Sepsis Detection using CBC data

Ross Brinkerhoff
Submitted October 1, 2024

Sepsis is a condition in which the body responds to an infection in an inappropriate way, causing the immune system to turn on the body and leading to problems with organ function, including potential organ failure and death. The condition is a leading cause of death in U.S. hospitals and worldwide according to the WHO and CDC.

One of the main keys in sepsis treatment is catching the condition early, so that the underlying infection can be treated and the patient stabilized before septic shock and organ failure occur. Significant efforts are already underway in to develop an ML based solution for predicting Sepsis onset based on medical data. There is tremendous life saving potential in such a system if it could reliably provide additional treatment time for sepsis patients.

For this project, I used data from two medical centers in Germany to build an ML model that predicts a sepsis diagnosis based on blood test data. The data had already been used as part of an effort to build a sepsis prediction model, and the goal of my project was take the same data and try new modeling approaches.

1: Data

The data used in this project was published as part of a paper (https://academic.oup.com/clinchem/article/70/3/506/7618099?login=false) which detailed efforts by a German team to build a sepsis detection model. The data used in the paper was published and was processed and labeled so that the model built by that team could be reproduced. These labels were not used in my final model as will be detailed below.

The full data set consisted of about 1.8 million rows, each of which contained basic information about a particular patients as well as information on their blood test results. the blood tests consisted of a CBC or “complete blood count” series. A CBC is a common blood test panel consisting of five tests, listed in Table 1.

A single row in the data represented a single blood test result, with some patients having multiple rows and some having only one row. For the Leipzig Medical Center 29.68% of the total patient IDs appear more than once. For Greifswald 54.43% of the Id’s appeared multiple times.

In addition to these 5 tests, two other tests were included for some patients, listed in Table 2. These were not consistently present in the data and were not used in modeling for this project.

Other columns included the patient age, sex, diagnosis (sepsis or not sepsis), and the Seconds from ICU admission that the blood test was taken.

Table 1: The five blood tests included in CBC - used for modeling

Feature	Measurement	Interpretation
HGB	hemoglobin in mmol/l	amount of oxygen carrying protein in the blood
MCV	mean corpuscular volume in fl (femtoliter)	average size of the red blood cells
PLT	platelets in Tpt/l	Low platelet count may indicate an infection
RBC	red blood count in Gpt/l	low RBC (anemia) is associated with sepsis
WBC	white blood count in ng/ml	low levels indicate vulnerability to infection - high levels indicate an infection is present.

Table 2: Two additional tests included for some patients - not used for modeling

Feature	Measurement	Interpretation
CRP	C-reactive protein in mg/l	higher levels associated with inflammation
PCT	procalcitonin in Gpt/l	rising levels are associated with bacterial infection.

The data was further divided from two medical centers as follows:

Leipzig Medical Center - 1.38 million rows
Greifswald Medical Center - 438,000 rows

Generally speaking, the data was built around the assumption that the information from the CBC panel contains enough information to predict a sepsis diagnosis. The idea being that an ML model using common blood tests, that can be conducted in most countries and settings, would have more utility than one relying on more complex or exotic medical data available only in specific high tech settings.

2: Method

I approached this problem as a binary classification problem (sepsis or not sepsis) for a given set of blood test results. There is one caveat to this, which is that I did not want to repeat the work done by the publishers of the original data and paper. They used a RUSBoost model to address the class imbalance in the data and make predictions. I deliberately avoided this method in my project to avoid duplicating their efforts.

My method also differed in that I built two models, one for each center in the data, instead of building one model and asking it to generalize successfully to the other center. Generalization between centers was poor when tried.

Finally, I defined the positive class differently than the publishers of the data. They used only cases within a time window of 0-6 hours from ICU admission as the positive class. My model used all cases where patients were eventually diagnosed with sepsis. Of these, only a small portion fell into a 0-6 hour window, but well over half fell within 2 weeks of ICU admission.

3: Data Cleaning

The data had already undergone extensive pre-processing by the publishers of the paper. This was in part to prepare it for publication by removing identifying patient info (each patient was assigned a random ID number). They had also done extensive work to process the raw medical data into something that would work for their modeling approach. This included combining diagnosis codes in the raw medical data, removing patients based on various criteria and labeling patients as sepsis or non-sepsis based on their actual conditions. This meant that the data structurally reflected the work of the previous team, but it also meant that the data was very clean. The only missing values were in the blood test columns for the two non-CBC tests. It did not seem wise to impute values for medical tests in this case, so these columns were not used in the modeling process moving forward.

I also removed thousands of rows which were marked as being excluded by the publishers based on medical criteria that I was not able to access directly. I chose to trust their judgement in this regard. Other than this issue, no real cleaning was required.

4: EDA

EDA revealed a number of important factors in the data.

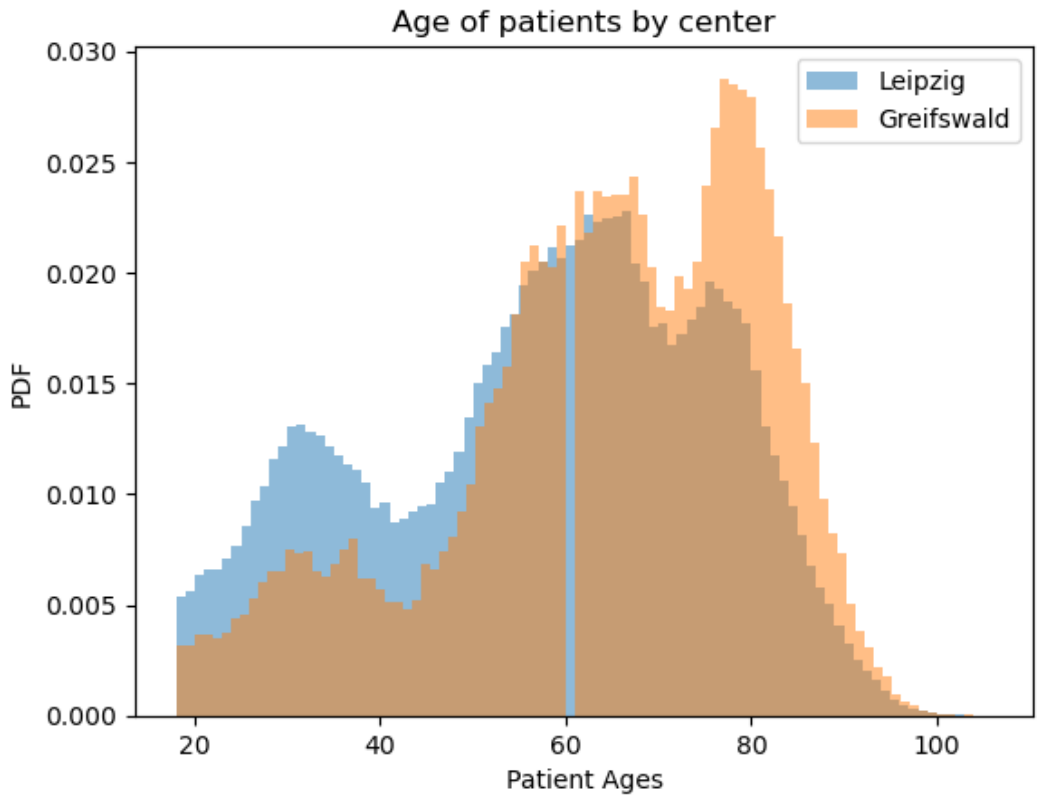
Large Class Imbalance

First, the class imbalance between sepsis and non-sepsis cases was very large.
Including all cases diagnosed as sepsis helped but even with that the class ratios (negative to positive were as follows):

Leipzig Medical Center	81.94 to 1 or 1.2% positive
Greifswald Medical Center	109.24 to 1 or 0.9% positive

Difficult to Compare Data from Each Center

Second, the two medical centers in the data did not appear to be comparable in terms of the number of available rows or the characteristics of the population. More specifically, the age of the Greifswald center was much higher. The difference was statistically significant based on both parametric and non-parametric tests.

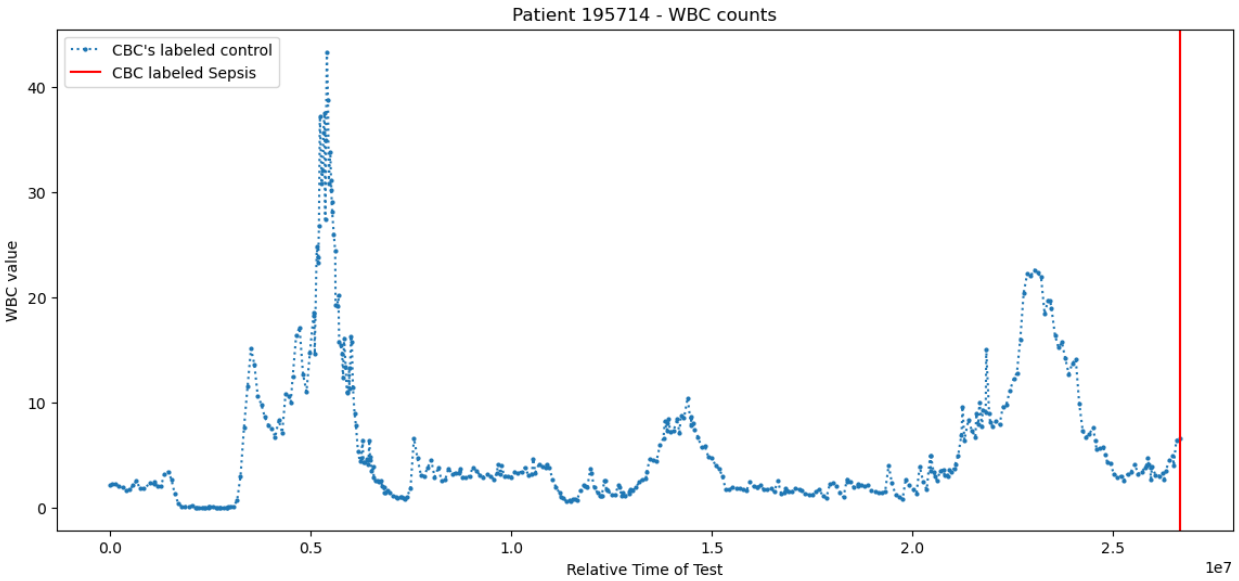


This combined with the fact that the centers are hundreds of miles apart, contributed to the decision to treat each of them separately in terms of modeling.

Moving forward, the primary modeling efforts were focused on building a model for the larger Leipzig Center data. Once developed, this modeling approach was then also applied to the smaller Griefswald center.

Potential for Time Series Dimension

Third, EDA revealed the potential for a significant time series dimension to the data. Most of the patients had only one blood test in the data, but some had multiple tests. These tests could be used to chart a time series of their results for a given blood test. There were not enough patients with multiple tests to really do a full time series analysis, but I would later build some additional features (see Feature Engineering Section) to capture some of this time dependent data. An example of time series data for a patient with a large number of tests is below.

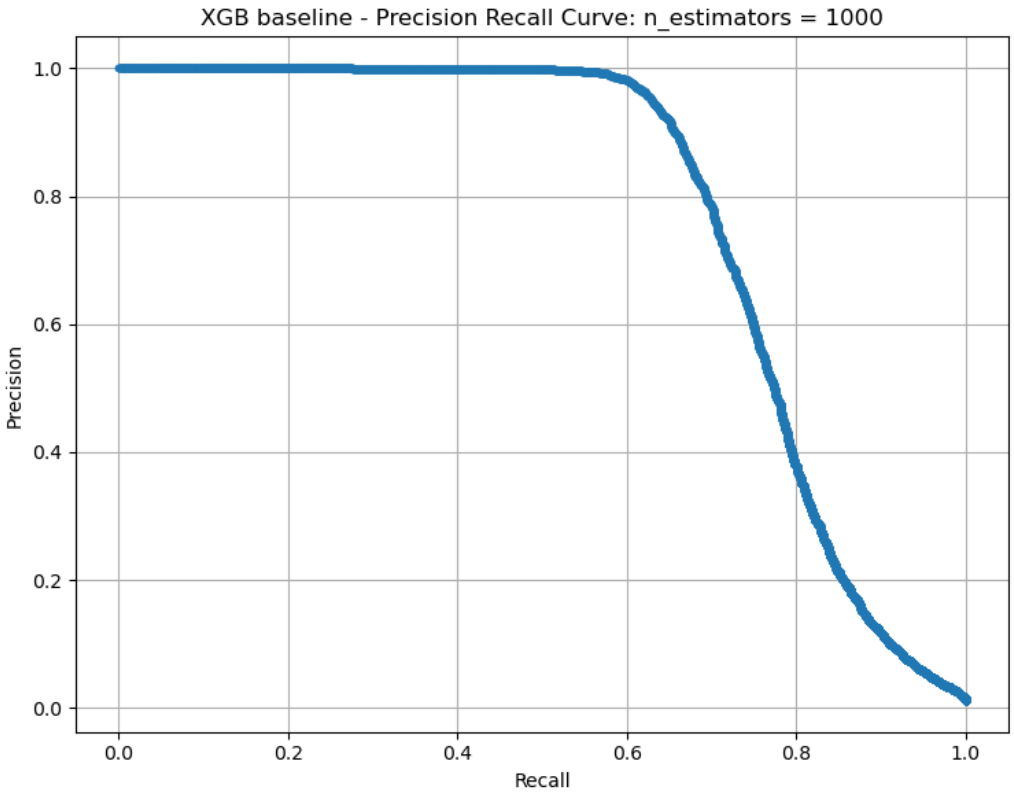


Example of time series data for a patient with multiple tests. The red line indicates the one test labeled as sepsis in the original paper. Tests in blue were oversampled in the original paper. In my model, the best performance was achieved by labeling all tests as sepsis.

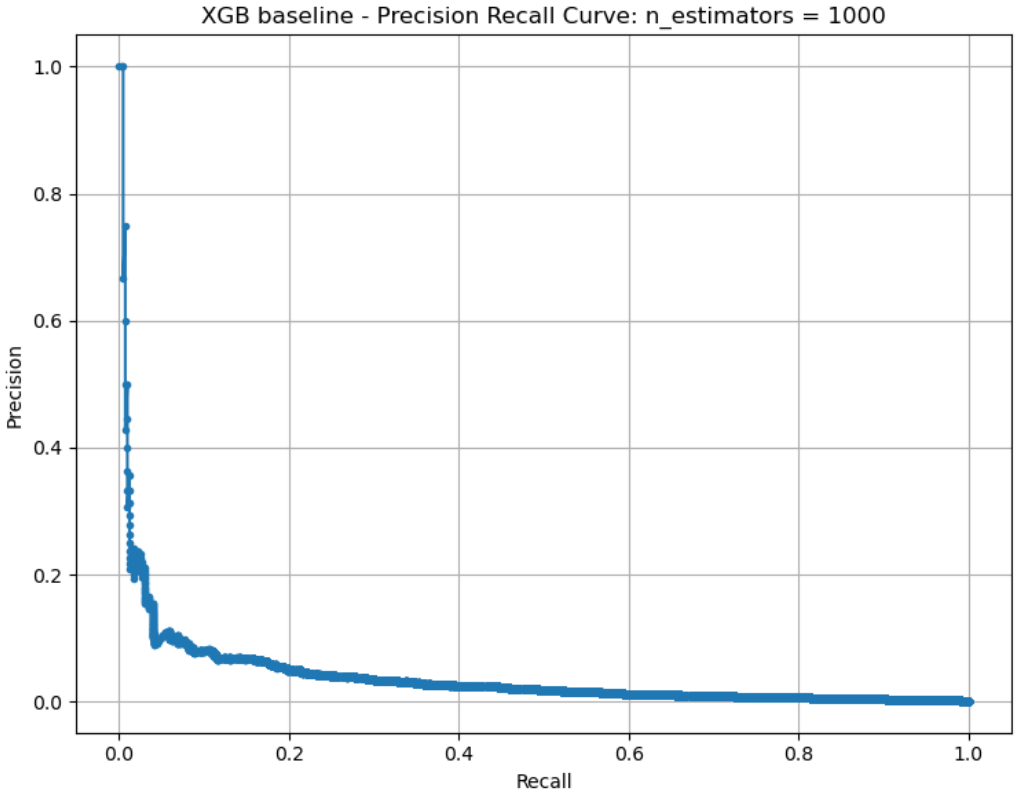
The previous team captured some of the time dimension of the data by oversampling the pre-septic blood tests from septic patients by a factor of 10 as part of building their model. Pre-septic in this case being any blood test from a patient outside their 6 hour window.

Issues with subsetting the data based on time from ICU admission

The prior work detailed in the paper used a 6 hour time window from ICU admission but subsetting the data in that way led to basically bad results in all the models I tried. I believe that this was because subsetting the diagnosed sepsis classes in this way lead to an even more extreme class imbalance. When more positive class examples were used, the model performance predictably improved. Based on this I decided to move forward with using the Diagnosis column as the determining feature for the initial modeling efforts. This allowed me to use all available cases with a positive diagnosis. Below are examples of the same un-tuned XGB model trained on all diagnosed cases and on only cases within a 6 hour window of ICU admission.



Precision and Recall or model trained on all diagnosed cases



Precision and Recall for model trained on 6 hour window positive class

Finally, as part of EDA, I also examined the data using K-Means clustering. The data clustered strongly into 2 groups, one much larger than the other. The groups showed a clear separation, and the differences in the two groups for the blood tests results was statistically significant. Unfortunately these differences did not seem to correspond to the Sepsis label very well. It's possible that the K-means model was separating the groups based on WBC (white blood count) or on some other factor that was not included in the data set. I wasn't able to gain much insight into the problem of separating the sepsis and non-sepsis classes from this portion of EDA.

5: Feature Engineering

The individual rows did not contain any time dependent features, even for patients that had multiple rows spanning long periods of time. To capture some of the time dependent characteristics in the data. I added several features to all rows.

For each blood test in the CBC panel, the following features were added:

- Cumulative Mean
- Cumulative Median
- Cumulative Standard Deviation
- Exponential Moving Average using 2 periods
- Exponential Moving Average using 6 periods

I also added additional features for:

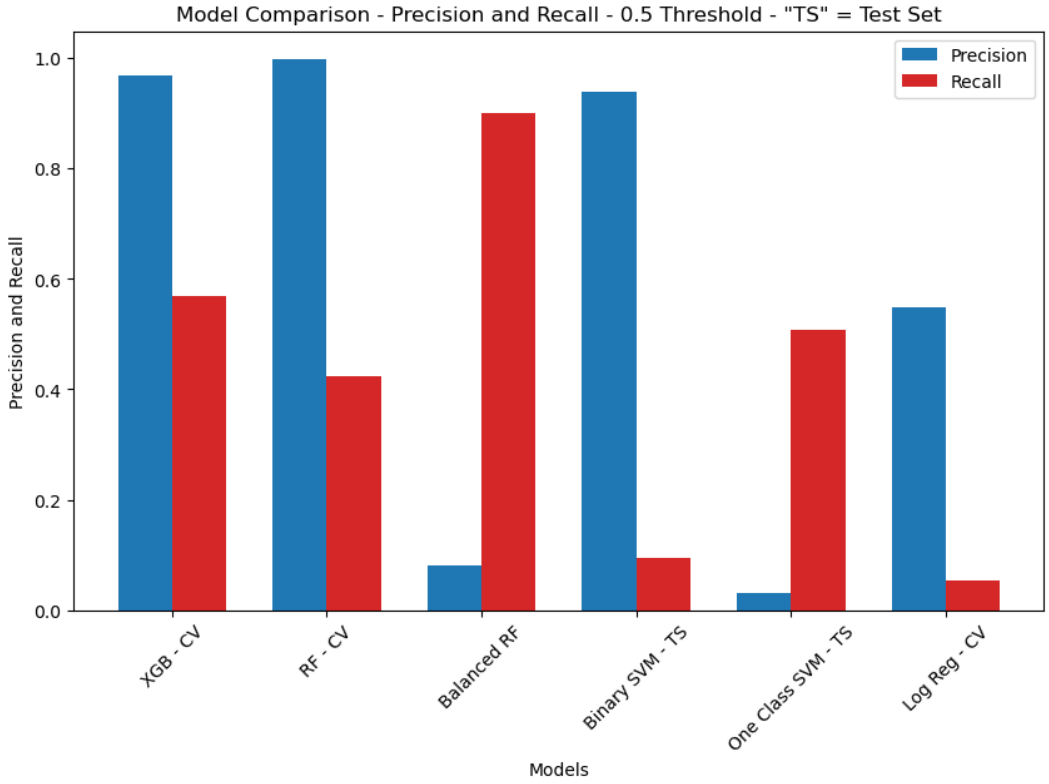
- test sequence
- interval from last test
- mean time between tests
- median time between tests

6: Algorithms and Modeling

I tried several algorithms initially to see which would do best at modeling the data. For initial results the models were used with their basic default settings. The following algorithms were tried:

- XGBoost
- Random Forest
- Balanced Forest
- SVM
- Single Class SVM
- Logistic Regression

In all cases the tree based models did best, and XGBoost was the best of these. Random Forest was a close second.



XGBoost showed the best out of the box performance

I also compared three sampling approaches to the data in an effort to correct the imbalance in the data set. Random under sampling was tried with poor results as well as SMOTE and SMOTEENN. SMOTE and SMOTEENN did better than under sampling, but both caused overfitting, and generalized poorly to the test set. Using the imbalanced data, without any over- or under sampling, gave the best performance.

7: Final models

Having chosen XGBoost as the final model, I tuned the hyper-parameters using the skopt module and focused on f2 as my metric. Because this is a medical diagnosis problem, using F2 instead of F1 made sense because it prioritized recall but still provided a balanced metric. Accuracy for all modeling efforts was 0.98 or better because of the class imbalance, making that metric useless for gauging actual performance. After tuning, I used cross validation to find the best classification threshold, and applied the tuned model with the optimal threshold to the test set. Results and details on the final models are available in the respective model metrics documents

After training a final model on the Leipzig data, I used the same methodology to train a model on the Grefiswald data. Both models did reasonably well on the test set from their medical center, but generalized very poorly to the data from the other medical center. Performance for the Greifswald model was not as good as the model for Leipzig. I believe this is because the number of positive class examples available for the Griefswald center was only about 23% the number available for training the Leipzig model. The class imbalance for the Greifswald model was also slightly greater.

From a business perspective, this difference in performance is important, because it speaks to the difficulty of implementing a local modeling approach. If the locality that is the focus of the model has less data, or lower data quality, that model will suffer accordingly.

8: Model Generalization

The models for each center did poorly when provided with data from the other center. Given the overall approach to the problem this was not a surprise. Performance of each trained model on it's own test set and the other medical centers data is as follows:

Leipzig Model Performance Metrics

	Accuracy	**Precision**	**Recall**	**F1**	**F2**	**MCC**
Scores on Test Set Threshold of 0.274	0.99405	0.74772	0.76434	0.75594	0.76095	0.75297
Scores on unseen Greifswald Data Threshold of 0.274	0.98711	0.17248	0.11072	0.13487	0.11926	0.13191

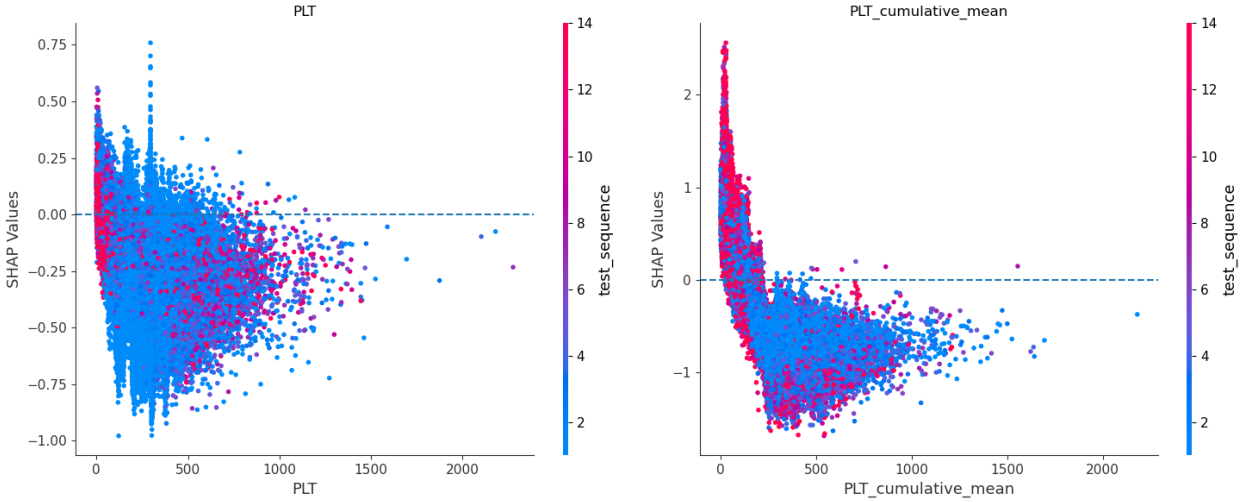
Greifswald Model Performance Metrics

	Accuracy	**Precision**	**Recall**	**F1**	**F2**	**MCC**
Scores on Test Set Threshold of 0.252	0.9945	0.71706	0.65031	0.68206	0.66265	0.68011
Scores on unseen Leipzig Data Threshold of 0.252	0.9782	0.14743	0.16903	0.15749	0.16422	0.14685

9: Model Interpretation

I used the SHAP package for model interpretation on both models. This revealed that the actual blood test numbers had a relatively small effect on the model output. This was not expected, since these features were the core of the data set. The higher impact features were all time based features, whose value was dependent on things like the cumulative mean of multiple blood tests over time. This was true for all the blood test features to some degree. The color coding below is for test sequence, which is a feature consisting of an integer that corresponds to the position of a test relative to other tests for that patient in the data. Patients with only one test in the data would have a test sequence value of one. The SHAP plot below if some the Leipzig model, but the SHAP analysis from the Greifswald model showed similar patterns. For all CBC values, the time dependent features had a higher importance than the actual CBC value.

Leipzig Model PLT vs. PLT cumulative Mean



The time dependent feature was higher in importance and shows a clearer impact.

10: Recommendations

The modeling process, and particularly the SHAP analysis, suggested a number of areas for future work. I was also able to speak to medical professionals who provided input on how they look for sepsis in a hospital setting. Those conversations informed the recommendations below as well. In general, the recommendations here focus on strategies for gathering additional relevant data.

1. Find additional data or additional features that better capture the time dependent component.
The SHAP analysis seemed to indicate that the blood test data in and of itself is not the most important factor in driving prediction, instead, the time dependent features seemed to have the greatest effect. This suggests that sepsis detection may be best approached as a time series problem. The data set here contains some time series information, which was the basis for the

time dependent features, but most of the blood test data in the data set does not have this component. Framing the problem explicitly as a time series problem may provide a good framework for additional data acquisition and modeling efforts.

2. Determine what other factors are used in the current diagnosis of sepsis and incorporate them into additional data gathering efforts.
The data used for modeling contained five features that were actual medical data, namely the CBC panel test results. The medical professionals I spoke with indicated that they use more than 5 criteria when evaluating a patient for the risk of sepsis. These additional data points, for example BPM, should be defined and incorporated into future modeling efforts.

3. Include prior conditions or actions, for example recent antibiotic use, which can be risk factors for sepsis.
The data set here did not include information on other or prior diagnoses. Including this sort of information in future data sets might lead to improved model performance.

4. Try new modeling approaches, for example deep learning.
New modeling approaches should be tried in combination with the data gathering efforts above to improve prediction performance and better capture complex relationships within the data. No deep learning models were built for this project, and I did not do any work to develop RUSBoost or standard ADA boost models as these were used by the publishers of the paper. Further development of those methods and of various deep learning methods should be tried.

5. Gather more data from single locations on blood tests taken close to ICU admission.
The data as it is now has a very high class imbalance, but there does seem to be some signal within the noise. If more positive class examples could be gathered, a better working model of CBC tests within a small window before ICU admission could likely be produced.

Model summary for final classifier for Liepzig Medical Center

The final model for this project was an XGBClassifier model. The model underwent 80 search rounds of the hyper parameter space using skopt gp.minimize function, which was set to minimize an inverted F2 score.

Training and Test Set Size and Class Imbalances

Training Set	Number of examples	Approximate Class Percentage
Negative class	1091763	98.79%
Positive Class	13323	1.2%

Test Set	Number of examples	Approximate Class Percentage
Negative class	272941	98.79%
Positive Class	3331	1.2%

Search Space

XGBClassifier Parameter	Parameter min	Parameter max
n_estimators	100	1500
max_depth	0	12
scale_pos_weight	1	50
eta	0.1	0.9
gamma	0	10
min_child_weight	0	10
subsample	0	10
lambda	1	10
alpha	0	10

