Machine Learning Engineer Nanodegree Capstone Project
Intensive Care Unit Mortality Prediction

Andrew Lee
March 2017

I. Definition

Project Overview

The intensive care unit (ICU) cares for the most critical patients who are admitted to the hospital for emergent issues that require immediate action and close management. Commonly requiring intubation and assisted ventilation, patients are vulnerable and their health statuses can change precipitously. Timing of clinical intervention is therefore paramount to attenuating morbidity and avoiding mortality, making mortality prediction an approach for identifying patient risk early enough for the right mortality-preventing intervention. Fortunately, the current paradigm of clinical decision-making in the ICU is supported by temporal data generated by a host of monitoring systems and lab assays, allowing machine learning to potentially solve mortality prediction.

Published literature provides a few examples of successful predictive models that can discriminate the risk of mortality and acute conditions in the ICU setting with high performance 4 [1, 2] and 12 [3] hours in advance. These examples utilize data that are widely accessible to most critical care electronic health records (EHR) produced by monitoring systems. The *AutoTriage* algorithm [3] uses eight clinical variables from electronic health records to predict ICU mortality with an Area Under Receiver Operating Characteristic (AUROC) of 0.88 with a sensitivity of 80% and specificity of 81%.

The dataset that will be analyzed is the MIMIC-III (Medical Information Mart for Intensive Care) dataset is a critical care database containing all ICU admits at the Beth Israel Deaconess Medical Center in Boston, Massachusetts between 2001 and 2012 that contains 57,786 unique admits for 46,476 unique adults. The data includes vital signs, medication, laboratory tests, and much more to illustrate the arc of care during an admit [4].

Problem Statement

The goal of this project will be to predict the risk of mortality in the ICU using temporal frameworks different from those already elucidated by literature. Timing of intervention is extremely important in mitigating the risk of mortality, and thus novel temporal frameworks must be investigated for their utility. Predictors will be collected from the first day of the admit to predict the incidence of death after the first day. The target population will be patients older than 18 years of age with a length of stay greater than one day. 81% of deaths occur after the first day at a slightly higher mortality rate of 11% (versus 10% on the first day). The first-day framework will be examined as a possible solution for healthcare settings that can take advantage of a nightly process that generates predictions for the subsequent day, which may be more adoptable than a real-time solution, like those published in literature.

This project will investigate the utility of decision trees and gradient boosted decision trees to classify the binary target, where 1=death and 0=no death. Decision tree algorithms are particularly well suited for this task since they are easier to interpret. Model interpretability in healthcare is paramount because misclassification costs could entail patient harm and adverse outcomes, so the model will need to be understood and validated against clinical thinking. Moreover, interpretability will be required to build trust with clinicians, who have been trained to be highly skeptical and cautious.

Metrics

Several performance metrics will be used to compare between models, including AUROC, sensitivity, specificity, positive predictive value, and cumulative lift.

- AUROC is a measure of how well a model can discriminate between target classes. Graphically, it is the integral area between a curve constructed as the true positive rate versus the true negative rate and a linear curve representing random chance of discrimination (or 50/50 guessing). The AUROC can be between 0 and 1. A value greater than 0.5 indicates that a model is better at discriminating than random chance.

- Sensitivity is the true positive rate and is calculated as the True Positives divided by the sum of True Positives and False Negatives. It represents the proportion of actual positives that are correctly identified.
- Specificity is the true negative rate and is calculated as the True Negatives divided by the sum of True Negatives and False Positives. It represents the proportion of actual negatives that are correctly identified.
- Positive predictive value (PPV) is the proportion of positive predictions that are correctly identified as true positives. This is important because it reflects the magnitude of false alarms generated from a low PPV.
- Cumulative lift, for this problem, is a measure of how effective the model can concentrate risk according to predicted probabilities when compared against the baseline prevalence of risk.

II. Analysis

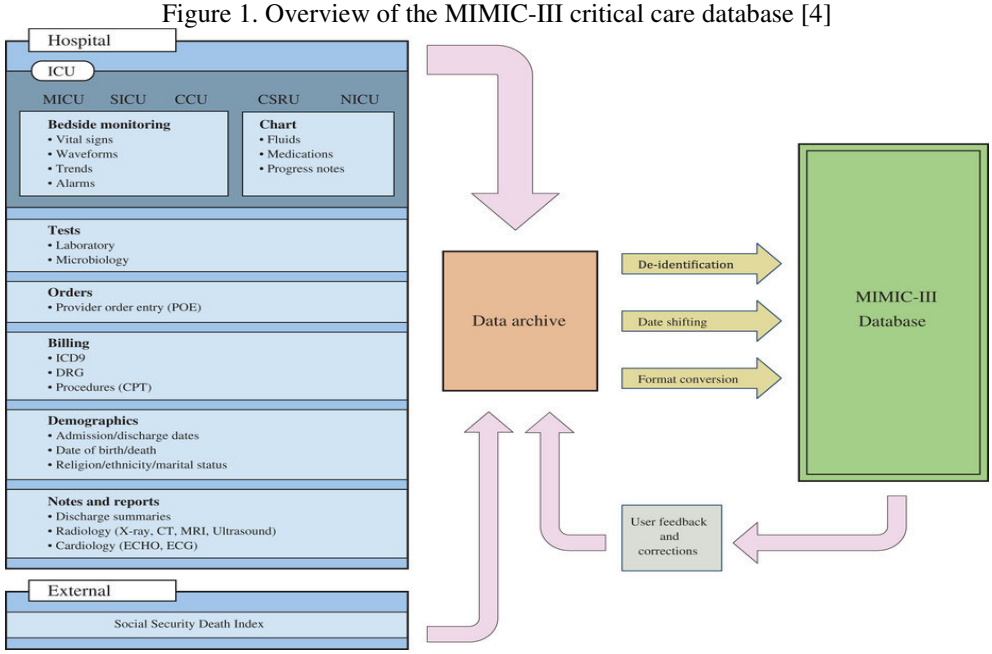
Data Exploration and Visualization

Dataset

The MIMIC-III database is constructed from several domains of data that are collected at different points during a patient's hospital admission. Most important are the time-varying datapoints. Bedside monitoring use real-time sensors to periodically record vital signs (eg, blood pressure, temperature), waveforms (eg, heart beat characteristics), trends, and alarms. Laboratory and microbiology tests are ordered by the physician according to the patient's health condition and are available for decision making hours later. Similarly to laboratories, imaging and tests can be ordered as well. Additionally, orders are recorded for medical supplies, medications, and services. Charts aggregate most of the intra-stay data with clinician notes for a complete picture of status and progress.

Moreover, less time sensitive data are captured before, during, or after the admission that include demographics, billing, and further notes.

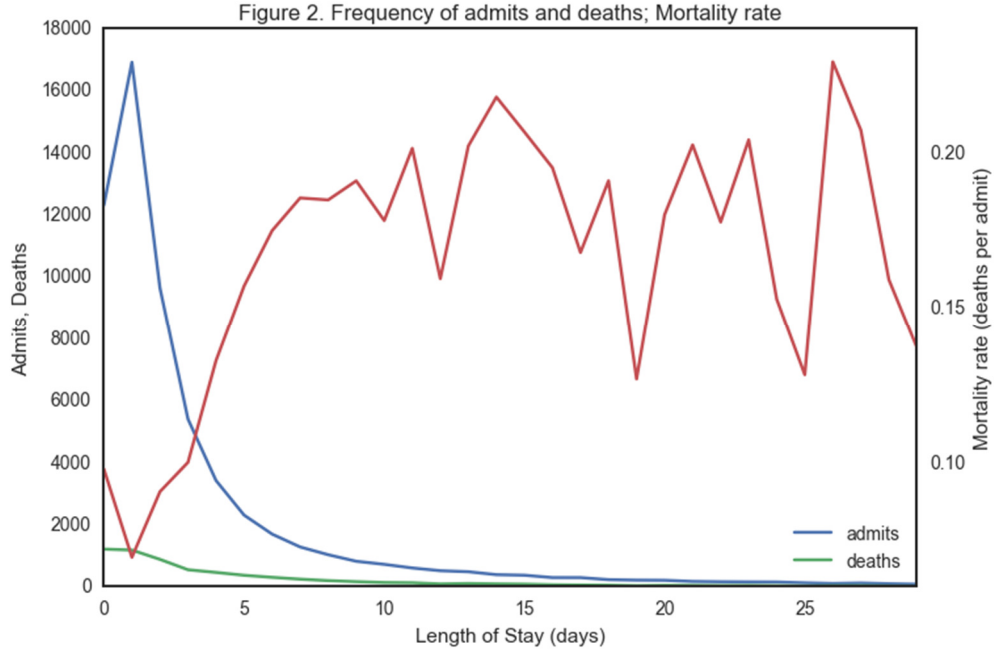
Figure 1. Overview of the MIMIC-III critical care database [4]



Cohort identification

Considering the entire dataset, there are 57,786 unique admits, 61,532 ICU stays, and 46,476 unique patients, yielding 1.2 admits per patient and 1.1 ICU stays per admit. The average length of stay (LOS) is 4.9 days and the mortality rate is 0.11 (deaths per admit). The figure below trends admits, deaths, and mortality rate by LOS.

Figure 2. Frequency of admits and deaths; Mortality rate



18.6% of deaths and 20.5% of admits occur on the first day, representing a mortality rate that is slightly lower than admits with LOS greater than one day. This may partly be due to a higher number of less emergent cases that do not require more than a day of treatment. Excluding first day admits would help concentrate the problem state to higher risk cases. Moreover, only LOS≤30 will be considered since LOS greater than 30 days are more likely to reflect chronic issues rather than acute.

As a large hospital system, Beth Israel has several critical care units, including medical ICU (MICU), surgical ICU (SICU), coronary care unit (CCU), child support recovery unit (CSRU), and neonatal ICU (NICU). Services are also varied, leading to a heterogeneous mixture of patients that are seen.

Table 1. Proportion of admits by critical care service

Service	**% of Admits**
Medical - general service for internal medicine	34.9%
Cardiac Medical - for non-surgical cardiac related admissions	12.9%
Newborn (NB) - infants born at the hospital	12.0%
Cardiac Surgery - for surgical cardiac admissions	11.2%
Surgical - general surgical service not classified elsewhere	6.9%
Neurologic Surgical - surgical, relating to the brain	5.3%
Trauma - injury or damage caused by physical harm from an external source	4.7%
Neurologic Medical - non-surgical, relating to the brain	3.8%
Thoracic Surgical - surgery on the thorax, located between the neck and the abdomen	1.8%
Orthopedic medicine - non-surgical, relating to musculoskeletal system	1.7%
Vascular Surgical - surgery relating to the circulatory system	1.6%
Orthopedic - surgical, relating to the musculoskeletal system	1.2%
Genitourinary - reproductive organs/urinary system	0.5%
Newborn baby (NBB) - infants born at the hospital	0.4%
Gynecological - female reproductive systems and breasts	0.3%
Ear, nose, and throat - conditions primarily affecting these areas	0.3%
Plastic - restoration/reconstruction of the human body (including cosmetic or aesthetic)	0.3%
Observation - not ill enough for a service but kept in hospital for observation	0.1%
Dental - for dental/jaw related admissions	0.0%
Psychiatric - mental disorders relating to mood, behavior, cognition, or perceptions	0.0%

Patients could have multiple ICU stays per admit because they are transferred between units or services. For example, a patient could be admitted for trauma and then be transferred to surgery. For the purpose of maintaining independent observations, prediction will only consider the first ICU stay per admit as well as the first admit per patient.

Some units and services will be excluded to improve homogeneity of the target population. Surgery services, including orthopedic and plastics, will be excluded, assuming that surgeries are more likely to be elective than emergent. Newborns will be excluded since their risk pool is vastly different from that of adults (≥18 years old). Observation, dental, and psychiatric will be excluded due to extremely low samples.

Table 2. Patient exclusions

Initial observations	**61534**
Younger than 18 years old	8200 *13%*
Subsequent ICU stays	15058 *24%*
Surgical service	19374 *31%*
Newborn service	9237 *15%*
Minor service	1167 *2%*
LOS≤1 and >30	13768 *22%*
Total excluded	**42266 *69%***

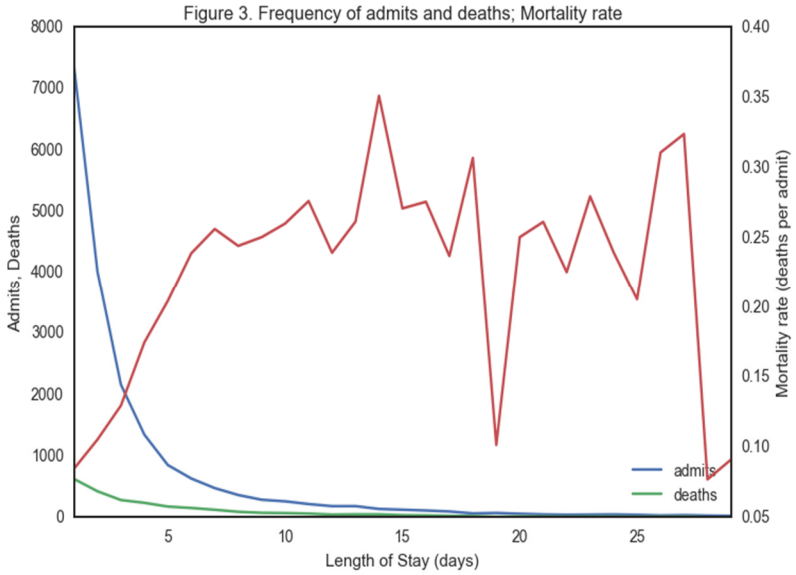


Table 3. Pre- and post-exclusion summary

Exclusion	**Admits**	**ICU stays**	**Patients**	**Admits / patient**	**ICU stays / admit**	**LOS**	**Mortality rate**
Pre	57786	61532	46476	1.24	1.06	4.92	0.11437
Post	19268	19268	19268	1.00	1.00	4.21	0.13618

Admits, deaths, and mortality rate are relatively similar in trend between pre- and post-exclusion. While 69% of ICU stays were excluded, the problem space has been narrowed to align more closely with prediction goals.

Feature domains

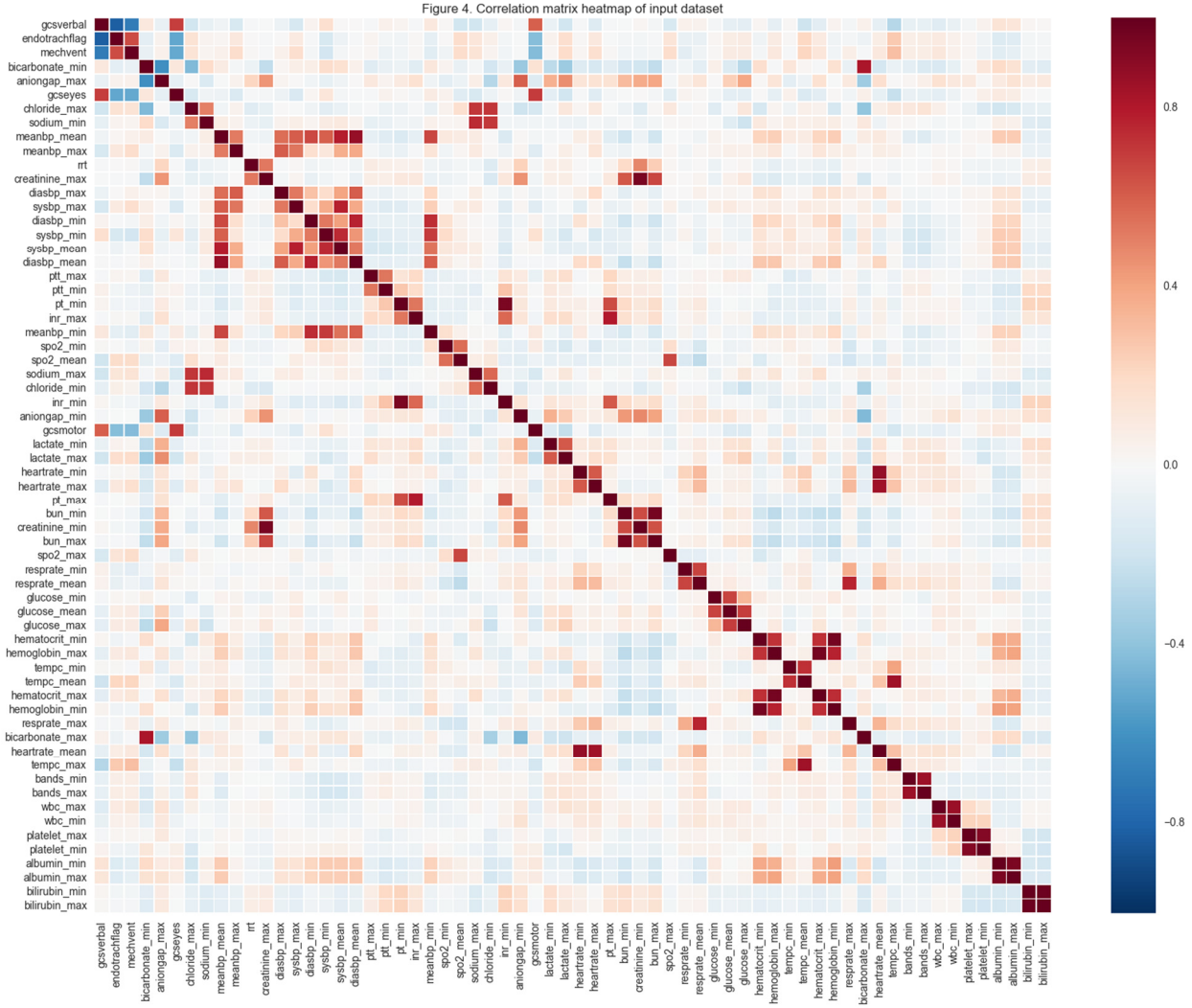
Features are readily available data collected on the first day of the admit and represent patient health status. Time-varying features, such as laboratories and vitals, are minimally altered and offer a wider view of changes across the first day. Average, minimum, and maximum aggregate calculations were applied to flatten many records per patient into a single record per patient and to describe within-patient variance.

Table 4. Feature domains

Feature domain	**Domain description**	**Features**
Glasgow Coma Scale (GCS)	Scoring system for describing the level of consciousness following a traumatic brain injury [5]	5
Laboratories	Blood biomarkers (eg, albumin, hemoglobin, glucose, lactate)	38
Renal replacement therapy	Whether dialysis was given	1
Ventilation	Whether ventilation was used	1
Vitals	Vital signs (eg, heart rate, blood pressure, oxygen saturation)	22

A correlation matrix was created to understand bivariate relationships between features. A heatmap scale was used to visualize the correlation coefficient of each relationship. Figure 4 is an abridged version of the correlation matrix that shows only highly correlated relationships with absolute correlation coefficients greater than or equal to 0.5 and less than 1.0.

Figure 4. Correlation matrix heatmap of input dataset



As expected, features within domains were most correlated with each other. Within the GCS domain, having an endotracheal tube or mechanical ventilation were significantly and inversely related with measures of consciousness (verbal, vision, and motor response).

Anion gap has several relationships that reflect its role in differentiating acidosis and its causes. Albumin and bicarbonate are correlated because of their roles in calculating anion gap. BUN and creatinine are also correlated when acidosis is caused by renal failure. Anion gap also has a role in modulating mechanical ventilation, so it is correlated to metrics related to respiration, like heart rate. It is correlated with glucose when acidosis could be driven by diabetic ketoacidosis, and with lactate when caused by lactic acidosis.

Other groupings of relationships exist for blood pressures (systolic, diastolic), electrolytes (sodium, chloride), renal failure (BUN, creatinine), blood clotting (PT, PTT, INR), and respiration (bicarbonate, lactate, heart rate).

The absence of the target, death, is notable because it does not correlate highly with any feature.

Algorithms and Techniques
Classifiers will be used to predict the binary target of death. Scikit-learn offers popular implementations of decision tree and gradient boosted decision tree classifiers [6] that will be optimized through an iterative process to tune hyper-parameters that maximize performance.

Decision tree classifier (DT)
The Scikit-learn decision tree is a non-parametric supervised learning method. It is simple to unpack and interpret because it requires little data preparation and outputs an intuitive if-then ruleset for segmenting inputs. The algorithm performs well with continuous and categorical features; however, some data missing values must be handled. The Scikit-learn implementation uses an optimized version of the classification and regression trees (CART) algorithm. Generally, the algorithm relies on Gini impurity, a measure of the degree of inequality represented by a set of values, to make discriminating decisions that maximize inequality, or the separation of the target variable.

Additionally, data preparation will be minimal since decision trees can inherently handle scaling and outliers.

Since the target prevalence is highly skewed, the input dataset will need to be sampled to create a balanced dataset such that the target is equally likely to be a 1 or a 0 to prevent the fitted model from becoming biased towards the more common target case.

Gradient boosted decision tree classifier (GB)
Gradient boosting is an ensemble method for combining the predictions from several estimators to improve generalizability and robustness of a base estimator. The base estimator in this case will be a regression tree. The algorithm builds weak learners in succession to iteratively optimize an arbitrary loss function. Each weak learner is only slightly better at discriminating the target variable. After each iteration, observations that were previously inaccurately predicted are reweighted so that the subsequent weak learner can focus on improving predictions where past learners were unsuccessful.

GridSearchCV optimization process
Scikit-learn's GridSearchCV function searches a parameter space for an optimal combination of parameter settings that maximizes a score function with cross-validation. In this project, the parameter space will be composed of the parameters below. Parameter settings will investigate recommendations from Scikit-learn, XGBoost, and Kaggle competition winners.

Decision tree classifier:

- Max_features Maximum number of features to consider when looking for the best split
- Max_depth: Maximum tree depth
- Min_samples_leaf: Minimum number of samples required to be at a leaf node

Gradient boosted decision tree classifier (in addition to decision tree parameters):

- N_estimators: Number of boosting stages
- Learning rate: Rate of change in the contribution of each tree

Benchmark
Calvert's AutoTriage algorithm [3] was developed using MIMIC-III data to predict mortality 12 hours in advance with 5 hours of data consisting of 8 features.

Table 5. Benchmark performance

Performance metric	AutoTriage
AUROC	0.88
Sensitivity	0.80
Specificity	0.81
PPV	0.44

While using the same data source and target, there are some differences in approach. AutoTriage only predicts MICU mortalities with an LOS between 17 and 500 hours. The temporal framework is the largest difference – only 5 hours of data were used to predict 12-hour mortality.

Timing is a crucial determinant of intervention selection, and achieving similar results as those of AutoTriage's would afford another temporal construct worth further research. Whereas AutoTriage focuses more on acute intervention for immediate needs (within 12 hours), this project investigates a time window that could be advantageous for longer-term decision making, such as resource coordination and preventative interventions. Potentially performing better than AutoTriage may motivate more research toward prevention as opposed to reaction.

III. Methodology

Data Processing

Minimal data preparation is needed since decision trees can inherently handle scaling and outliers. Binning will not be needed since splits should naturally find segmentation in the data that binning would otherwise provide. Only missing values need to be processed because Scikit-learn decision trees cannot handle missing values.

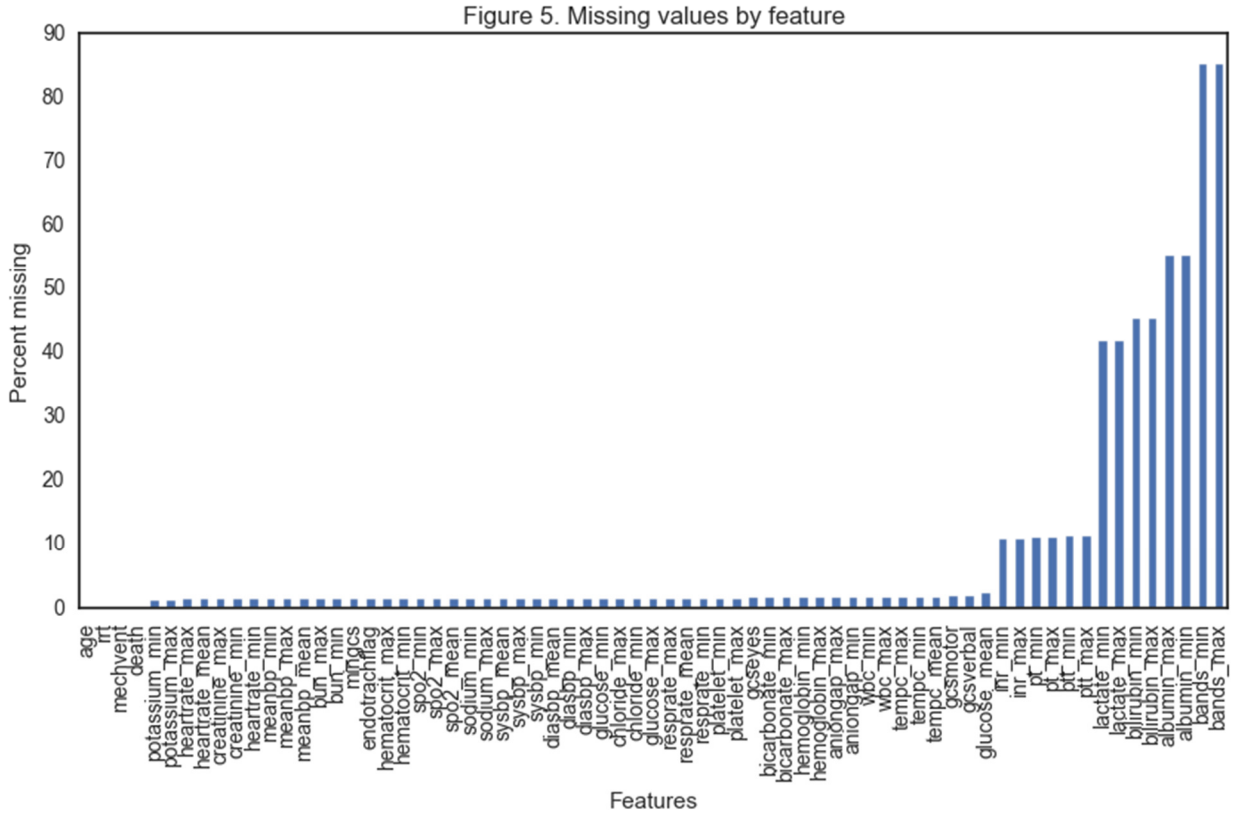
Data aggregation

Diagnostic lab values and sensor data are created with timestamp information as several datapoints for the same feature could exist through time. Data was aggregated through the first day of the hospital stay by applying average, minimum, and maximum functions over the correct time interval.

Missing values

Missing values should be expected since not all conditions are treated the same. Missing values in healthcare should be considered carefully since test should be ordered when necessary; therefore, non-missing values may be biased and fill strategies must consider the implications of using population statistics. The prevalence of missing values was plotted for each feature in Figure 5.

Figure 5. Missing values by feature



Bands is missing most frequently. Bands is the count of immature neutrophils in the blood used to diagnose infections, like sepsis. Sepsis is the primary diagnosis in roughly 5% of admits. Since15% of admits contain a value for bands, then I will assume that this is normal since not everyone should receive a test for bands if it is not necessary. A bands value of greater than 10% indicates likelihood of sepsis, so missing values will be filled with 0%, indicating that there was no indication of sepsis.

Albumin, bilirubin, and lactate are missing in 40-60% of records. These labs are ordered for specific reasons, so non-missing values are biased. Values will be filled with the population min or max in the direction of a favorable outcome to reflect normal outcomes.

All other features are missing in less than 10% of records, so they will be filled with the population median.

Implementation

Helper functions

- *Scoremetrics* tabulates key evaluation metrics for comparison between multiple candidate models and includes Scikit-learn's roc_auc_score, precision_score, and recall_score scorers, as well as a helper function for specificity.
- *Specificity* calculates specificity by extracting true negatives and false positives from the confusion matrix of a classifier's prediction.
- *Cvscore* performs cross validation with the trained model and 10 folds, then calculates the mean and standard deviation of all iterations.
- *Visualizetree* generates visual tree diagrams using pyplot and DecisionTreeClassifier's tree_ attribute. *Visualizetree* opens fitted models for interpretation.
- *Visualizeimportance* generates a bar chart of Gini importance from classifiers' feature_importances_ attribute. Visual representation allows for better understanding of features' relative contribution to predictions.
- *Cumulativelift* generates the data to for charting cumulative lift, which is the relative improvement in correctly identified predictions versus baseline, or random discrimination. Cumulative charts will be used extensively for comparing performance between models and for choosing a final model.

Data

19,268 observations were used for model development and split into 60% training and 40% testing samples. A higher proportion of test samples was chosen to ensure enough target examples were available for training and testing sets since the target prevalence of 13.6% was close to being a rare event (≤10%). The target prevalence in the training and testing sets were 13.5% and 13.9%, respectively.

Benchmark models

Benchmark models consisting of mostly default parameters were created for each candidate classifier type. DT default settings were highly prone to overfitting. Without pruning, the default max_depth=5 led to overly complex trees, so max_depth=3 was set to gain a better understanding of baseline performance. Additionally, class_weight='balanced' was used to balance the target class weighting since the target class is heavily biased towards the negative class. GBDT settings were not adjusted for benchmarking because they are generally accepted as robust starting points; however, GBDT was fit using sample_weight equal to the proportion of positive cases in the training set to balance the target class.

Figure 6 shows the training versus testing performance across the primary evaluation metrics. Ideally, training and testing should perform similarly, with testing usually performing slightly less; thus, markers on should appear reasonably around the reference line between [0,0] and [1,1]. Performance appearing far away from the reference line, signaling a large discrepancy between training and testing would otherwise indicate overfitting.

Both benchmarks performed reasonably well with AUROC between 0.63 and 0.70. DT had higher specificity, but lower specificity than GBDT. GBDT had particularly high specificity and PPV. Figure 7 compares the cumulative lift of the test samples between models. GBDT was clearly more successful at concentrating mortality risk among

patients. GBDT was 4 times more likely to accurately identify patients in the 90-99th percentiles of propensity than baseline and 1.3 times more likely than DT. Moreover, DT seemed to perform sub-optimally in the top percentiles of its cumulative lift curve given that it broke from a monotonically increasing trend.

Figure 6. Benchmark model performance

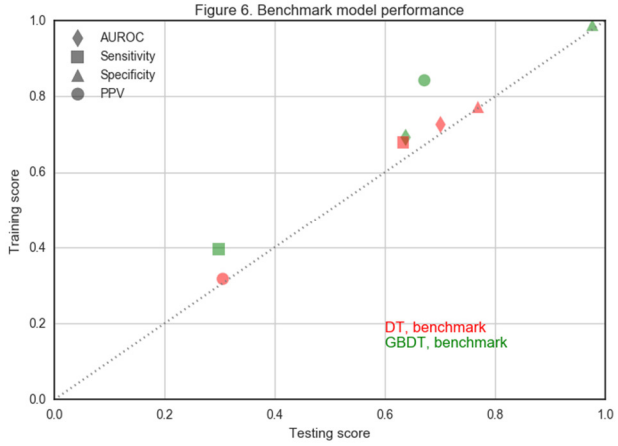
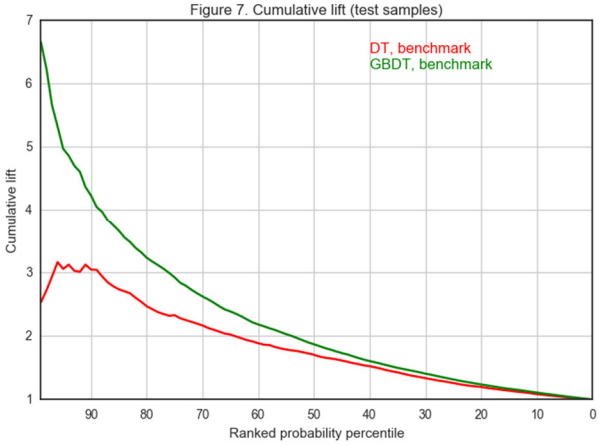


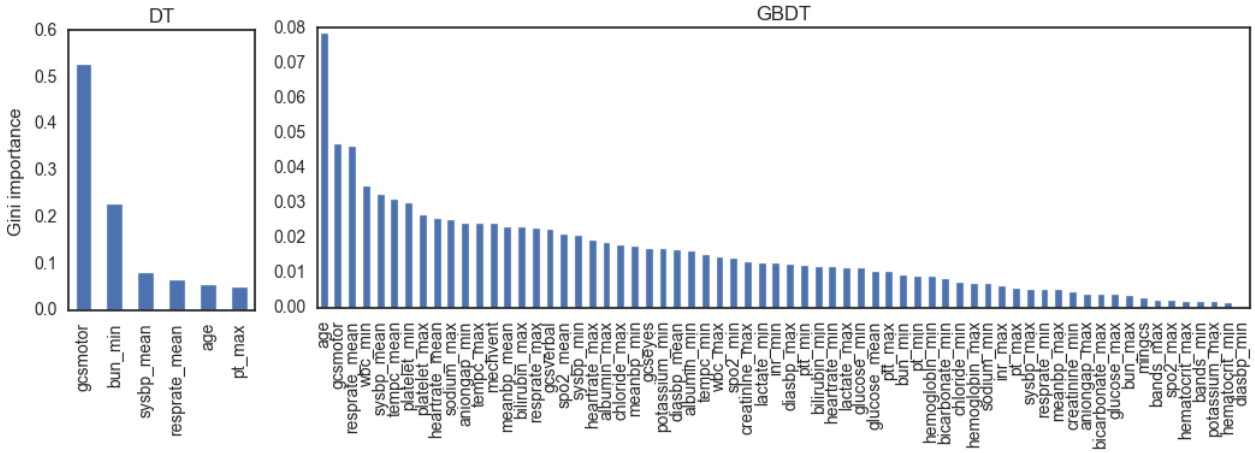
Figure 7. Cumulative lift (test samples)



Cross validation was performed on the fitted models using test data, with 10 folds, and AUROC scoring. The mean AUROC and standard deviation for all iterations were 0.78 ± 0.04 and 0.86 ± 0.03 for DT and GBDT, respectively. There were no outliers in performance. Fitted models were stable. (The difference in AUROC calculated using roc_auc_score and cross validation's roc_auc scoring is that the former calculation is impacted by set thresholds whereas the latter is calculated with the predicted probabilities and no thresholds. ROC charts will be addressed in the Results section.)

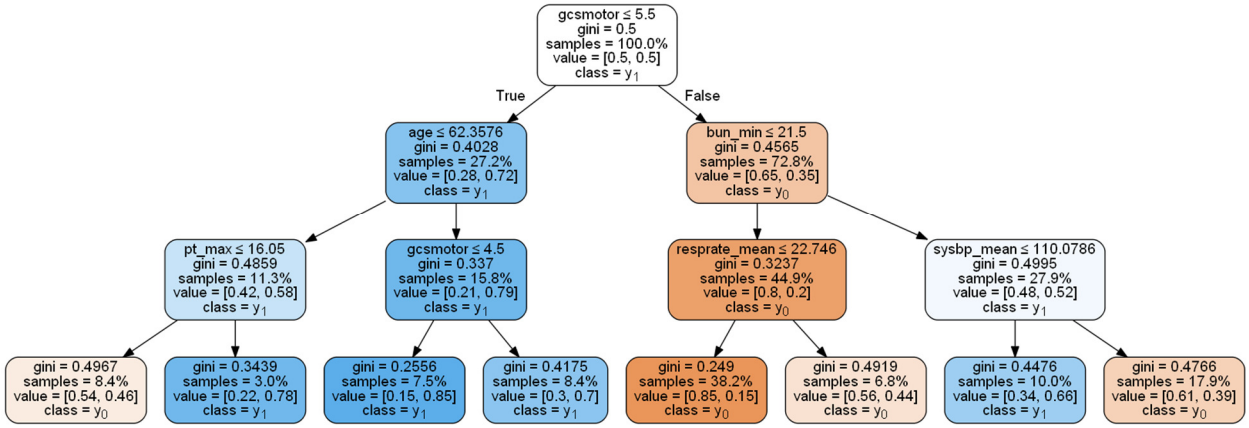
There was generally good agreement in the selected features. GCS motor response, age, mean respiration, and mean systolic blood pressure appear at the top of both models.

Figure 8. Feature importance of DT and GBDT (different scales)



The default max_depth=3 restricted the max number of unique features in the fitted decision tree to seven features. The most important variable, GCS motor response, indicates the level of motor response to pain in trauma patients. It is split on 5.5 on a 1-6 scale (higher is favorable), separating patients that feel some pain from no pain. Age could be indicative of frailty, comorbidity, access to healthcare, employment status; but generally in this context, older age may be more associated with the risk of death. Minimum BUN, mean systolic blood pressure, mean respiration, and pt_max capture some of the most leveraged metrics in managing critically ill patients. Ultimately, the selected features were reasonable and logical.

Figure 9. Decision tree benchmark



Refinement

DT hyper-parameter tuning

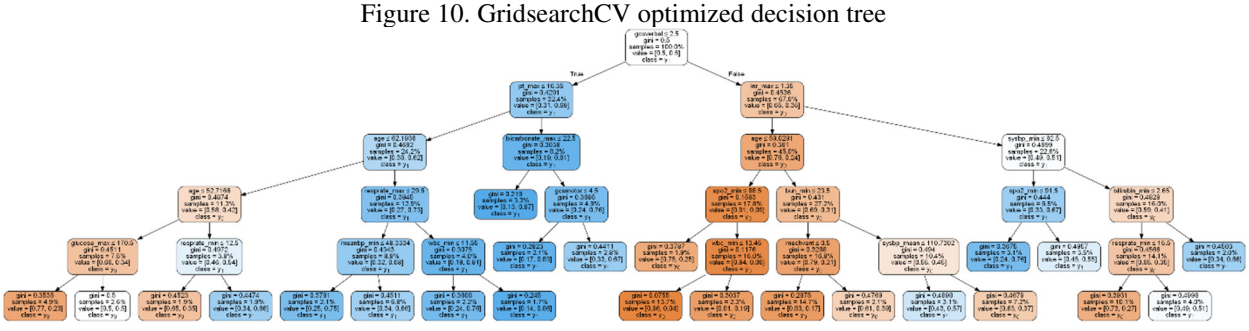
A parameter grid was created for max_depth, max_features, and min_samples_leaf to manage complexity and generalizability while maximizing AUROC.

Table 6. DT parameter space

Hyper-parameter	Parameter range	Selected setting
Max depth	2-5	5
Max features	1-9	6
Min samples leaf	150-400, every 25	200

The optimized tree was fit with max_depth=5, max_features=6, and min_samples_leaf=200. Selected features and splits were reasonable. The branch that led to the majority of positive predictions reflected older, frailer, and less responsive patients. A lower bound of 150 samples per leaf was used to prevent the classification of patient groups smaller than ~1% of the total population. A tree with max_depth=6 was considered. The tree was reasonable but much more complex with marginal AUROC gain, thus max_depth=5 was chosen.

Figure 10. GridsearchCV optimized decision tree



The optimized model had a slight AUROC improvement of 0.01. Sensitivity and specificity were inverted, resulting in higher sensitivity and lower specificity. Improvement in sensitivity led to a marked improvement in the 90-99th percentile of cumulative lift. The optimized model began performing similarly to the benchmark after the 70th percentile of patients.

Figure 11. Model performance

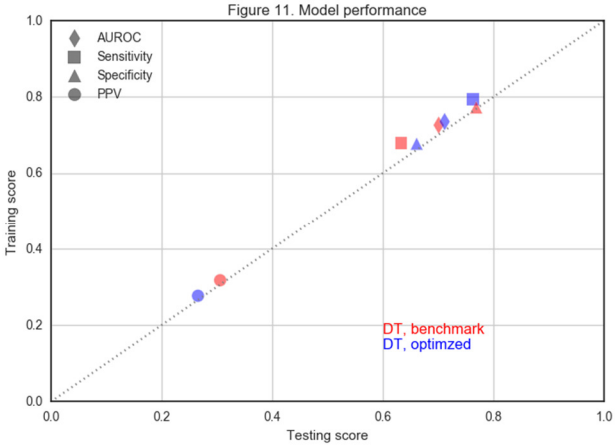
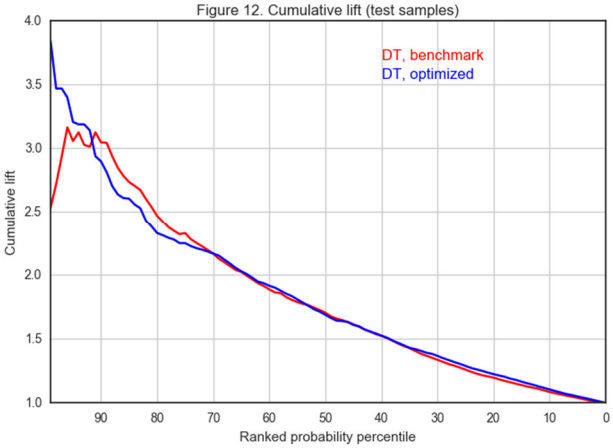


Figure 12. Cumulative lift (test samples)



Cross validation performed on the optimized model with test data and 10 folds yielded a mean AUROC of 0.77 ± 0.05, which was slightly less stable than the benchmark model.

GBDT hyper-parameter tuning

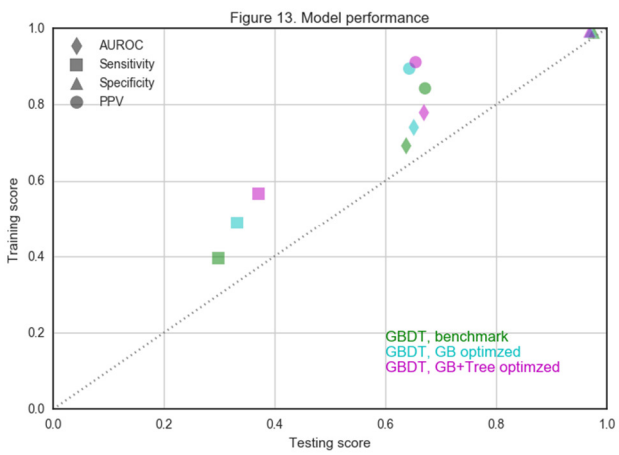
A parameter grid was created for learning_rate, n_estimators, max_depth, max_features, and min_samples_leaf to manage complexity and generalizability while maximizing AUROC. Grid search was implemented in two stages: first, with gradient boosting parameters, learning_rate and n_estimators, then with the remaining tree parameters. Not only did this improve processing time, but learning rate and estimator number are inversely related and should be adjusted together.

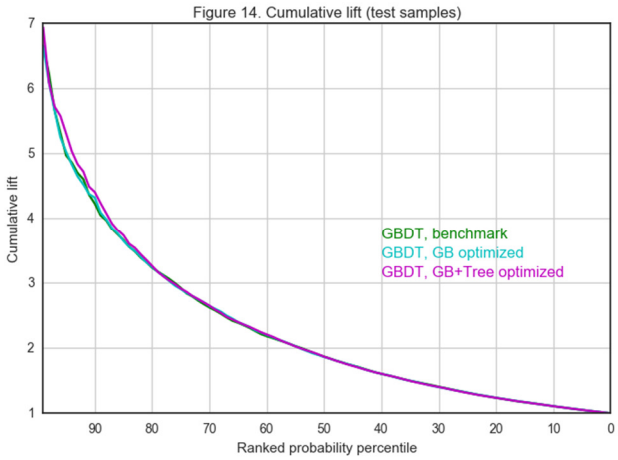
Table 7. GBDT parameter space

Hyper-parameter	Parameter range	Selected setting
Learning rate	0.01, 0.025, 0.05, 0.1, 0.15, 0.2	0.05
Estimators	100-1000, every 100	400
Max depth	2-5	6
Max features	1-9	6
Min samples leaf	150-400, every 25	200

The optimized GBDT was fit with learning_rate=0.05 and n_estimators=400, which were considerably divergent from the default 0.1 learning rate and 100 estimators, but in line with winning Kaggle GBDT schemes. Selected tree parameters were max_depth=6, max_features=6, and min_samples_leaf=200. Gradient boosting and tree parameters were additive to benchmark performance; however, the training-testing discrepancy in evaluation metrics was wide.

AUROC improved in both tuning stages from 0.64 to 0.65 in the GB tuned model and to 0.67 in the GB+Tree tuned model because of improvements in sensitivity. Specificity and PPV minimally worsened. Cross validation of test data showed stable results in both tuning stages with an AUROC of 0.86 ± 0.03.



