Classifying Septic Shock using TREWScore Algorithm

Kelly Cheng
Georgia Institution of Technology, Atlanta, GA, USA

Abstract

Annually, septic shock accounts for between twenty and thirty percent of hospital deaths, but early intervention can significantly decrease the likelihood of mortality. This means that the early identification of patients who are likely to develop septic shock is critical to increase the odds of patient survival. This study defines a classification model for determining whether patients in ICUs will experience septic shock. The methods used on this project are largely based off of the TREWScore Algorithm introduced in the paper "A targeted real-time early warning score (TREWScore) for septic shock"[1]. Elements from other literature were used to tweak the algorithm to seek improvement from the original. The TREWScore Algorithm obtains an Area Under the Receiver Operating Characteristic Curve (AUC) of 0.83 with a median of 28.2 hours before onset of septic shock. Using data from the MIMIC-III database, this study resulted in an AUC of 0.86 with a prediction time of 28 hours before onset of septic shock.

Introduction and Motivation

Sepsis and septic shock are identified as a leading cause of death in the United States, with seven hundred and fifty thousand patients developing them per year, accounting for 10% of all ICU admissions and 20-30% of hospital deaths[1]. The sheer amount of patients that develop septic shock as well as the rate of mortality among those who develop them motivates the study to identify potential warning signs early so that health care professionals can begin treatment, delaying or eliminating the onset of sepsis or septic shock.

There have been several models for predicting septic shock. The TREWScore algorithm developed by Henry et. al. identified patients before the onset of septic shock with a AUC of 0.83 and identified patients at a median of 28.2 hours before onset. The Modified Early Warning Score (MEWS)[2] achieved an AUC of 0.73. The Real-time Early Warning Score (REWS)[4] did not have an AUC specified, but was indicated to be less reliable than MEWS. PRESEP Early Warning Score (PEWS) had a large AUC of 0.93, however, dataset was much smaller (MEWS AUC was 0.77 in this dataset)[3].

This study defines a classification algorithm which uses a combination of features identified by importance of features in related literature. It achieves an AUC of 0.86 at 28 hours before onset.

Problem Formulation

Henry et al. fit a Cox Proportional Hazard Model[1] to better estimate the non-uniform severity of sepsis with time. Because the Cox proportional hazard model is not available in Spark's machine learning library (MLlib)[6], this study does not recreate the model. Instead, it defines a classification algorithm using a fixed time point of 28 hours as its prediction time. This facilitates its comparison to TREWScore's median prediction time.

Approach and Implementation

The original work on TREWScore was developed using MIMIC II data. This study uses data from MIMIC III. Data is gathered from the database by downloading the MIMIC III database's CSV files. PostgresSQL, an implementation of Structured Query Language (SQL), was used to initially view the data and make decisions on cleaning and interpreting it into usable data (namely, identifying Item IDs of key features). To facilitate speed, PostgresSQL was also used to pre-filter data, as it is much faster than a Spark solution. Data processing was done in Spark 1.6.1, using the Apache libraries as well as the ML Library. This study made use of local clusters for Spark and python.

Key data is gathered from the MIMIC III database. Key data includes patients, along with whether they develop septic shock, and index date. Other key data is the features that are used to determine the algorithm.

Table 1 shows a summary of the patients included in the cohort. There are a total of 38,646 patients included in the cohort with 7,874 patients excluded due to age. The cohort is limited to adults (patients aged 15 and up).

Septic shock (the target) is determined based on diagnosis gathered from the MIMIC III diagnoses table (ICD9 codes 670, 785, 995). Patients are labeled 0 for did not experience septic shock (control) or 1 for experiencing septic shock (case). Of the 38,646 patients, 4,158 were identified as case patients.

Table 1. Cohort Construction

Total Patients in Cohort	38,646
Case Patients	4,158
Control Patients	34,488

Index date identification was taken from a combination of other literature and the TREWScore Algorithm. Based on the TREWScore algorithm, a prediction window of 28 hours before onset of sepsis was used so that a comparison could be made. The index date was defined as 28 hours before onset for case patients and time of discharge for control patients[1]. The onset of septic shock in this case is defined as when a patient who has been diagnosed with sepsis, septic shock, or severe sepsis has hypotension for at least ninety minutes based on Shavdia[5]. The observation window includes any event prior to the patient's index date, as all of that information is available in the database and presumably available in the patient's EHR at the hospital.

Features that are being considered include: diagnoses for diabetes, heart failure, hematological malignancy, immunocompromised, liver disease, metastatic carcinoma, organ insufficiency, and renal insufficiency; other events heart rate, systolic blood pressure, heart rate/systolic blood pressure, time since last antibiotics in hours, creatinine, BUN, and BUN/creatinine. These were identified either in the TREWScore algorithm or in other literature with values indicative of sepsis or septic shock[1,4,5]. For diagnoses, the values were 0 or 1 for whether the patient was diagnosed or not. Diagnoses were based on ICD9 codes associated with the patient. For other events, unless otherwise stated, the values of each unique feature were taken at the last time before the patient's index date. For features that had multiple values with the same timestamp, the average of the values was taken.

The patient/feature data is split into 80% training and 20% test using a random split. Existing literature has split 80%/20% for their data[1,5]. The random split is the fairest way to split training and testing data to make sure there is not bias.

Experiment Design and Evaluation

Modeling Pipeline. The modeling pipeline begins by copying data from the MIMIC III database onto the local machine. It is then imported into PostgresSQL and prefiltered. The prefiltering separates the patients in the cohort from the patients outside of the cohort. This massively decreases the number of records that need to be accessed. Also, to facilitate greater speeds, the patients were labeled and index date identified in PostgresSQL. This was so that events outside of the observation window – after the index date – could be filtered out, greatly decreasing the number of events. At this point, unnecessary data is also filtered out to decrease the volume that Spark needs to have input. Unnecessary data includes non-related events, medications, and diagnoses, redundant column values and columns that are never used in analysis. Essentially, the goal in this portion was to reduce as much as possible the volume of data importing into Spark. This is then written into CSV files that can be read into Spark.

The filtered features were manipulated and analyzed in Spark. Patients were labeled as described in the study design section. Features for each patient were aggregated into sparse vectors where the feature values are defined for each feature. These were gathered into LabeledPoints, where each LabeledPoint contained the label for the patient (case or control) and the sparsevector of feature values. This is now formatted in a way that MLLib library methods can analyze the data. It was fed into analysis methods in the Spark MLLib library to create the model. Finally, the Receiver Operating Characteristic curve (ROC) and Area Under the Curve (AUC) are calculated using the BinaryClassificationMetrics methods built in MLLib library.

Results and Evaluation. Multiple models were considered.

Two of the models that were created used Support Vector Machines with Stochastic Gradient Descent (SVM with SGD) and Logistic Regression. The figure below show the ROC for the two different regression models.

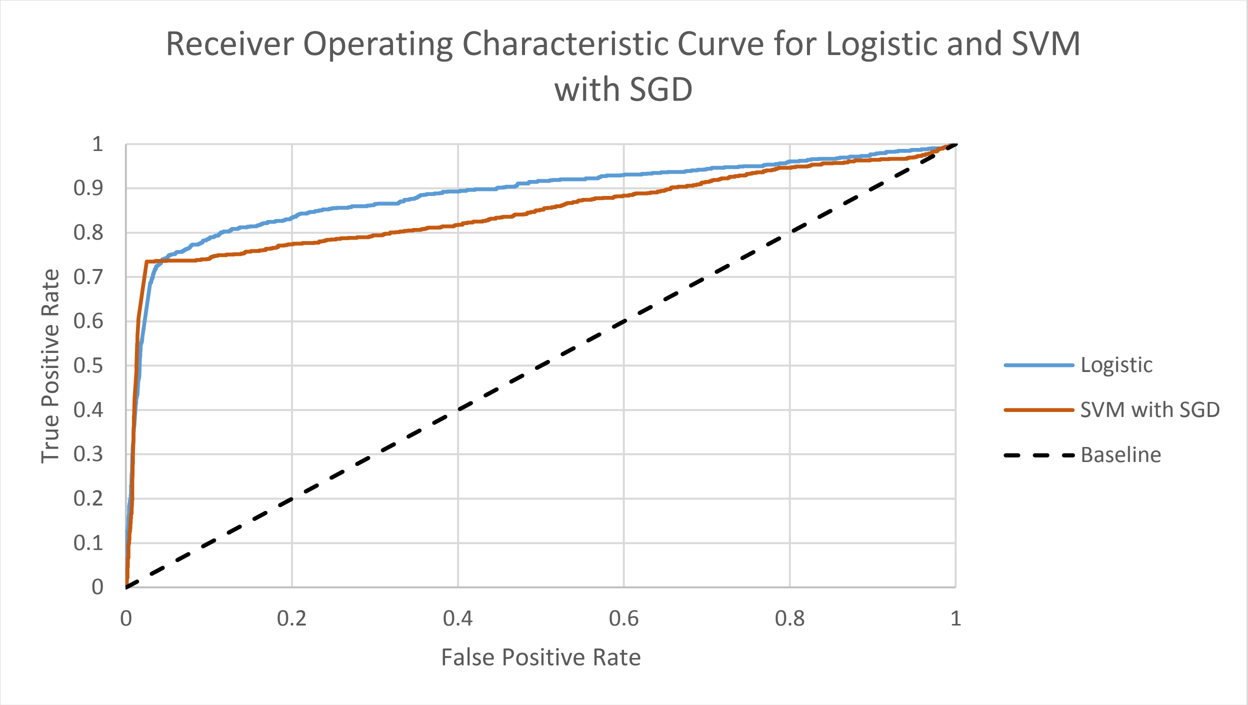

Figure 1. ROC of Logistic and SVM with SGD regression models.

