predictions. Using machine learning approaches, we developed a prediction model with great potential for aiding in the early diagnosis of sepsis in critical care units. This work significantly impacts patient care and outcomes in treating sepsis.

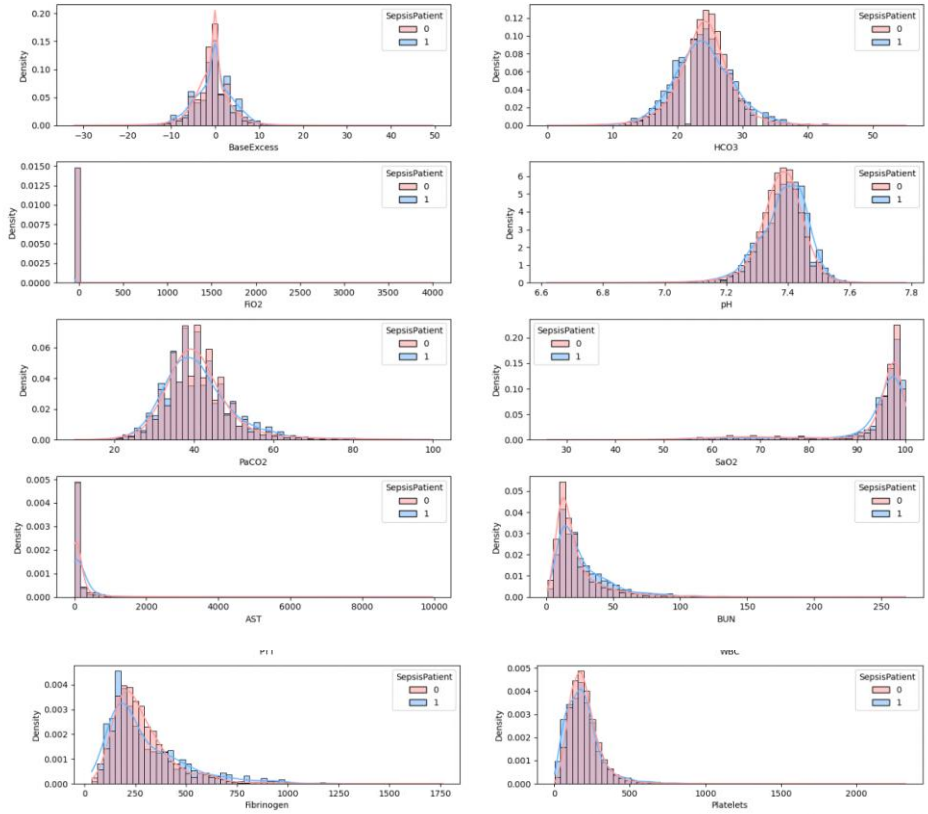
Finally, developing our machine learning model highlights the need for data analysis and prediction algorithms in critical care circumstances. Further research and the use of such models may assist in the early detection of sepsis, resulting in better patient treatment and outcomes.

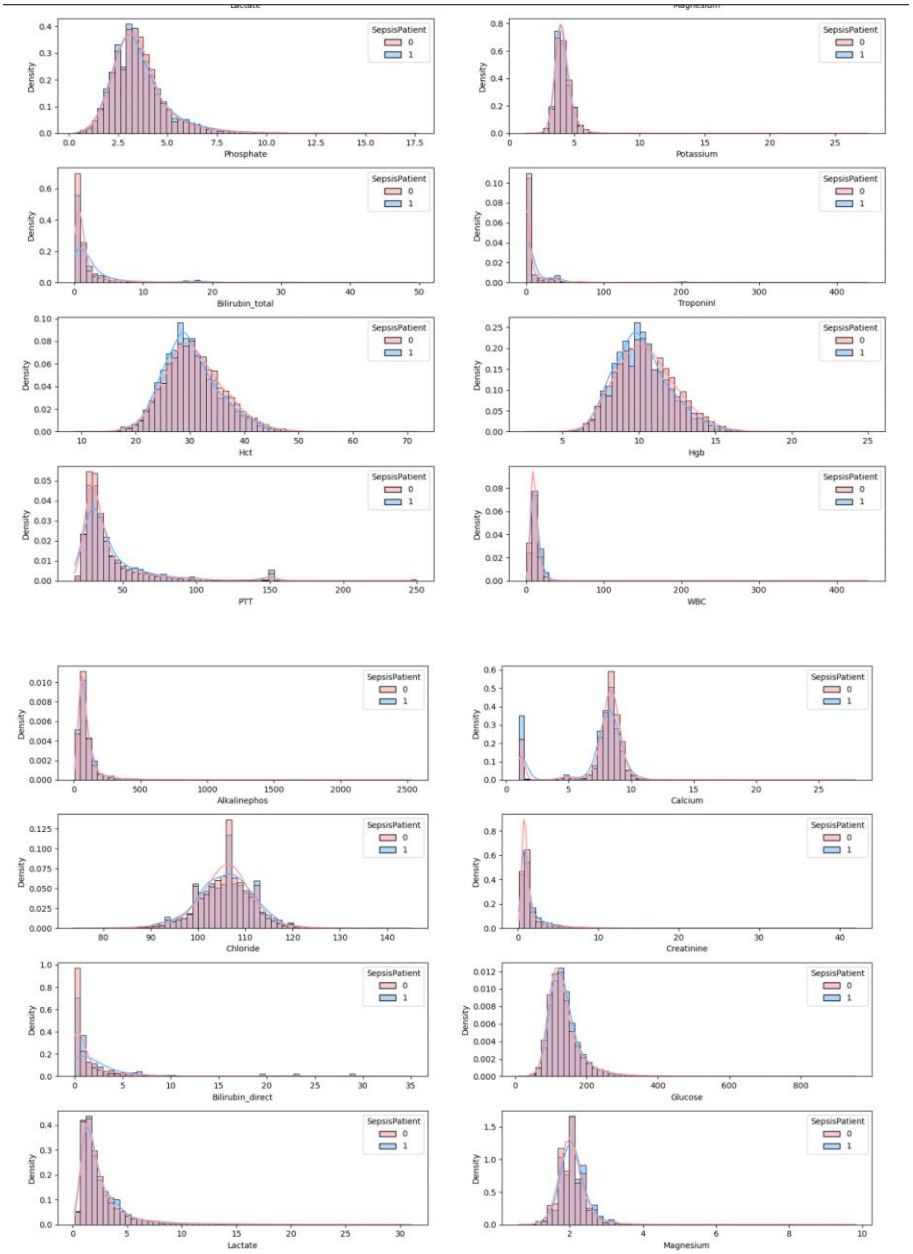
# 6. APPENDIX

Vital signs features:




Laboratory features:




SepsisPatient
0
1
Density
Phosphate
Potassium
Bilirubin_total
TroponinI
Hct
Hgb
PTT
WBC
Alkalinephos
Calcium
Chloride
Creatinine
Bilirubin_direct
Glucose
Lactate
Magnesium

Demographics features: