SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTING TECHNOLOGIES
18CSP107L / 18CSP108L - MINOR PROJECT

Early Prediction of Sepsis using Ensemble Learning

Guide name: Dr. G. Abiraami

Designation: Assistant Professor

Department: Department of Computing Technologies

Batch ID:

Student 1 Reg. No:RA2011003010642
Student 1 Name: Anamika Nahar

Student 2 Reg. No:RA2011003010654
Student 2 Name:K. Sai Hemanth

ABSTRACT

- Sepsis is a life-threatening syndrome with diverse clinical presentations. Timely identification and treatment are crucial to reduce mortality and improve outcomes.
- Existing prediction systems have limitations at the individual level, prompting the use of machine learning techniques to develop more effective models.
- This projects presents machine-learning-based sepsis prediction models using vital signs, laboratory tests, and demographics from the physionet dataset which are considered in both model development and application design.
- Various ensemble machine learning models have been used to provide greater accuracy.

PROBLEM STATEMENT

- The timely and accurate detection of sepsis is critical for improving patient outcomes.
- Existing methods for sepsis detection often lack the required sensitivity and specificity at the individual level.
- This can result in delayed interventions and increased mortality rates.
- To address this issue, there is a need to develop an ensemble learning-based approach for early sepsis detection to enhance clinical decision-making and patient care.

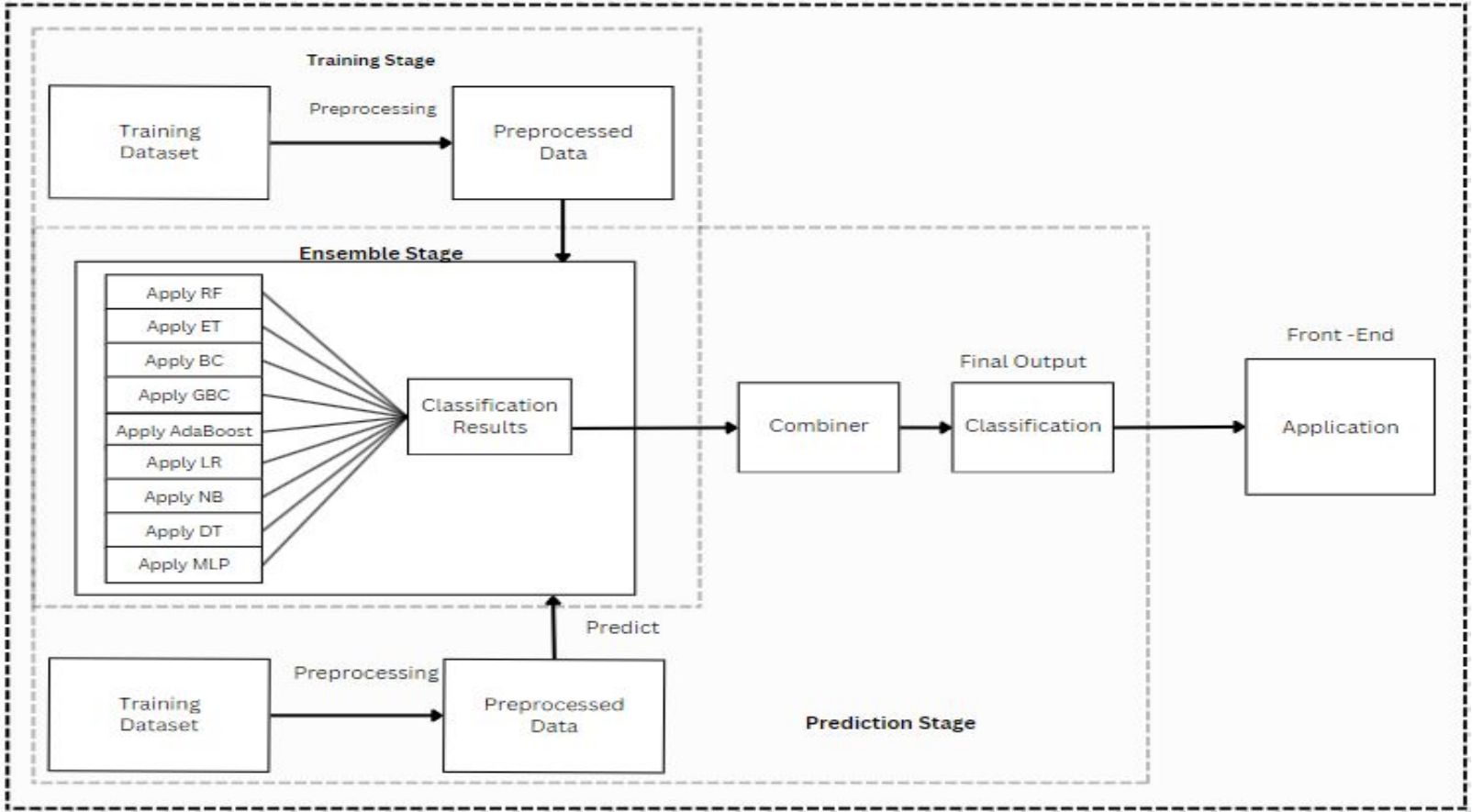
OBJECTIVES

- **Data Set Utilization**: Utilize relevant datasets, including the PhysioNet Challenge 2019 dataset, for sepsis diagnosis research.
- **Literature Review**: Review existing work in the domain of early diagnosis of sepsis using artificial intelligence (AI) systems, summarizing the state of the science.
- **Algorithm Implementation**: Implement machine learning algorithms such as MLP, AdaBoost, Gradient Boosting, GaussianNB, Linear Discriminant Analysis, Random Forest Classifier, Extra Tree Classifier, and Bagging Classifier for sepsis diagnosis.
- **Parameter Optimization**: Fine-tune algorithm parameters to improve diagnostic accuracy and performance.

OBJECTIVES

- **Feature Engineering**: Apply feature engineering techniques to enhance the effectiveness of selected algorithms.
- **Early Detection**: Focus on early detection of sepsis, striving to predict sepsis onset in advance to facilitate timely intervention and reduce patient mortality.
- **Performance Evaluation**: Evaluate the performance of implemented algorithms using appropriate metrics such as accuracy, F1-score, and ROC AUC score.