Both ROCs look similar, with the logistic regression's model being slightly higher than the SVM with SGD regression model. Though the AUCs appear high, note that Table 1 shows that these models have extremely low recall and f-Score. This is most likely because of the large size difference in the cohort between case and control patients.

Table 1. Results for different regression models.

Statistic	**Logistic Regression**	**SVM with SGD**
AUC	.8868	.8472
Precision	.9639	1.0
Recall	.001	.001
f-Score with beta = 1.0	.002	.002

Note that though the AUCs are exceptionally high, with Logistic Regression at 0.89 and the precision also being high, recall and f-Score are exceptionally low. This is due to an overgeneralization of these patients. The algorithms will most likely mark all patients as not experiencing septic shock, which does have high accuracy because most will not, but will cause a lot of false negatives.

To solve for this, the study settled on using ensemble methods. These resulted in slightly lower, but still very comparable, ROCs / AUCs and lower precision scores but significantly higher recall and f-Score. Thus, these were determined to be better models. The reason for this is because ensemble methods produce a number of decision trees as base models and ensemble them together into a model that can more robustly describe off-balance data sets. The two models used were the two most widely support by Spark's MLLib: Random Forest and Gradient-Boosted Trees (GBT). Maximum depth for the algorithms were set at 10, and number of trees was varied to find the most appropriate model.

The figure below shows the ROCs for each.

