



Predicting Sepsis

Team Four

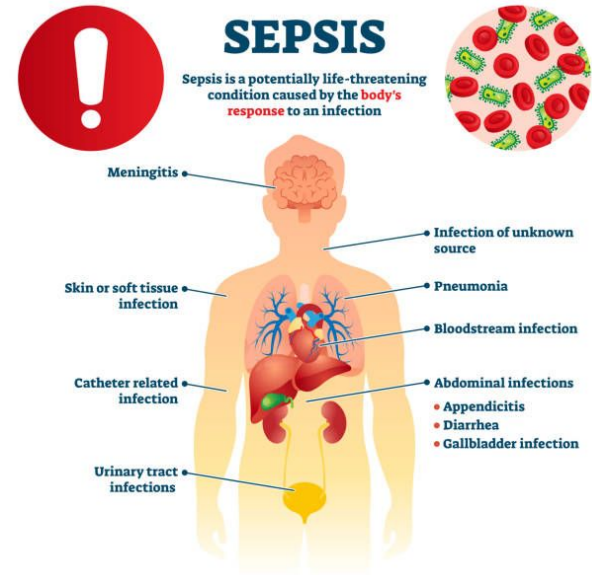


BOSTON
COLLEGE

By Alesiada Baka, Maxwell Davidoff, Song Lu, and Alice Tang

I. Introduction

- Sepsis is a **life-threatening condition** that arises when the body's response to an infection causes injury to its own tissues and organ.
- ~ 11 million deaths in the world, ~1.7 million annually in the U.S.
- By specifically implementing our sepsis detection algorithm in the U.S. we can potentially:
 - Save **19,210 lives** annually
 - Free up **3.4 million bed days** per year
 - Generate **\$10.476 billion** in cost savings annually
- How do we detect it?

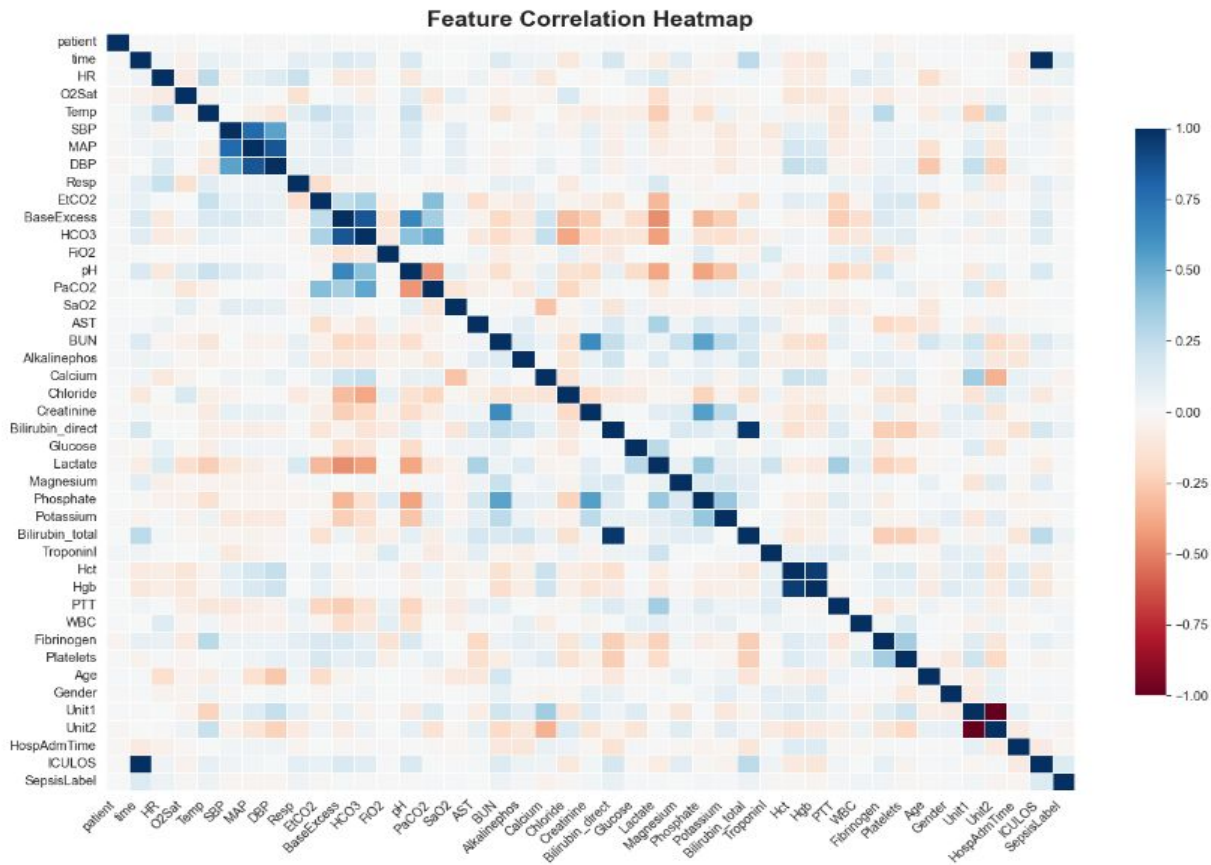


Our algorithm could save 19,000+ lives, free 3.4 million bed days, and save \$10.476 billion annually in the U.S.

Table of Contents

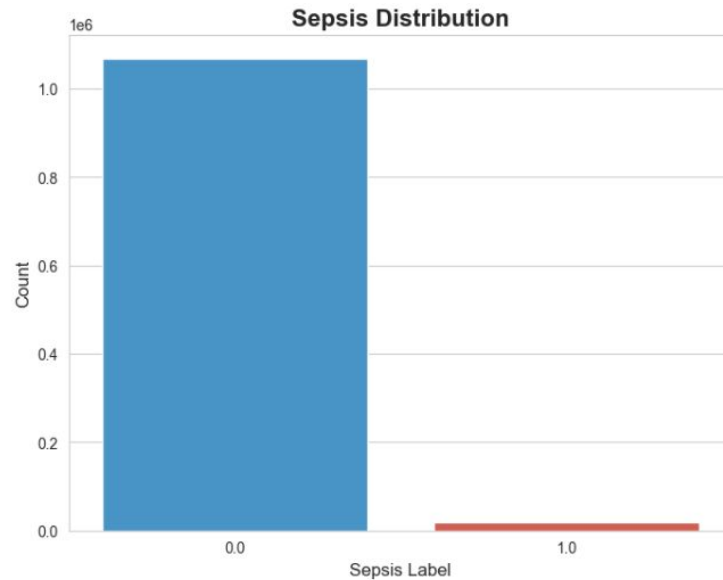
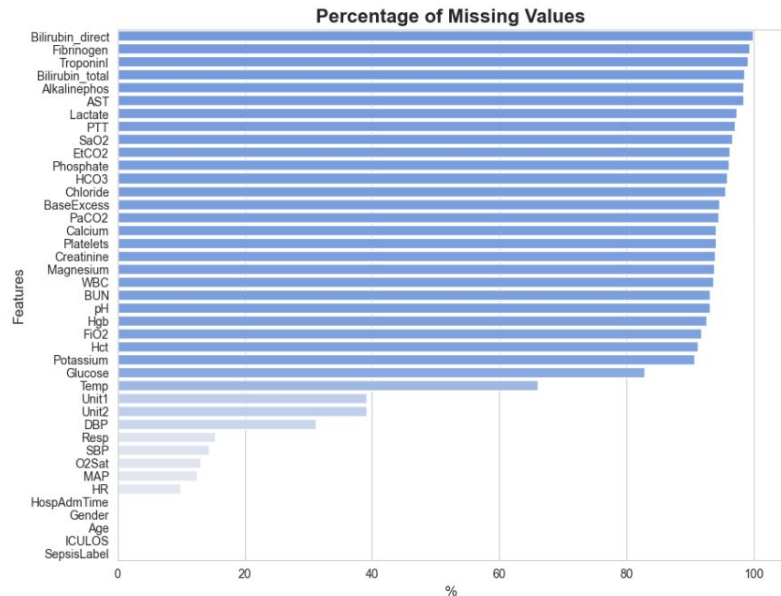
- I. Introduction
- II. Exploratory Data Analysis
- III. Handling Missing Data
- IV. Handling Class Imbalance, Feature selection, Isolation Forest
- V. Model Selection
- VI. Model Results
- VII. Key Takeaways and Business Implications

II. Exploratory Data Analysis (1)



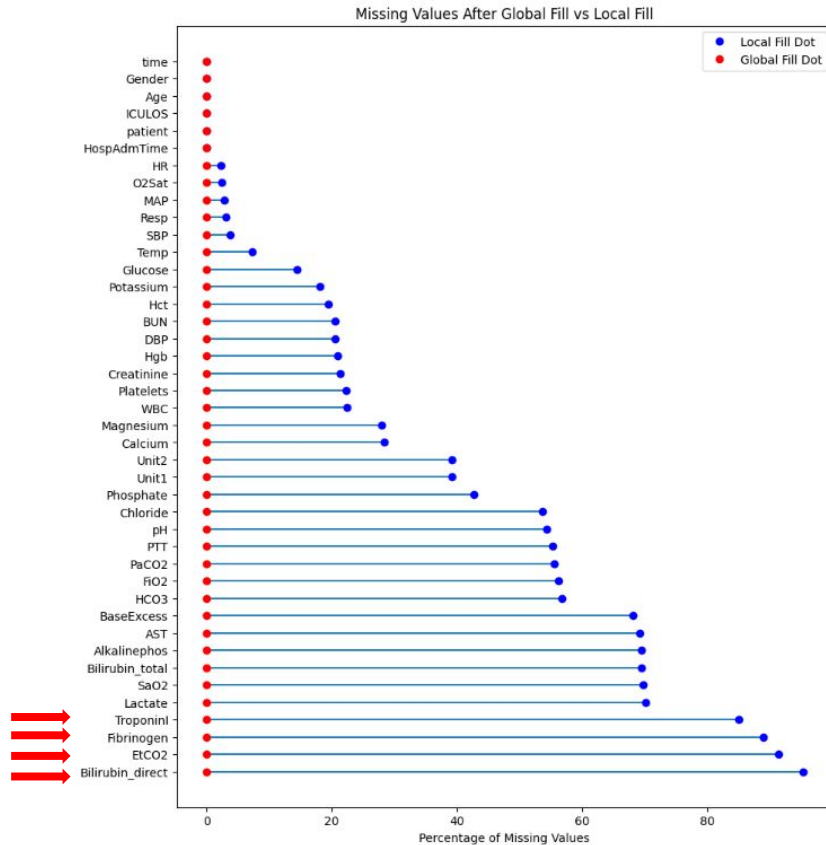
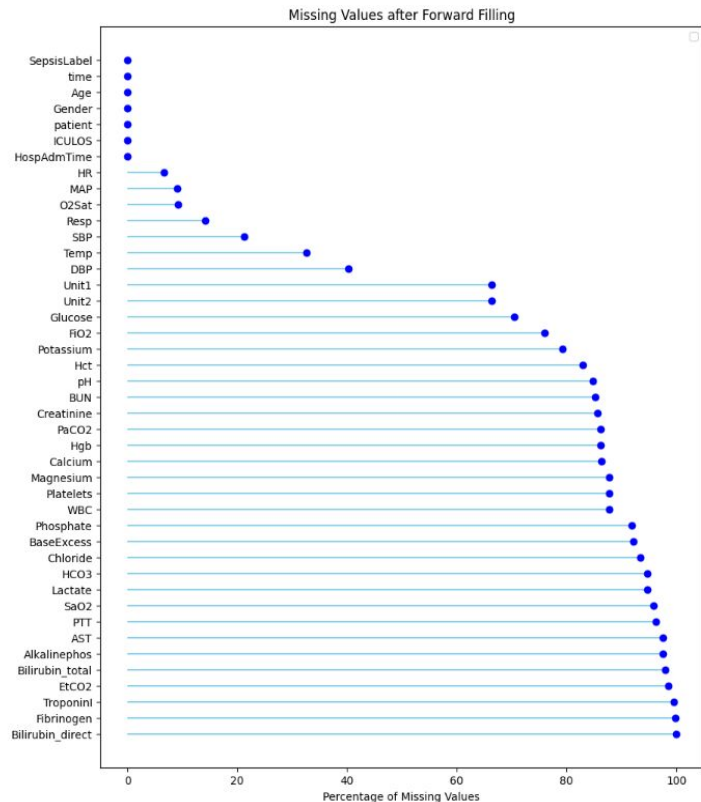
No strong predictors of sepsis.

II. Exploratory Data Analysis (2)



Data had two issues: significant amount of missing data & large class imbalance.

III. Handling Missing Data



Forward fill and global fill allowed us to effectively clean the dataset and eliminate missing values.

IV. Handling Class Imbalance, Feature selection, Isolation Forest (2)

THint (Temperature-Heart Rate Interaction)

Reasons: The product of Temp (temperature) and HR (heart rate) can capture the interaction between these two vital signs. In medical contexts, certain conditions might cause both temperature and heart rate to rise or fall together.

RHratio (Respiratory Rate-Heart Rate Ratio)

Reasons: The ratio of Resp (respiratory rate) to HR (heart rate) can provide insights into the balance between these two vital signs. In sepsis, the body's response to infection can affect both respiratory and heart rates. The ratio can highlight abnormalities in this balance, which might be indicative of sepsis.

By creating these new features, we can provide the models with a more nuanced view of how these variables relate to each other in the context of sepsis.

THint and RHratio show interactions between Temp and HR, and Resp and HR.

IV. Handling Class Imbalance, Feature selection, Isolation Forest (4)

Validation Data

- In our project, we split the training data into **85% training** and **15% validation** sets based on **unique patient IDs**. This ensures that the model is evaluated on patients it hasn't seen during training, providing a more accurate assessment of its performance. Here are the benefits:
 - **Tune hyperparameters of the models**
 - **Evaluate the impact of feature engineering and preprocessing steps.**
 - **Ensure that the model is not overfitting to the training data.**

```
unique_patients = sepsis_train_final['patient'].unique()
```

```
train_patients, val_patients = train_test_split(unique_patients, test_size=0.15, random_state=10)
```

```
X_train = sepsis_train_final[sepsis_train_final['patient'].isin(train_patients)]
```

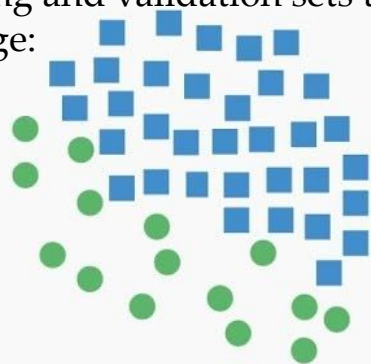
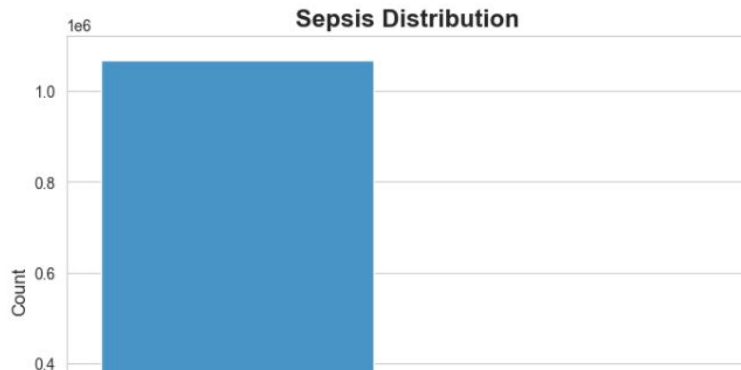
```
X_val = sepsis_train_final[sepsis_train_final['patient'].isin(val_patients)]
```

Split the test data into 85% training and 15% validation sets.

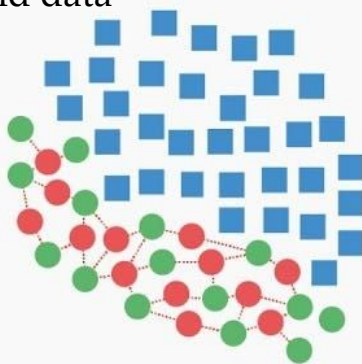
IV. Handling Class Imbalance, Feature selection, Isolation Forest (1)

Synthetic Minority Over-sampling Technique (SMOTE)

- **SMOTE** generates new synthetic samples by interpolating between existing minority class samples. This helps to create more diverse and informative training data.
- After applying **SMOTE**, we ensured that there were no common patients between the training and validation sets to avoid data leakage:



Original Dataset



Generating Samples



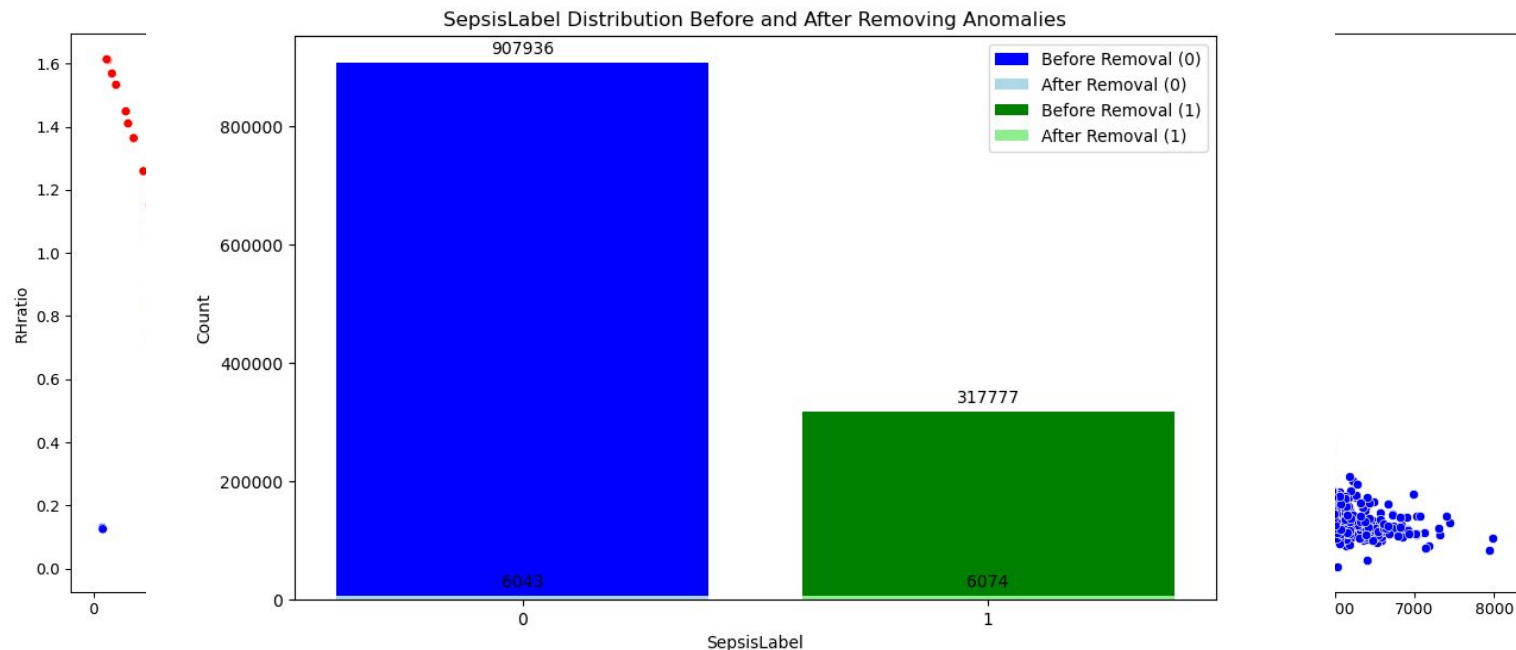
Resampled Dataset

SMOTE generates new synthetic samples by interpolating between existing minority class samples.

IV. Handling Class Imbalance, Feature selection, Isolation Forest (3)

Isolation Forest

- Anomaly Detection:** By setting the **Contamination** to 0.01, our Isolation Forest identified **12,118 anomalies (outliers)** and **1,199,598 normal data points** in the training set. A small portion of the training data (about 1%) was considered anomalous and removed to improve the quality and performance of the model.



Removed 12,118 anomalies from 1,199,598 normal data points in the training set.

V. Base Model Selection and Development (1)

LightGBM:

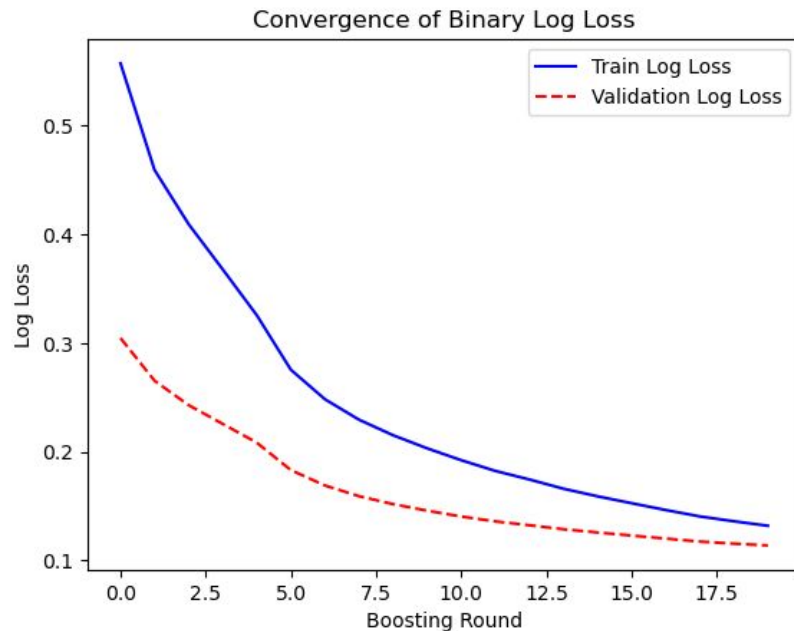
- Grows leaf-wise
- Computationally efficient
- Builds decision trees iteratively learning from previous tree
- Captures complex relationships between features
- Interpretable

CNN-LSTM Hybrid:

- Feature extraction through CNN layers (spatial information)
- Temporal dependencies through LSTM layers
- Handles complex data well predicting values per patient per time step

V. LightGBM (2)

- **Variables: 36 feature variables used.** Only columns dropped from the dataset after initial data preprocessing were Unit 1, Unit 2, Time, and Patient.
- **Min-Max Scaling (Normalization)**
- **Hyperparameters:**
 - Binary log loss
 - 31 leaves per tree
 - Learning rate of .01
 - Feature fraction of .9
- **Training:** 1000 boosting rounds

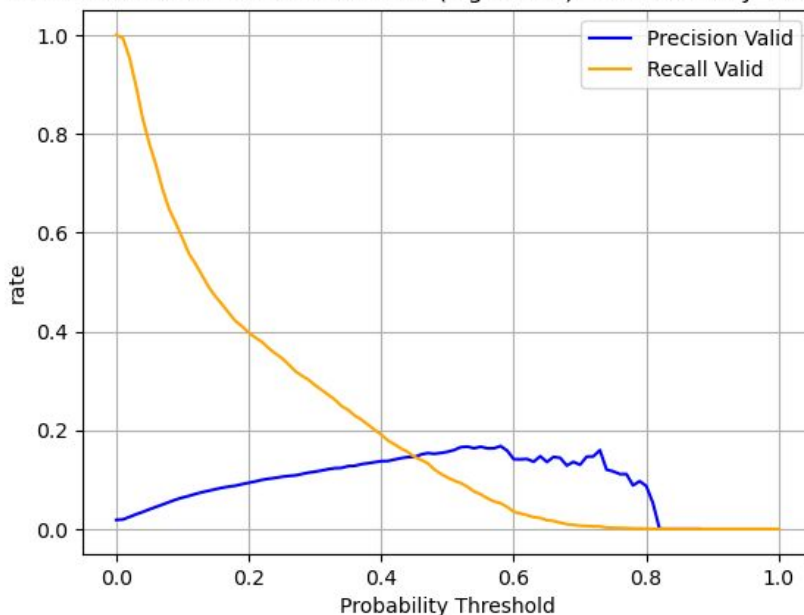


The LightGBM model showed smooth convergence in each boosting round for both datasets.

V. LightGBM's Threshold and Model Optimization (3)

- **Understanding LightGBM's Performance:** To understand how the model is performing on the valid data set we calculate performance metrics at each threshold from 0 to 1 in .01 increments.
 - This allows if the model is performing poorly and optimize our predictions for the F1 metric based on where precision and recall are close to intersecting.
 - This was done for every model including the ensemble model at the end.

Precision and Recall for Valid Data (LightGBM) vs. Probability Threshold



Thresholding at .4 gives us close to the intersection of precision and recall.