Hyper-Parameter Tuning

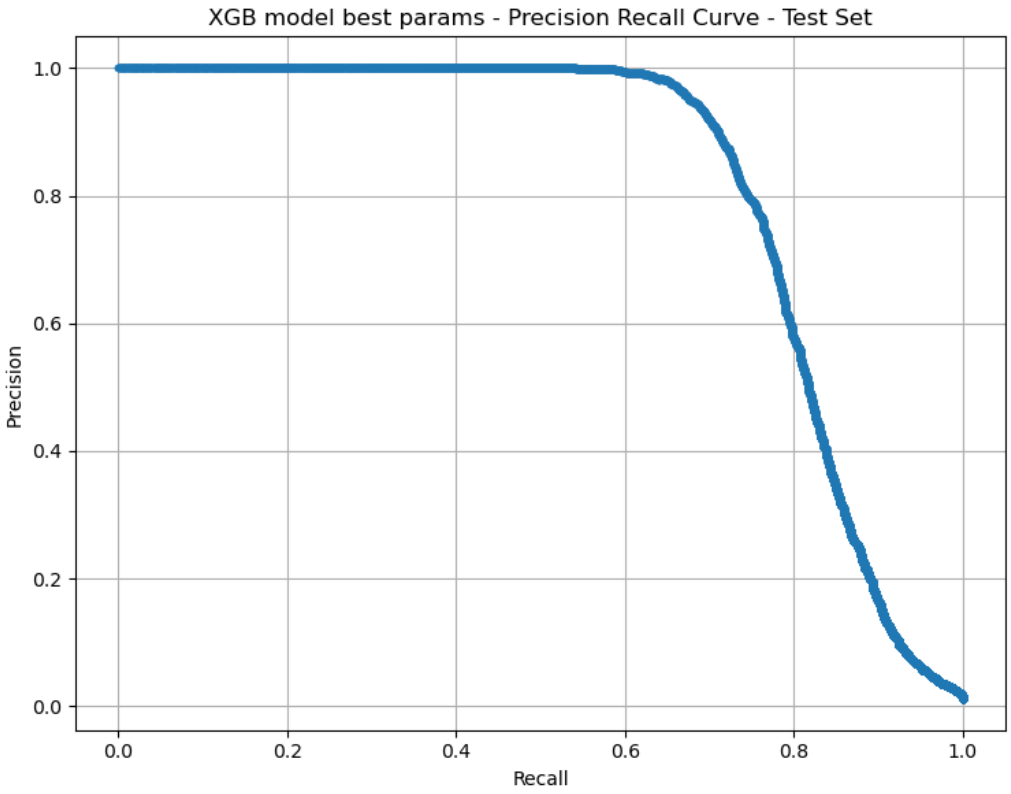
After hyper-parameter tuning, the best parameters found were as follows. Parameters with default values are not listed.

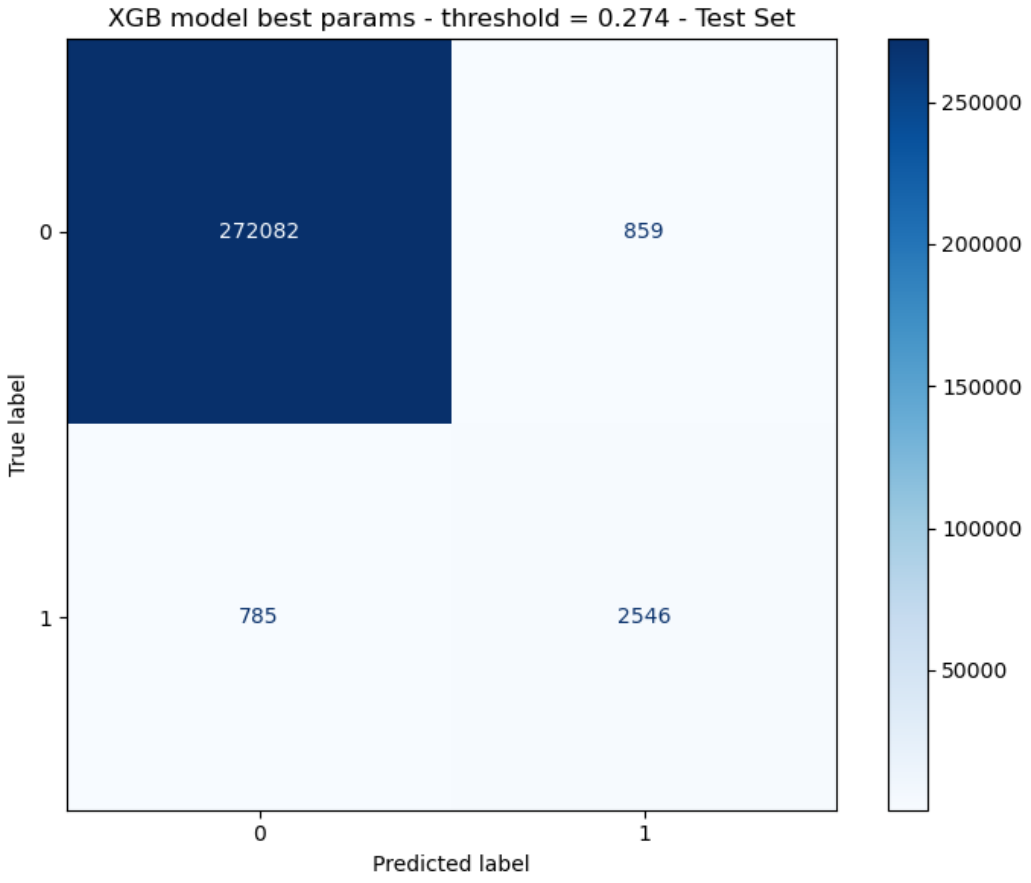
The model was further tuned by using cross validation to find the best classification threshold, which was approximately 0.274

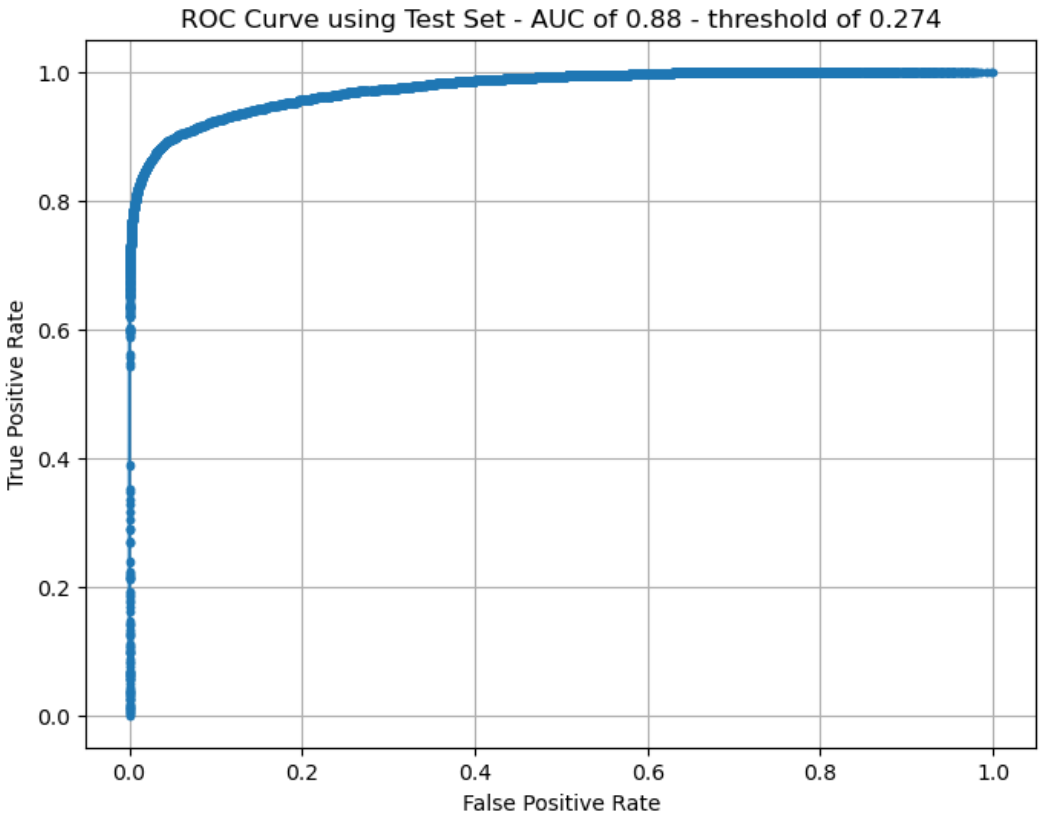
Parameter	Value
objective	binary:logistic
eval_metric	logloss
gamma	0.0
max_depth	11
min_child_weight	10.0
n_estimators	1500
random_state	42
reg_lambda	1.33291364852387
scale_pos_weight	50
subsample	0.92492835736912
eta	0.1
alpha	10.0