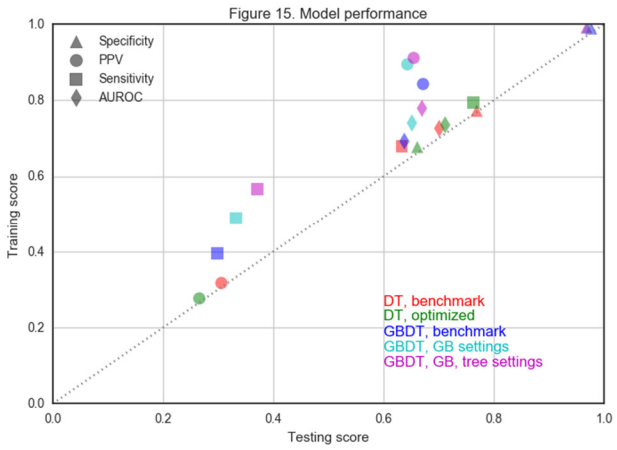


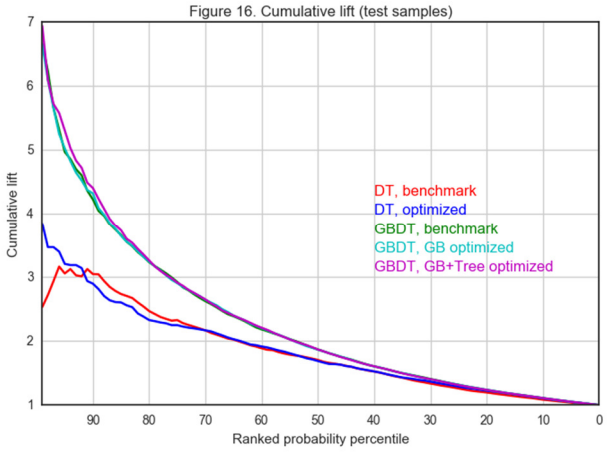
IV. Results

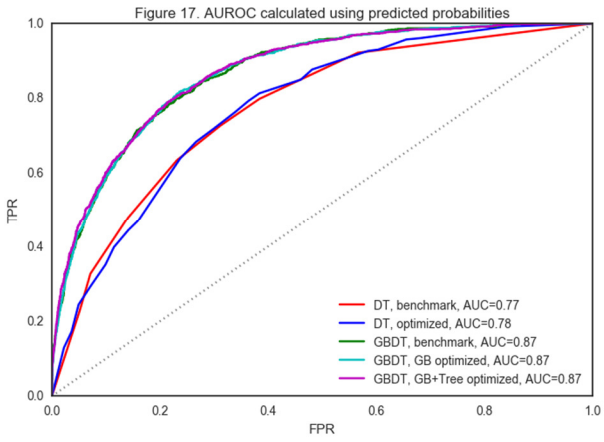
Model Evaluation and Validation

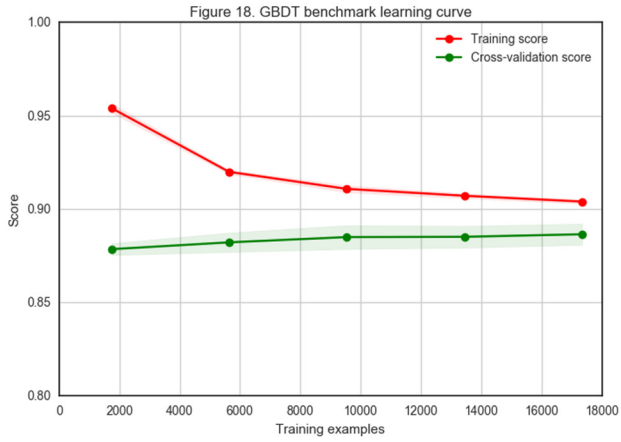
All candidate models were plotted together in Figure 15 and 16. ROC curves were calculated from predicated probabilities to plot the ROCs without thresholds in Figure 17. GBDT clearly outperformed DT in all evaluation metrics. With real-world implementation in mind, GBDT would more accurately discriminate target classes while effectively concentrating mortality risk to high-risk patients. Cumulative lift performance between 7X and 4X in the top 10% of propensity scores would prove to be highly valuable.

Overall, GBDT models were equivalent and differed slightly in specific regions of propensity probabilities. The benchmark model performed slightly better in the first 20% of propensities with higher AUROC and cumulative lift, and reached equivalency in the subsequent 80% of propensities. The benchmark was also the most model; primarily, it used 100 estimators as opposed to 400 and a max depth of three instead of six. Additionally, its learning curve showed that the fitted model generalized well to new data and learned the data's complex concepts since incremental training data would have only marginally increased accuracy.

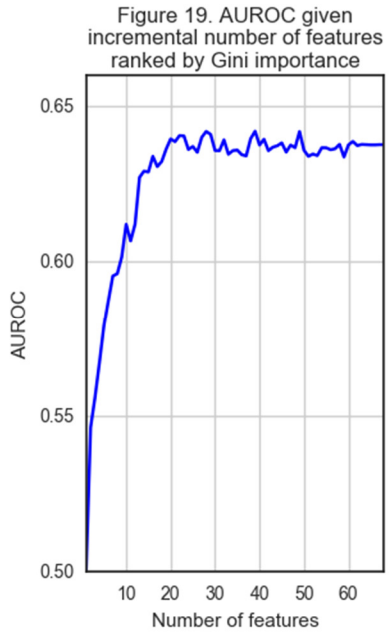


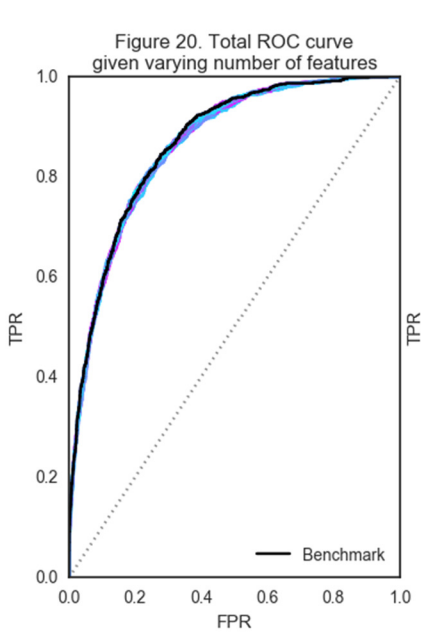






The benchmark model was studied further to understand the utility of all features. The assumption was that some utilized features were picked because the algorithm had to select a feature if stopping criteria had not been met, and each pick came from a random set of features, which sometimes included features that were actually sub-optimal for splitting. A sensitivity analysis was performed by iteratively including features from the most important feature (determined by the benchmark model's feature importances) to the least important feature to understand if there could be an optimal set of features whereby subsequent features would not add any information that leads to better prediction, thus yielding a more parsimonious solution.





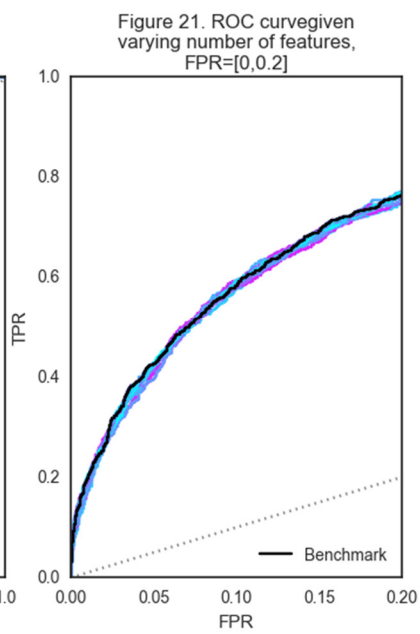


Figure 19 shows that AUROC steeply increases between 1 and 20 features, and maximizes with 39 features. Figure 20 plots the ROC curves for models built with 20 to 65 features (every 5 shown) compared with the benchmark model, which uses 68 features. Figure 21 shows the same ROC plot for FPR=[0,0.2] to magnify differences between the models.