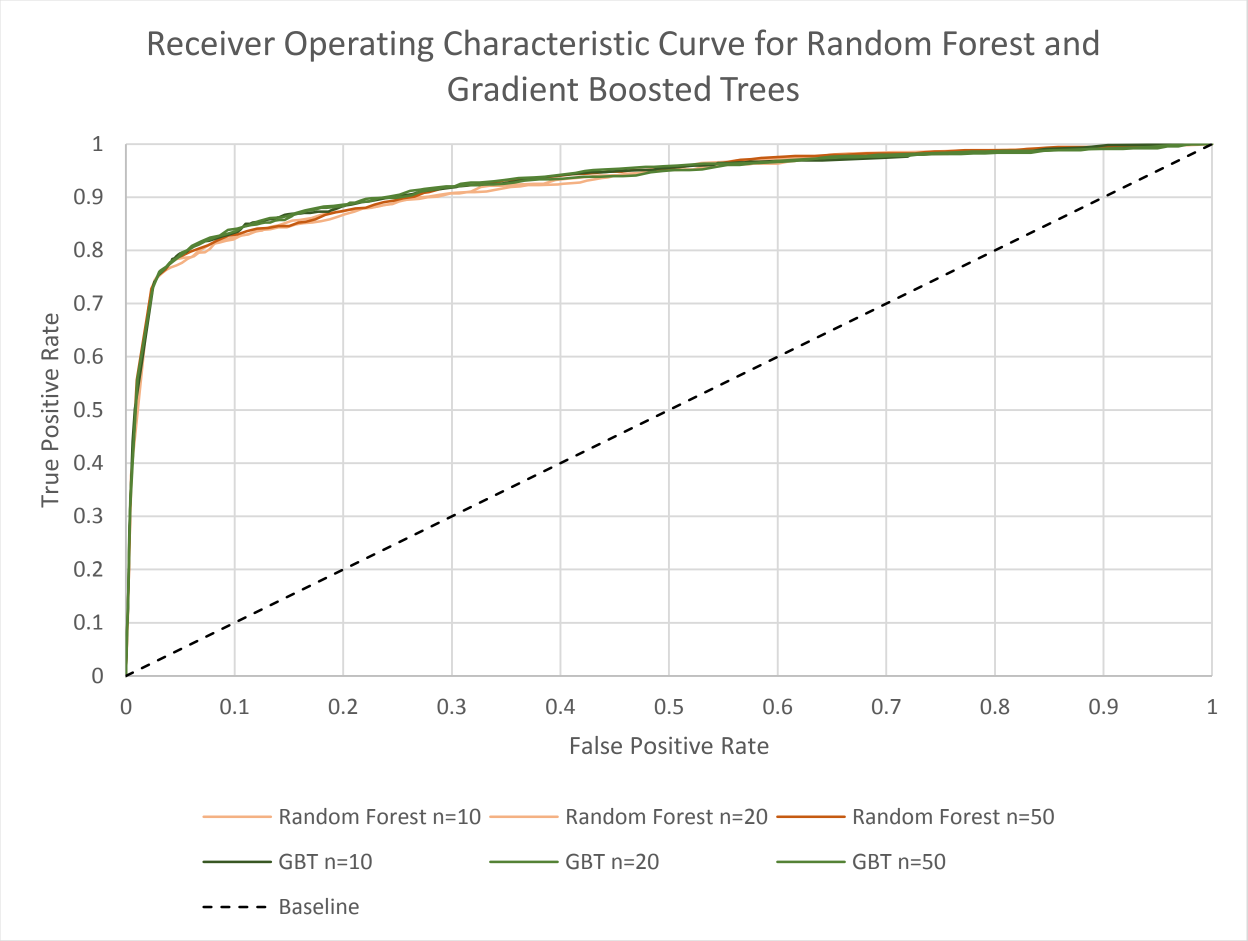

Figure 2. ROC of Random Forest and Gradient Boosted Trees.

The ROCs for Random Forest and GBT were extremely similar, and both appear to have high AUCs, and more importantly in the context of comparison to Logistic and SVM with SGD, higher recall and f-Score. The tables below summarize the statistics for Random Forest and GBT.

Table 2. Results for Random Forest

Statistic	**Random Forest n=10**	**Random Forest n=20**	**Random Forest n=50**
AUC	.8544	.8553	.8560
Precision	.7774	.7825	.7848
Recall	.7348	.7360	.7372
f-Score with beta = 1.0	.7555	.7586	.7603

Table 3. Results for Gradient-Boosted Tree

Statistic	**Gradient-Boosted n=10**	**Gradient-Boosted n=20**	**Gradient-Boosted n=30**
AUC	.8572	.8564	.8575
Precision	.7844	.7821	.7817
Recall	.7396	.7384	.7408
f-Score with beta = 1.0	.7613	.7596	.7607

Both ensemble methods significantly outperform the original models in terms of recall and f-Score. These should be considered extremely important to score well in, as they indicate how well the model actually associates to a real world setting. This is because in an ensemble method, multiple decision trees are part of the final model, making it a very successful machine learning model for classification. Ensembling greatly reduces the risk of overfitting and gives a better result for cohorts with small case proportion like this one. This increases the recall ability of the algorithm. For this reason, the model that was implemented in the final product of the study were those that included ensembling. Each of the models resulted in an AUC in the 0.86 range, slightly higher than that of the TREWScore Algorithm (0.83).

Conclusion

The final models that were chosen included ensembling methods, Random Forest and Gradient Boosted Trees with various numbers of trees. These resulted with AUCs in the range of 0.86 and precision in the range of 0.78. The recall ended in the 0.74 range and f-Score in the 0.76 range. They ended up being slightly higher than TREWScore's AUC of 0.83. The slight improvement is most likely due to the use of ensembling methods, as it can be seen through comparison with non-ensemble methods like Logistic Regression or SVM with SGD Regression that though their AUCs were high, they were not good classifiers. Ensembles, on the other hand, can combine many different iterations to create a final model, which allows it to be more robust.

References

1. K. Henry, D. Hager, P. Pronovost, S. Saria. A targeted real-time early warning score (TREWScore) for septic shock. Science Mag 2016.
2. J. Gardner-Thorpe, N. Love, J. Wrightson, S. Wash, N. Keeling. The value of modified early warning score (MEWS) in surgical in-patients: a prospective observational study. Annals Royal College of Surgeons of England 2014.
3. O. Bayer, C. Hartog, D. Schwarzkopf, C. Stumme, A. Stacke, F. Bloos, C. Hohenstein, B. Kabisch, C. Weinmann, J. Winning. The PRESEP score: an early warning scoring system to identify septic patients in the emergency care setting. Crit Care Med 2014.
4. K. Henry, C. Paxton, K. Kim, J. Pham, S. Saria. REWS: Real-time early warning score for septic shock. Crit Care Med 2014.
5. D. Shavdia. Septic shock: providing early warnings through multivariate logistic regression models. Harvard-MIT Division of Health Sciences and Technology 2007.
6. X. Meng, J.Bradley, B.Yavuz, E.Sparks, S.Venkataraman, D.Liu, J.Freeman, D.Tsai, M. Made, S.Owen, et al., “Mllib: Machine learning in apache spark,” arXiv preprint arXiv: 1505.06807, 2015.

Supplemental Materials

1. Presentation Slides: https://docs.google.com/presentation/d/1wTaLRC94SFC2DH4dhLHGUFuWzXyBRgcI5qF1wumfaDQ/edit?usp=sharing
2. Presentation Recording: http://youtu.be/nsflC_hebmM