Performance Metrics

	Accuracy	Precision	Recall	F1	F2	MCC
Scores on Test Set Threshold of 0.274	0.99405	0.74772	0.76434	0.75594	0.76095	0.75297


XGB model best params - Precision Recall Curve - Test Set
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


XGB model best params - threshold = 0.274 - Test Set
True label
Predicted label
272082
859
785
2546
0
1
0
1
250000
200000
150000
100000
50000


ROC Curve using Test Set - AUC of 0.88 - threshold of 0.274
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

Model summary for final classifier for Greifswald Medical Center

The final model for this project was an XGBClassifier model. The model underwent 80 search rounds of the hyper parameter space using skopt gp.minimize function, which was set to minimize an inverted F2 score.

Training and Test Set Size and Class Imbalances

Training Set	Number of examples	Approximate Class Percentage
Negative class	347282	99.09%
Positive Class	3179	0.9%

Test Set	Number of examples	Approximate Class Percentage
Negative class	86821	99.09%
Positive Class	795	0.9%

Search Space

XGBClassifier Parameter	Parameter min	Parameter max
n_estimators	100	1500
max_depth	0	12
scale_pos_weight	1	50
eta	0.1	0.9
gamma	0	10
min_child_weight	0	10
subsample	0	10
lambda	1	10
alpha	0	10

Hyper-Parameter Tuning

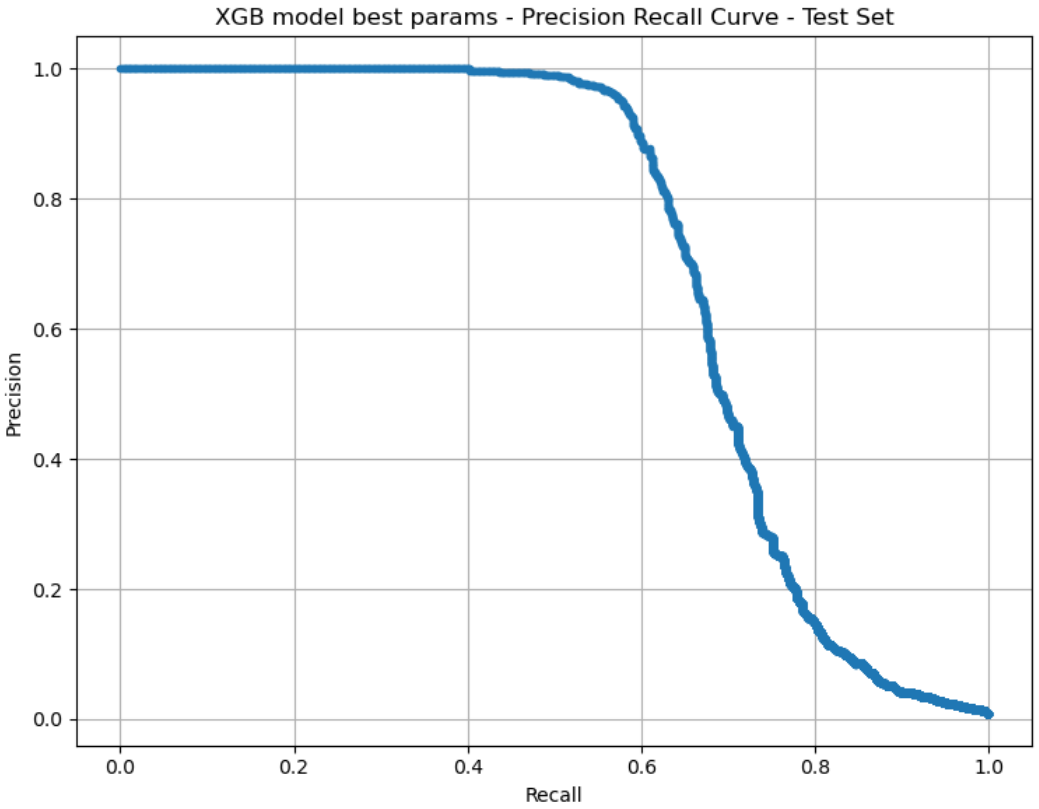
After hyper-parameter tuning, the best parameters found were as follows. Parameters with default values are not listed.