Analysis of cumulative lift, sensitivity, specificity, PPV, and cross-validation confirm that the 20-feature model performs nearly exactly to the 68-feature model. Grid search on the 20-feature model yielded the same gradient boosting and tree parameters, which once applied produced performance that was exactly the same as those of the

68-feature model. The 20-feature GBDT model with default parameter settings was the most parsimonious model without performance trade-offs, and is therefore the final model.

The model will be referred to as ParsiBoost (parsimonious gradient boosted decision tree) henceforth.

Justification

ParsiBoost outperforms AutoTriage in terms of its discriminative ability as evidenced by similar AUROC and higher PPV; however, could be considered less balanced in terms of sensitivity versus specificity. Ultimately, how the final model compares to Autotriage depends on how each model will be applied given the constraints of the problem. For example, finite ICU resources may limit the number of identified high-risk patients to be intervened, thus patients must be prioritized from highest to lowest risk. ParsiBoost performs exceptionally well in risk stratification—high-risk patients are seven times more likely to die than low-risk patients. The final model could be adjusted such that a probability cutoff is used to classify all patients above that threshold as high-risk to yield a total number of positive outcomes in line with resource capacity. Unfortunately, Calvert *et al*. did not provide cumulative lift data for comparison.

In addition to the cumulative lift, Parsiboost's good PPV performance is valuable because fewer false positives would lead to fewer wasted resources, which in an operationally-stressed environment like the ICU is a deciding factor for whether a model such as Parsiboost is implemented. Fewer false alarms also helps to minimize clinician alarm fatigue, which is a large problem in the ICU because many alarms are generated by the array of monitors for each patient.

Table 8. Performance comparison

Performance metric	AutoTriage	ParsiBoost
AUROC	0.88	0.87
Sensitivity	0.80	0.29
Specificity	0.81	0.98
PPV	0.44	0.67

Mortality predictions by both models are distinct due to differing temporal constructs underlying the features (first-day versus last five hours) and mortality event (mortality anytime 2-30 days from admit versus 12 hours from prediction), and would be applied in different ways.

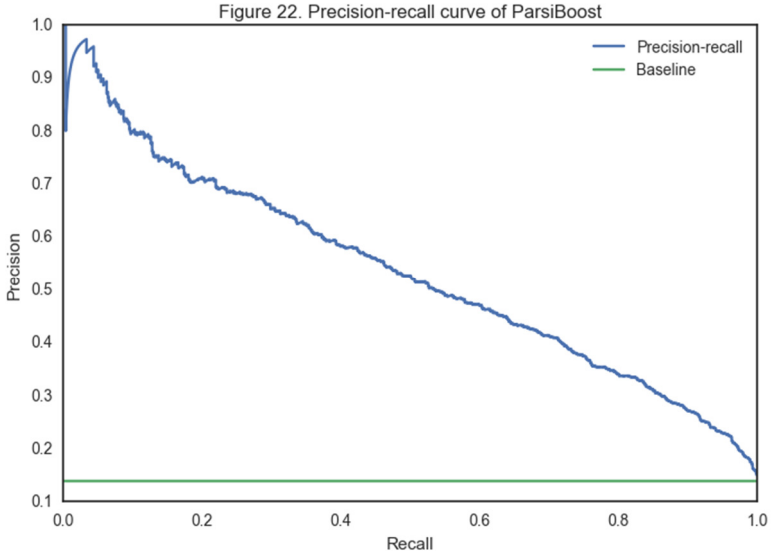
AutoTriage would run in real-time to coordinate resources to attenuate impending events, such as cardiac or respiratory arrests, that require immediate action in highly chaotic situations. The occurrence or severity of these situations could be mitigated with the five hour window afforded by AutoTriage. Conversely, ParsiBoost predicts a mortality event in days 2 through 30 of the stay, which could be used for longer-term planning in the ICU and beyond the ICU. For example, the model could be used to forecast rest-of-hospital resource demand when ICU patients transfer to the ward. Furthermore, some ICUs may experience gross under-resourcing and would benefit from having both models.

V. Conclusion

Free-form Visualization

Figure 22 plots the precision-recall curve of ParsiBoost to highlight the relationship between its favorable PPV (precision) and unfavorable sensitivity (recall) performance. This curve is especially important for model calibration prior to deployment because it determines the trade-off between the proportion of true positives in the prediction set with the proportion of true positives in the population. Improving PPV is important for minimizing false alarms; however, the consequence is a reduction in the total opportunity of potential true positives that will be captured by the purview of the model.

Figure 22. Precision-recall curve of ParsiBoost



Reflection

ParsiBoost successfully represented mortality risk as declining trends in consciousness and homeostatic biomarkers using 20 commonly collected data variables in the ICU. Univariate and bivariate relationships between the initial set of 69 variables were considered. Decision trees helped to elucidate the multi-factorial patterns of features that led to mortality in an easily interpretable format. Once the rulesets generated by the decision tree benchmark was understood and accepted, gradient boosting decision trees were developed that extended predictive capability through a more advanced approach. Parameters in decision tree and gradient boosting decision trees were tuned in parallel using grid search. Several tests of robustness were applied, including learning curve analysis, cross-validation, and attempts to reduce complexity. Finally, sensitivity analysis determined that a 20-feature model with default parameter settings would be as effective as the grid searched model with 68 features.

Interestingly, first-day aggregations of features worked well for an outcome window as wide as 2 to 30 days, which would have captured considerable noise that could have obstructed some predictions. This project demonstrated that a first-day temporal construct could be meaningful in predicting mortality and provided another approach for handling time-varying information. It also showed that gradient boosting, a contemporaneous and less applied algorithm in the real-world, could substantially augment performance beyond decision trees, making a strong case for further research in applying gradient boosting in healthcare. This may be feasible since decision trees are becoming more prevalent as decision support in clinical settings due to their interpretability, and gradient boosting may be sufficiently proximal in concept that it could experience the same adoptability that decision trees have demonstrated.

Improvement

It could be argued that the utility of predicting mortality events as far into the future as 30 days may not be as useful as predictions a few days into the future given the rapidity of change in ICU settings. The problem space could be investigated further to understand with what definition of the target does performance become optimized, and at what time interval between prediction and event could utility be maximized for end-users. ICUs vary in patient population, available services, access to healthcare, and economics; so, it would be reasonable to assume that end-users have differing needs that could benefit from multiple versions of ParsiBoost. It would also be interesting to understand if ParsiBoost could be run in real-time on 24 hours of data to provide predictions commensurate with the progression of disease.

Information about the varying needs of different end-users would be helpful for calibrating the model further. For example, knowing the exact issues that afflict ICUs and ICU staff roles and the associated economics could determine where to set probably cutoffs in the model versus operational capacity, resources, and the costs of false positives and false negatives.

VI. References

1. Calvert, J. High-performance detection and early prediction of septic shock for alcohol-use disorder patients. 2016. http://www.sciencedirect.com/science/article/pii/S2049080116300413.
2. Desautels, T. Prediction of Sepsis in the Intensive Care Unit With Minimal Electronic Health Record Data: A Machine Learning Approach.
   2016. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5065680/.
3. Calvert, J. Using electronic health record collected clinical variables to predict medical intensive care unit mortality. 2016. http://www.annalsjournal.com/article/S2049-0801(16)30131-5/pdf.
4. Johnson, A. MIMIC-III, a freely accessible critical care database. 2016.
   http://www.nature.com/articles/sdata201635.
5. Glasgow coma scale. http://www.brainline.org/content/2010/10/what-is-the-glasgow-coma-scale.html
6. Scikit-learn: Machine Learning in Python. Pedregosa *et al*. JMLR 12, pp. 2825-2830, 2011.
7. Ensemble methods. http://scikit-learn.org/stable/modules/ensemble.html