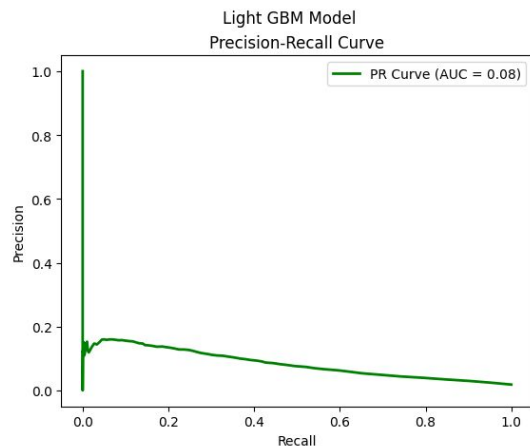
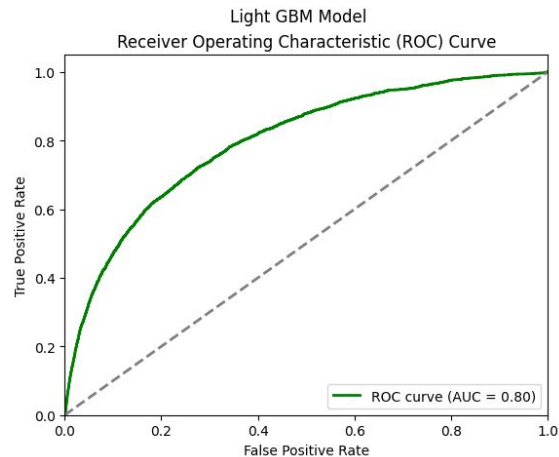
V. Final LightGBM Performance Metrics (4)

- Thresholding set to probability > .4:

Accuracy	Precision	Recall	F1-score	AUC
0.9631	0.1372	0.1907	0.1596	0.08

LightGBM Confusion Matrix

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	573 True Positives	2431 False Negatives
	Negative	3604 False Positives	157035 True Negatives



Using the .4 threshold, we can see higher recall than precision but still a strong F1.

V. CNN-LSTM Hybrid Additional Preprocessing (5)

- **Variables: 35 feature variables used.** Only dropped columns from the dataset after initial data preprocessing were Unit 1, Unit 2, Gender, Time, and Patient.
- **Min-Max Scaling (Normalization)**
- **Padding:** To feed the data into the model, padding the sequences of the data to equal the maximum time steps that a patient had in the test data (454). This allows us to return a prediction for each patient per time step.
- **Creating a 3D Tensor:**
 - (Patient, Time Steps, Features)
- **Reshaping Sepsis Label:** The outcome SepsisLabel needed to be reshaped to (Patient, Time Step, 1) for a binary prediction.

Import Data



Drop Columns



Normalize



Padding



3D Tensor



Reshaping

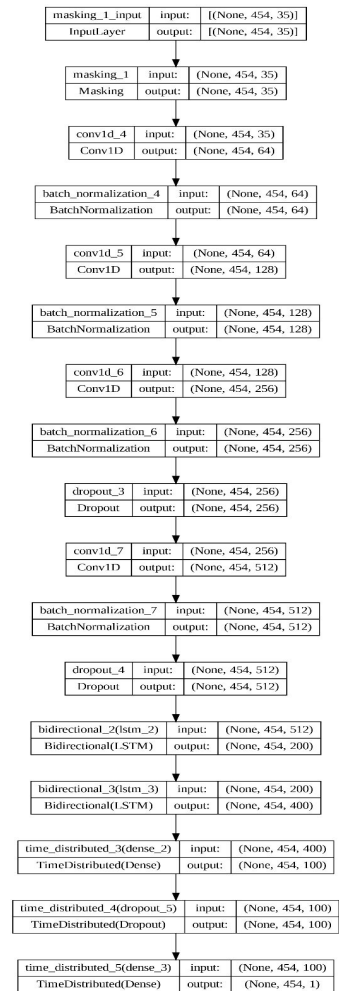
Multiple steps were needed to ensure smooth training and the desired outcome.

V. CNN-LSTM Hybrid Architecture (6)

- **Masking Layer:** Ensures the model focuses more on non-padded data.
 - Mask value equals 0
- **CNN Layers:**
 - 4 Layers
 - Filters = 64, 128, 256, 512
 - Activation = RELU
 - Kernel size = 3
 - Batch normalization
 - Batch dropout on the last two layers
 - padding = same
- **LSTM Layers:**
 - 2 Layers
 - Nodes = 100, 200
 - Bidirectional
 - Return sequence equals true to ensure a prediction for each patient at each time step
 - Recurrent and batch drop out = .2
- **Fully Connected Layer:**
 - Time distributed: each time step
 - 100 nodes
 - RELU activation
 - Batch dropout = .5
- **Output Layer:**
 - Time distributed
 - Sigmoid activation

(454, 35)
Model: "sequential_1"

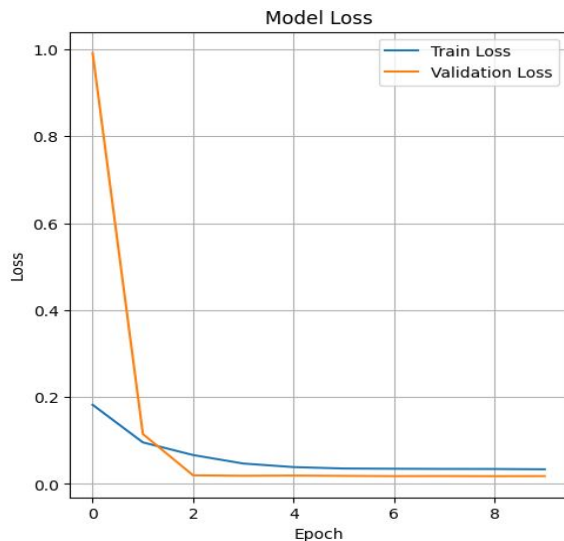
Layer (type)	Output Shape	Param #
=====		
masking_1 (Masking)	(None, 454, 35)	0
conv1d_4 (Conv1D)	(None, 454, 64)	6784
batch_normalization_4 (Batch Normalization)	(None, 454, 64)	256
conv1d_5 (Conv1D)	(None, 454, 128)	24704
batch_normalization_5 (Batch Normalization)	(None, 454, 128)	512
conv1d_6 (Conv1D)	(None, 454, 256)	98560
batch_normalization_6 (Batch Normalization)	(None, 454, 256)	1024
dropout_3 (Dropout)	(None, 454, 256)	0
conv1d_7 (Conv1D)	(None, 454, 512)	393728
batch_normalization_7 (Batch Normalization)	(None, 454, 512)	2048
dropout_4 (Dropout)	(None, 454, 512)	0
bidirectional_2 (Bidirectional)	(None, 454, 200)	490400
bidirectional_3 (Bidirectional)	(None, 454, 400)	641600
time_distributed_3 (Time Distributed)	(None, 454, 100)	40100
time_distributed_4 (Time Distributed)	(None, 454, 100)	0
time_distributed_5 (Time Distributed)	(None, 454, 1)	101



Model architecture was built up from a simple model after analyzing convergence and performance.

V. CNN-LSTM Hybrid Training and Hyperparameters (7)

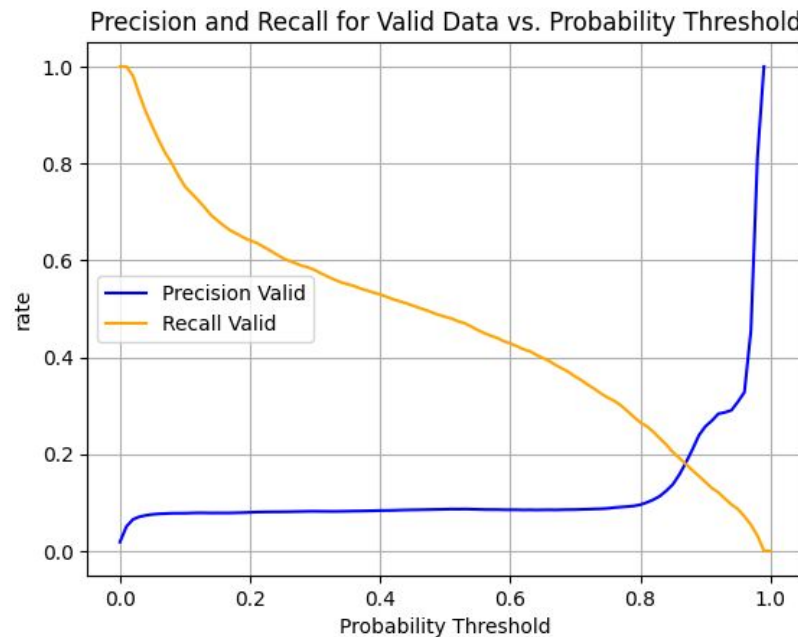
- **Optimizer:** An Adam optimizer is used specifying the global learning rate to .00001. If convergence stalled, we allowed the learning rate to adjust down to .000001.
- **Loss Function:** Binary Cross-Entropy
- **Class Weights:** A class weight of 3 is set for the minority class matching the distribution after SMOTE penalizing mismatches of the minority class three times as much.
- **Batch Size:** Batch size is set to 64.
- **Epochs:** Trained for 10 epochs.



The CNN-LSTM hybrid model training is smooth seen in the convergence.

V. CNN-LSTM Hybrid Threshold Optimization (8)

- **Converting Output to Original Form Function:** Iterating over patient ID and then iterating over each time step for the patient to return a dataframe with predictions in the original form.
- **Initial .5 Threshold Check:** **Train F-1 score = .21, Valid F-1 score = .14**
 - Recall increases substantially on the valid data set after we introduced class weights.
- **Threshold Optimization:** Due to the increased recall, the optimal threshold was around .87.



The optimal threshold for the CNN-LSTM hybrid model is at .87.

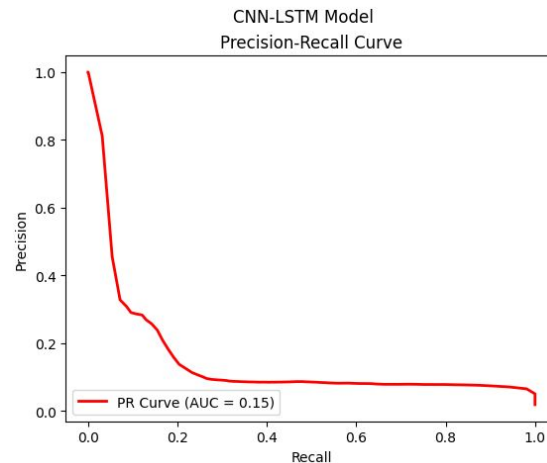
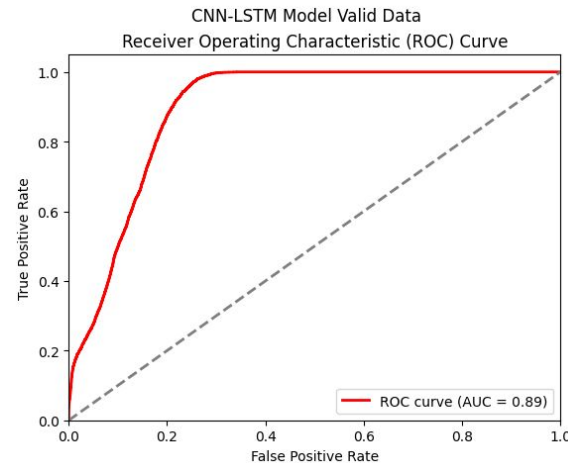
V. Final CNN-LSTM Hybrid Performance Metrics (9)

- **Thresholding set to probability > .87:**
 - Best Test Submission: .21132 public, .19135 private

Accuracy	Precision	Recall	F1-score	AUC
0.970	0.1823	0.1791	0.1807	0.15

CNN-LSTM Confusion Matrix

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	538 True Positives	2466 False Negatives
	Negative	2413 False Positives	158226 True Negatives



Thresholding at .87 gives us strong performance metrics.

VI. Ensemble Strategy for Sepsis Prediction (1)

- **Why Ensemble CNN-LSTM and LightGBM?** To leverage both model's strengths for improved prediction reliability.
- **Combining Predictions**
 - Used weighted average of the CNN-LSTM and LightGBM models' probabilities
 - **Weights: 60% CNN-LSTM** and **40% LightGBM**, based on each model's optimized F1 score.

General Ensemble Formula:

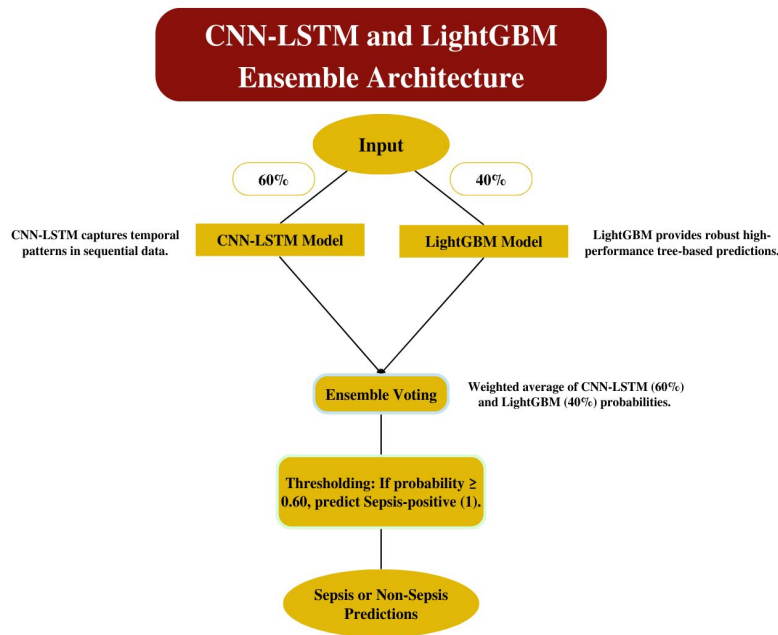
$$P_{\text{ensemble}} = (w_{\text{CNN-LSTM}} \times P_{\text{CNN-LSTM}}) + (w_{\text{LightGBM}} \times P_{\text{LightGBM}})$$

$w_{\text{CNN-LSTM}}$: Weight assigned to CNN-LSTM model

$P_{\text{CNN-LSTM}}$: Probability prediction from CNN-LSTM model

w_{LightGBM} : Weight assigned to LightGBM model

P_{LightGBM} : Probability prediction from LightGBM model

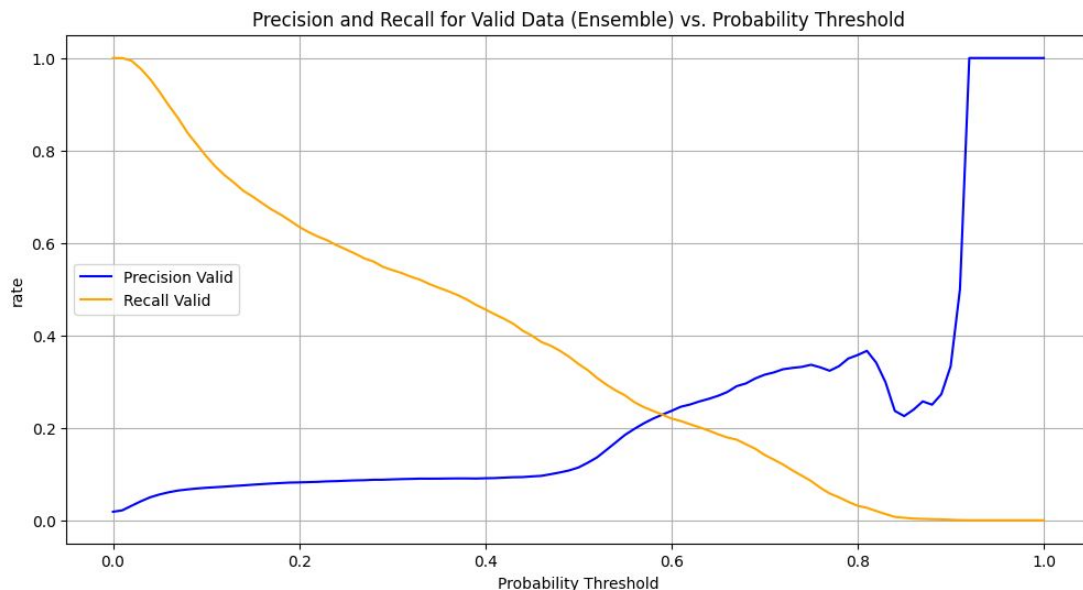


Leveraging CNN-LSTM's time-series strengths and LightGBM's accuracy to improve sepsis prediction.