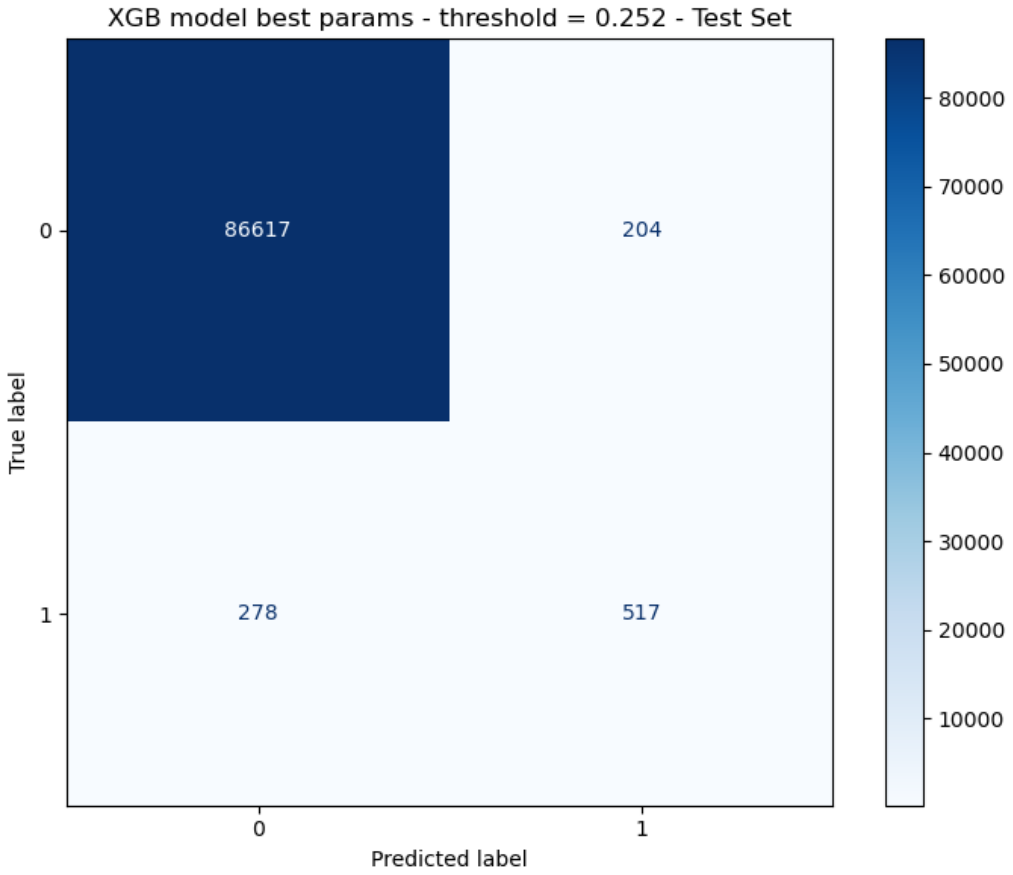
The model was further tuned by using cross validation to find the best classification threshold, which was approximately 0.252.

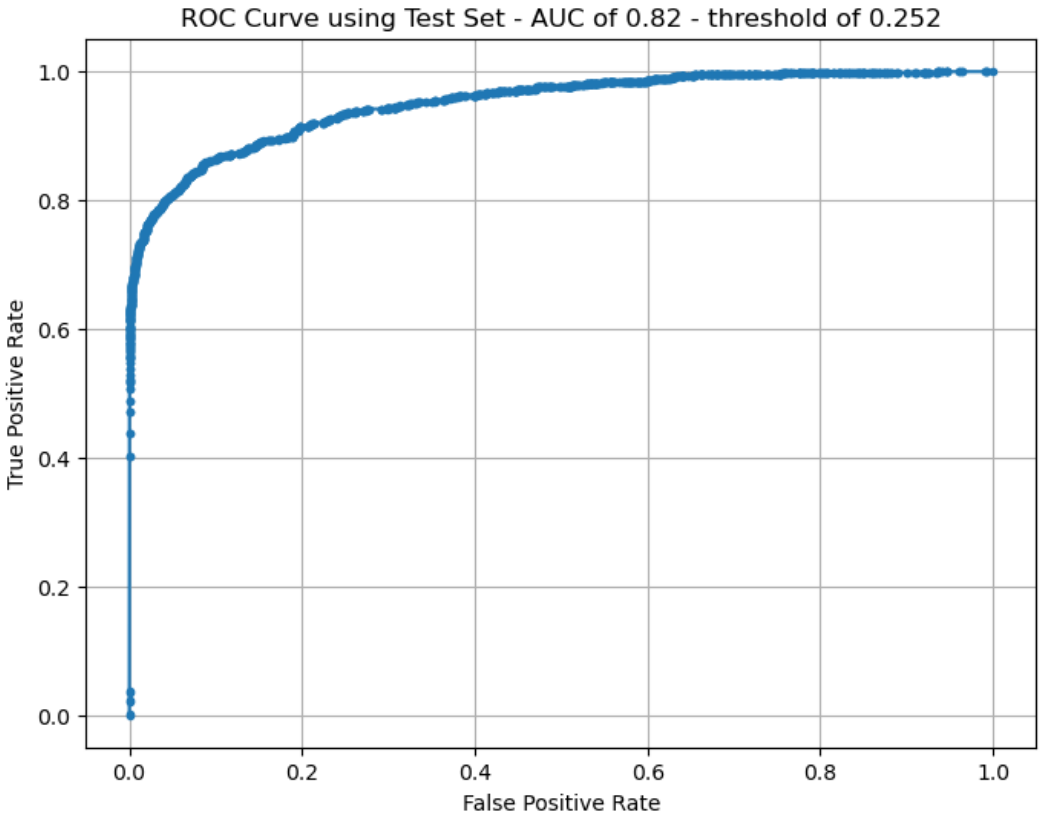
Parameter	Value
objective	binary:logistic
eval_metric	logloss
gamma	0.0
max_depth	12
min_child_weight	6.78158325402969
n_estimators	1500
random_state	42
reg_lambda	9.18278158022484
scale_pos_weight	50
subsample	0.738982779620338
eta	0.1
alpha	10.0

Performance Metrics

	Accuracy	Precision	Recall	F1	F2	MCC
Scores on Test Set Threshold of 0.252	0.9945	0.71706	0.65031	0.68206	0.66265	0.68011


XGB model best params - Precision Recall Curve - Test Set
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


XGB model best params - threshold = 0.252 - Test Set
True label
Predicted label
86617
204
278
517
0
1
0
1
80000
70000
60000
50000
40000
30000
20000
10000


ROC Curve using Test Set - AUC of 0.82 - threshold of 0.252
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0