VI. Ensemble Strategy for Sepsis Prediction (2)

- **Thresholding for Final Predictions**
 - Applied **0.60 threshold** on the combined probabilities.
 - Classified predictions as sepsis-positive (1) if ensemble probability ≥ 0.60 .
- **Balanced Thresholding**
 - The ensemble blends **LightGBM's high precision at lower thresholds** with **CNN-LSTM high recall at higher thresholds**, maximizing the F-1 score.

$$\text{Final Prediction} = \begin{cases} 1 & \text{if } P_{\text{ensemble}} \geq 0.6 \\ 0 & \text{if } P_{\text{ensemble}} < 0.6 \end{cases}$$

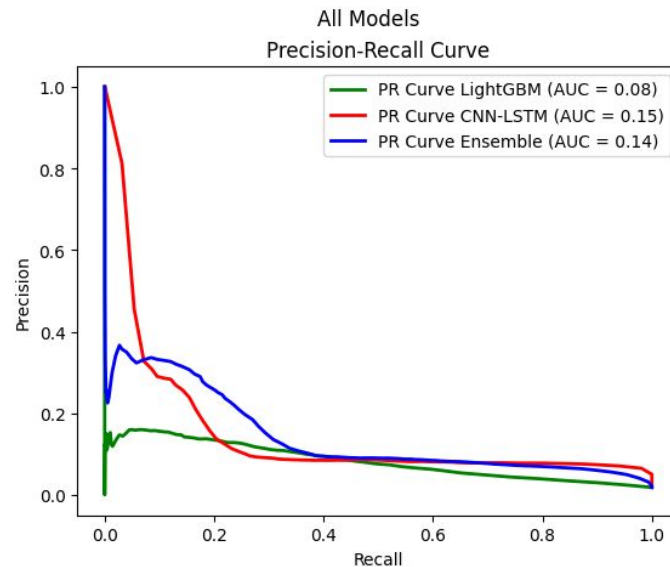
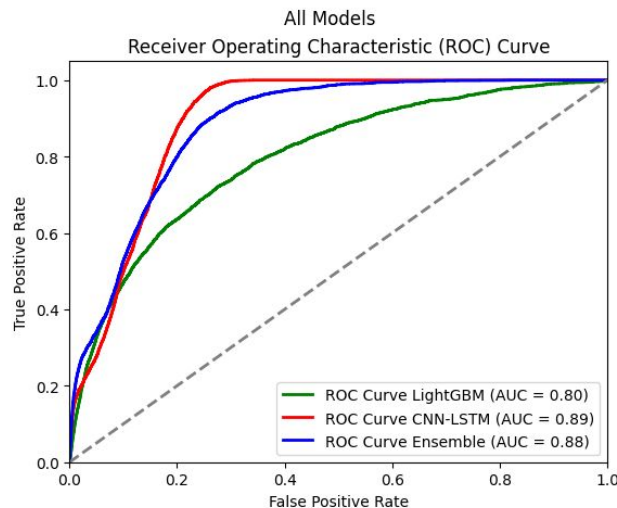


Thresholding at 0.60 helps balance sensitivity and specificity, promoting reliable sepsis detection.

VI. Ensemble Strategy for Sepsis Prediction (3)

Comparison of Model PR Curves and ROC-AUC Curves

- Higher PR AUC for CNN-LSTM implies more consistent positive predictions across thresholds.
 - Shows balance between precision and recall over various thresholds, not just the final one chosen.
- PR AUC is average performance metric, **doesn't guarantee best results at the chosen threshold.**



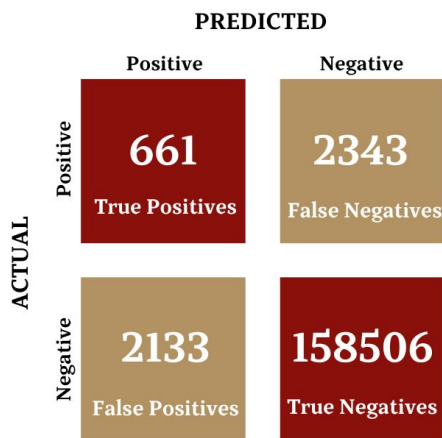
PR AUC shows balance, but final-threshold performance defines the best model for real-world application.

VI. Ensemble Strategy for Sepsis Prediction (4)

Comparison of Performance Metrics

Model	Accuracy	Precision	Recall	F1-score
LightGBM	0.9631	0.1372	0.1907	0.1596
CNN-LSTM	0.970	0.1823	0.1791	0.1807
Ensemble	0.9726	0.2366	0.2204	0.2280

Ensemble Model Confusion Matrix



- **Better Precision and Recall:** Ensemble has higher precision and recall at chosen threshold, **detecting more true positives with greater accuracy.**
- **F1 Score:** Ensemble’s higher F1 score shows **balanced improvement**, important for sensitive cases like sepsis detection
- **Real-World Application Priority:** Ensemble’s strong threshold-based metrics make it better choice for deployment, **focusing on reliable positive class predictions over average PR AUC.**

Ensemble offers better balance of precision and recall, making it the strongest choice for model deployment.

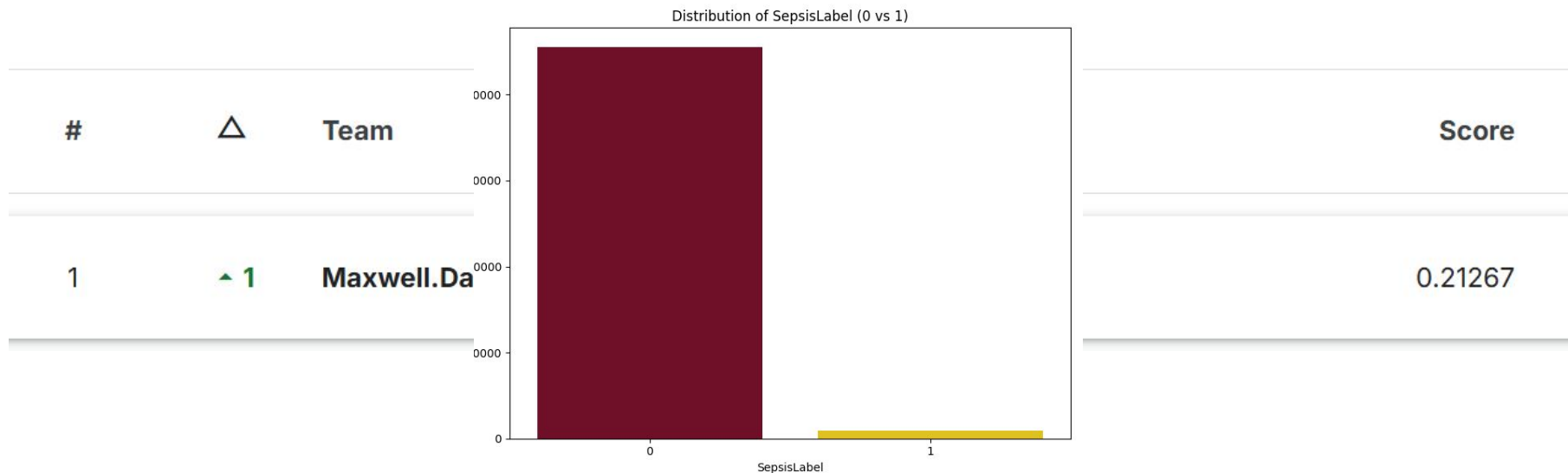
VI. Ensemble Strategy for Sepsis Prediction (5)

- **Results Summary**

- Final submission was created with weighted probabilities.
- Classified **9428 cases** as sepsis-positive based on the ensemble output.
- Ensemble model outperformed individual models, showing stronger predictions.

- **Leaderboard Scores**

- **Public— 0.22293 F-1 score, Private— 0.21267 F-1 score** achieving the highest private score and demonstrating a strong performance.



Ensembling models enhanced performance, facilitating strong performance on leaderboard.

VII. Key Takeaways and Business Implications

- **Potential Improvements**
 - Exploring other models with **diverse architectures**— KNN, Random Forests, Transformers
 - Using external datasets to **recreate clinical indices** like the SOFA (Sequential Organ Failure Assessment) index could improve model accuracy by capturing more comprehensive patient health metrics.
- **Business Implications**
 - **Early detection = Better outcomes:** **Reduced mortality** and **improved patient survival rates**.
 - **Cost-effective care:** Allocates resources efficiently, reducing unnecessary treatments and hospital stays— an **economically efficient** solution as hospital resources are limited.
 - **Integrating with healthcare systems:** **enhances decision-making, streamlines interventions**, and **maximizes patient safety**.

Improving the model with diverse architectures and data may boost prediction accuracy and patient outcomes.

References

- Reyna, Matthew A., et al. "Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019." *PhysioNet*, <https://physionet.org/content/challenge-2019/1.0.0/papers/CinC2019-054.pdf>. Accessed 14 Nov. 2024.
- Perez Alday, Erick A., et al. "Classification of Clinical Time-Series Data with Neural Networks: The PhysioNet/Computing in Cardiology Challenge 2019." *PhysioNet*, <https://physionet.org/content/challenge-2019/1.0.0/papers/CinC2019-367.pdf>. Accessed 14 Nov. 2024.
- "Sepsis Fact Sheet." *National Institute of General Medical Sciences*, <https://www.nigms.nih.gov/education/fact-sheets/Pages/sepsis.aspx#:~:text=Sepsis%20is%20a%20serious%20condition,are%20readmitted%20to%20the%20hospital>. Accessed 14 Nov. 2024.
- Goldberger, Ary L., et al. "PhysioNet/Computing in Cardiology Challenge 2019." *PhysioNet*, <https://physionet.org/content/challenge-2019/1.0.0/>. Accessed 14 Nov. 2024.
- Rangel-Frausto, Michael S., et al. "The Natural History of the Systemic Inflammatory Response Syndrome (SIRS): A Prospective Study." *National Center for Biotechnology Information*, <https://pmc.ncbi.nlm.nih.gov/articles/PMC11348270/>. Accessed 14 Nov. 2024.
- "Expenses per Inpatient Day." *KFF: Health Costs*, <https://www.kff.org/health-costs/state-indicator/expenses-per-inpatient-day/?currentTimeframe=0&sortModel=%7B%22colId%22:%22Location%22,%22sort%22:%22asc%22%7D>. Accessed 14 Nov. 2024.
- Vincent, Jean-Louis, et al. "The Epidemiology of Severe Sepsis and Septic Shock." *National Center for Biotechnology Information*, [https://pmc.ncbi.nlm.nih.gov/articles/PMC6250243/#:~:text=The%20average%20length%20of%20stay,for%20septic%20shock%20\(6\)](https://pmc.ncbi.nlm.nih.gov/articles/PMC6250243/#:~:text=The%20average%20length%20of%20stay,for%20septic%20shock%20(6)). Accessed 14 Nov. 2024.
- EvidentlyAI Team. "Explain ROC Curve." *Evidently AI*, <https://www.evidentlyai.com/classification-metrics/explain-roc-curve>. Accessed 14 Nov. 2024.
- Song, Michael. "Understanding the ROC-AUC Curve." *Medium*, <https://medium.com/@msong507/understanding-the-roc-auc-curve-cc204f0b3441>. Accessed 14 Nov. 2024.
- "Precision and Recall Curves." *Analytics India Magazine*, <https://analyticsindiamag.com/topics/precision-and-recall-curves/>. Accessed 14 Nov. 2024.
- "Choosing Performance Metrics." *Towards Data Science*, <https://towardsdatascience.com/choosing-performance-metrics-61b40819eae1>. Accessed 14 Nov. 2024.

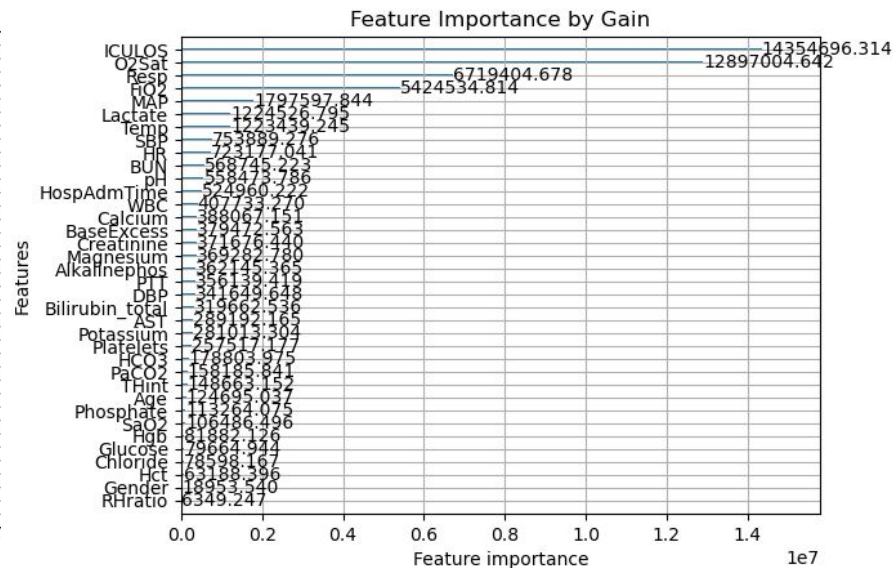
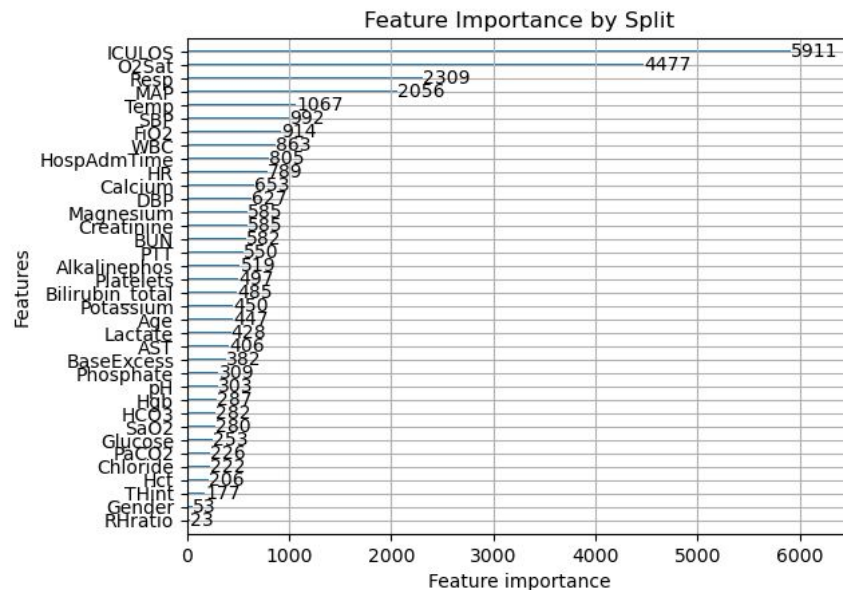
Thank you for listening!

Q&A Session



BOSTON
COLLEGE

VIII. Extra Slides: LightGBM Feature Importance



VIII. Extra Slides: Ensemble Model Example Calculation

$$P_{\text{ensemble}} = (w_{\text{CNN-LSTM}} \times P_{\text{CNN-LSTM}}) + (w_{\text{LightGBM}} \times P_{\text{LightGBM}})$$

Weight of **CNN-LSTM**: 0.6

Weight of **LightGBM**: 0.4

Threshold: 0.6

Example:

What would the prediction be if the **probability prediction of CNN-LSTM is 0.7** and the **probability prediction from the LightGBM model is 0.4**?

Our ensemble model calculations would look like this:

$$P_{\text{ensemble}} = (0.6 \times 0.7) + (0.4 \times 0.4)$$

$$P_{\text{ensemble}} = 0.42 + 0.16 = 0.58$$

If the value of **0.58** is **greater than or equal** to our **threshold value of 0.6**, we would classify it as sepsis-positive (1).

In this case, **0.58 < 0.6** and the ensemble will therefore be **classified as non-sepsis (0)**.

$$\text{Final Prediction} = \begin{cases} 1 & \text{if } P_{\text{ensemble}} \geq 0.6 \\ 0 & \text{if } P_{\text{ensemble}} < 0.6 \end{cases}$$

VIII. Extra Slides: LightGBM Model Formulas

Sequential Trees: New tree corrects errors from previous tree.

Leaf Wise Splitting: Splits based on highest gain across all potential splits.

$$\text{Binary Log Loss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

$$\text{Gain} = \frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{(G_L + G_R)^2}{H_L + H_R}$$

Gain Function Components:

- **G** = Sum of residual for split
- **H** = Hessians for the split

Regularization:

Max Leaves = 31

- 31 maximum number of leaves per tree

Feature Fraction = .9

- Only considers a .9 of the features when splitting a node

VIII. Extra Slides: CNN-LSTM Hybrid Model Formulas

CNN Layers:

$$\text{Conv}(x)_t = \sigma \left(\sum_{j=1}^k f_j \cdot x_{t+j-1} + b \right)$$

σ = Relu Activation

f_j = Weight of the filter at the jth position

x_{t+j-1} = Input value at a specific position

LSTM Layers:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (\text{forget gate})$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (\text{input gate})$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (\text{candidate cell state})$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (\text{updated cell state})$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (\text{output gate})$$

$$h_t = o_t \odot \tanh(C_t) \quad (\text{hidden state})$$

W = Weight terms at each gate

h = Hidden states

b = Bias terms at gate

σ = Sigmoid function at output gate

\odot = Element wise multiplication

Bidirectional:

$$H_t = [\vec{h}_t; \overleftarrow{h}_t]$$

VIII. Extra Slides: CNN-LSTM Hybrid Model Formulas (Cont.)

Time Distributed Dense Layer:

$$o_t = \sigma(W_d \cdot H_t + b_d)$$

σ = Relu Activation

W_d and b_d = Weight and bias term for dense layers

H_t = Hidden state from LSTM layers

Output Layer (Sigmoid):

$$p_t = \frac{1}{1 + e^{-o_t}}$$

p_t = Probability output at each time step per patient

o_t = Output from dense layer

VIII. Extra Slides: F-1 Score, Precision, Recall Definitions

- The F-1 score is the harmonic mean of precision and recall, taking values between 0 and 1.

$$\text{F1 score} = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

or

$$\text{F1 score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

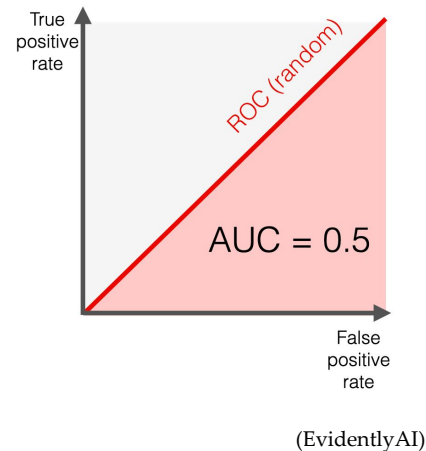
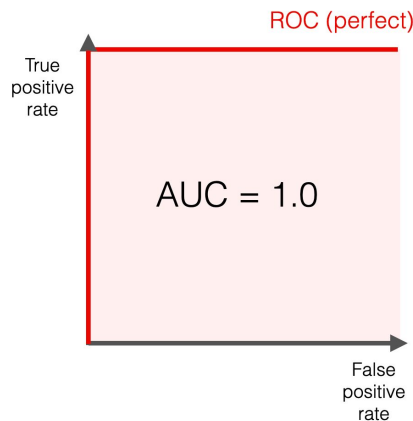
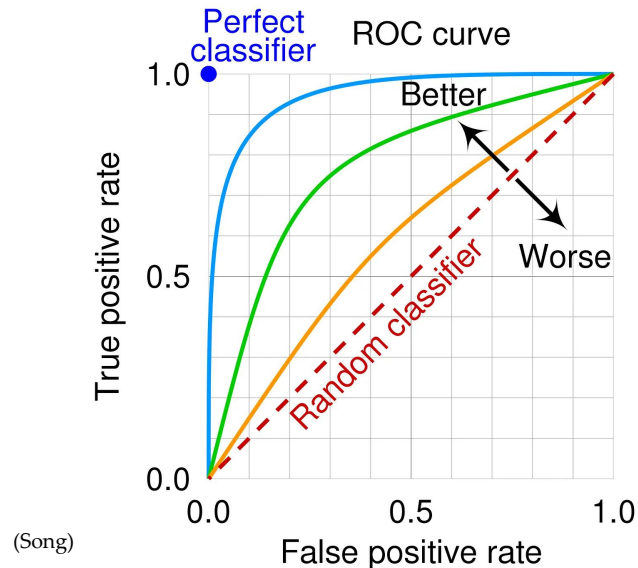
(Lanier)

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

(Lanier)

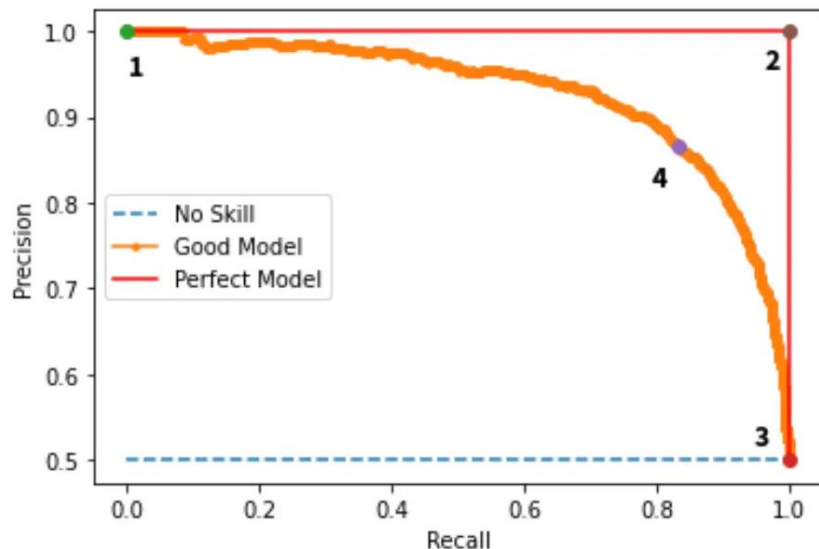
VIII. Extra Slides: Understanding ROC-AUC

- ROC curve shows how well classification model distinguishes between classes at various thresholds.
- ROC-AUC is misleading when class imbalance is severe
 - Bad models can still achieve high AUC through correctly identifying the majority class while barely detecting the minority class.
 - ROC curve also doesn't directly account for the imbalance, does not reflect how well the model captures the minority class.
- ROC-AUC score of 0.5 = random guessing, anything above shows slight predictive power



VIII. Extra Slides: Understanding Precision-Recall Curve

- Precision Recall curve is often used when classes are imbalanced.
- PR-AUC particularly emphasizes model performance on the positive class
- Reflects average performance across various thresholds



(Dharwal)

VIII. Extra Slides: Estimated Economic Value Estimations for the U.S.

- **Estimations and Assumptions:**

- **Annual Sepsis Cases:** 1.7 million (National Institute of General Medical Sciences)
- **Annual Sepsis Deaths:** at least 350,000 (National Institute of General Medical Sciences)
- **Percentage of U.S. Healthcare Expenses Attributed to Sepsis:** 13% (Reyna et al.)
- **Percentage of Sepsis Death Rate:** 17.5% (La Via et al.)
- **Mortality Reduction Due to Our Algorithm:** 1.13%
- **Reduced Length of Stay (LOS):** 2 days per patient, around 4-5 day average for Sepsis patients (Paoli et al.)
- **Cost Per Hospital Day:** \$3025 per day (KFF.org)
- **Average Cost per Sepsis Patient:** \$24,000 (Paoli et al.)

1. **Lives Saved**

- a. Lives Saved = Annual Sepsis Cases * Mortality Reduction Due to Algorithm
- b. Lives Saved = $1,700,000 * 0.0113 = \mathbf{19,210 \text{ lives}}$

2. **Bed Days Saved**

- a. Bed Days Saved = Annual Sepsis Cases * LOS Reduction
- b. Bed Days Saved = $1,700,000 * 2 = \mathbf{3,400,000 \text{ bed days}}$

3. **Cost Savings from Reduced LOS**

- a. Cost Savings = Bed Days Saved * Cost per Hospital Day
- b. Cost Savings = $3,400,000 * \$3,025 = \mathbf{\$10,285,000,000}$

4. **Cost Savings from Reduced Mortality**

- a. Cost Savings = Lives Saved * Average Cost Per Sepsis Patient
- b. Cost Savings = $19,210 * \$24,000 = \mathbf{\$461,040,000}$

5. **Total Cost Savings**

- a. Total Cost Savings = Cost Savings from Reduced LOS + Cost Savings from Reduced Mortality
- b. Total Cost Savings = $\$10,285,000,000 + \$461,040,000 = \mathbf{\$10,746,040,000}$

Note: these are simplified estimations on a nuanced issue.

Other factors to consider— patient quality of life or post-hospitalization care, and any potential additional treatment costs that still might come up in successfully treated cases.

However, the purpose is to show the substantial benefits the algorithm could bring, acknowledging that a more detailed calculation would include these factors.