ARCHITECTURE



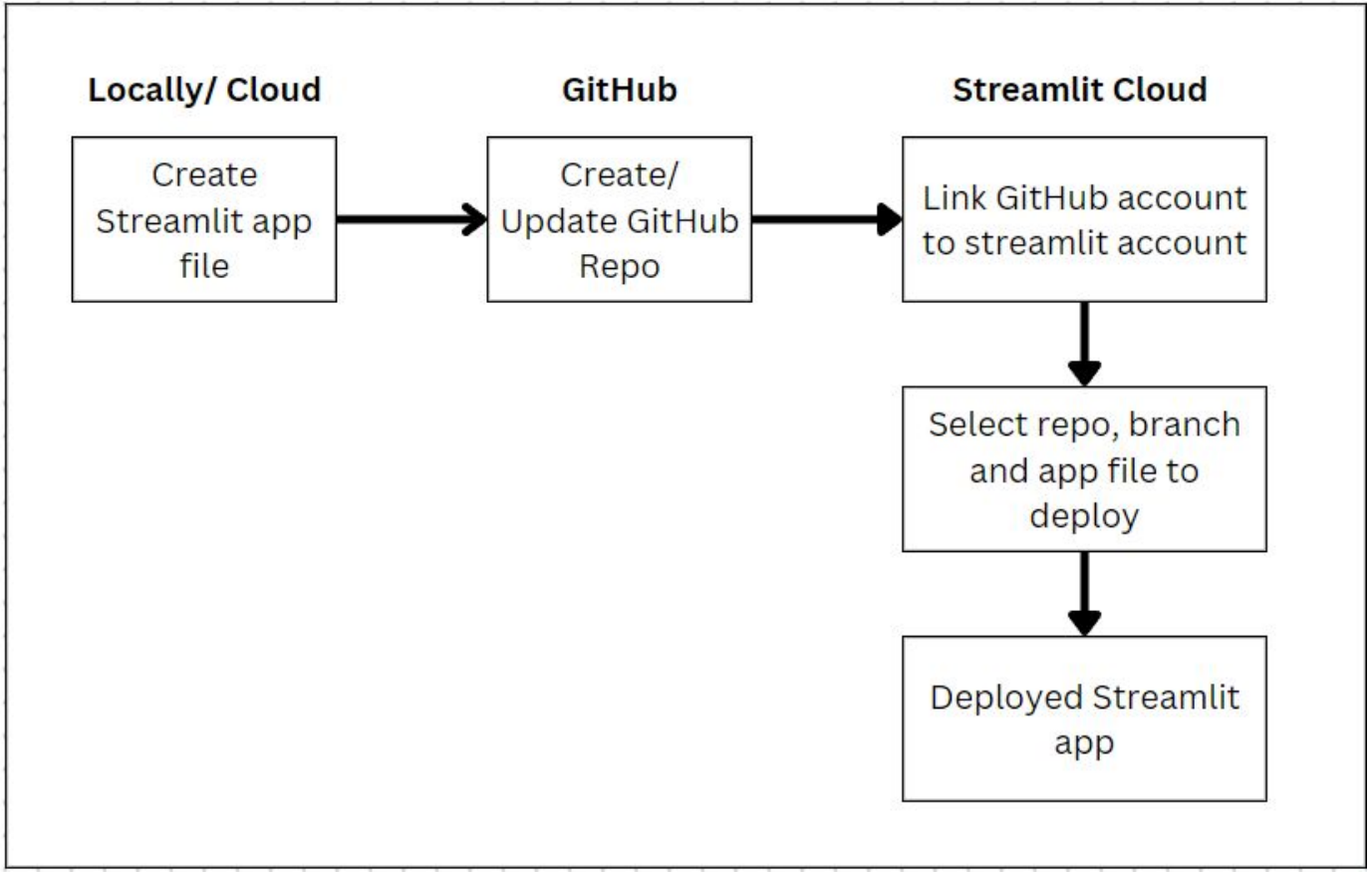
ARCHITECTURE

- The project's architecture includes a training phase and a testing phase, each involving the preprocessing phase, assembly stage, combiner, classification phase, final output generation phase, and front-end phase.
- The training dataset module collects and organizes data for training machine-learning models, serving as the foundation for model training.
- Preprocessed Data module cleans and prepares raw data for analysis, including handling missing values, scaling, normalization, and categorical encoding.
- The ensemble stage combines multiple models using techniques like bagging, boosting, and stacking to enhance system performance.
- The combiner module merges individual model predictions into a consolidated final prediction using techniques like averaging, voting, or weighted averaging.

ARCHITECTURE

- The classification module assigns predefined categories or labels to data based on patterns identified during training, using various classification algorithms.
- The front-end module provides a user interface for interacting with the system, inputting data, configuring settings, and viewing results.
- The testing dataset module evaluates model performance on unseen data, offering insights into real-world effectiveness.
- The Streamlit-based front-end architecture involves a PKL file for serialized machine-learning models, a Streamlit application for user interaction, create/update functionality, a GitHub repository for source code and files, and deployment in the Streamlit Cloud.
- The backend design includes raw data collection, data preprocessing, exploratory data analysis, data balancing, train-test data split, model training, hyperparameter tuning, ensemble algorithms, and serialization of the model into a PKL file for the frontend.

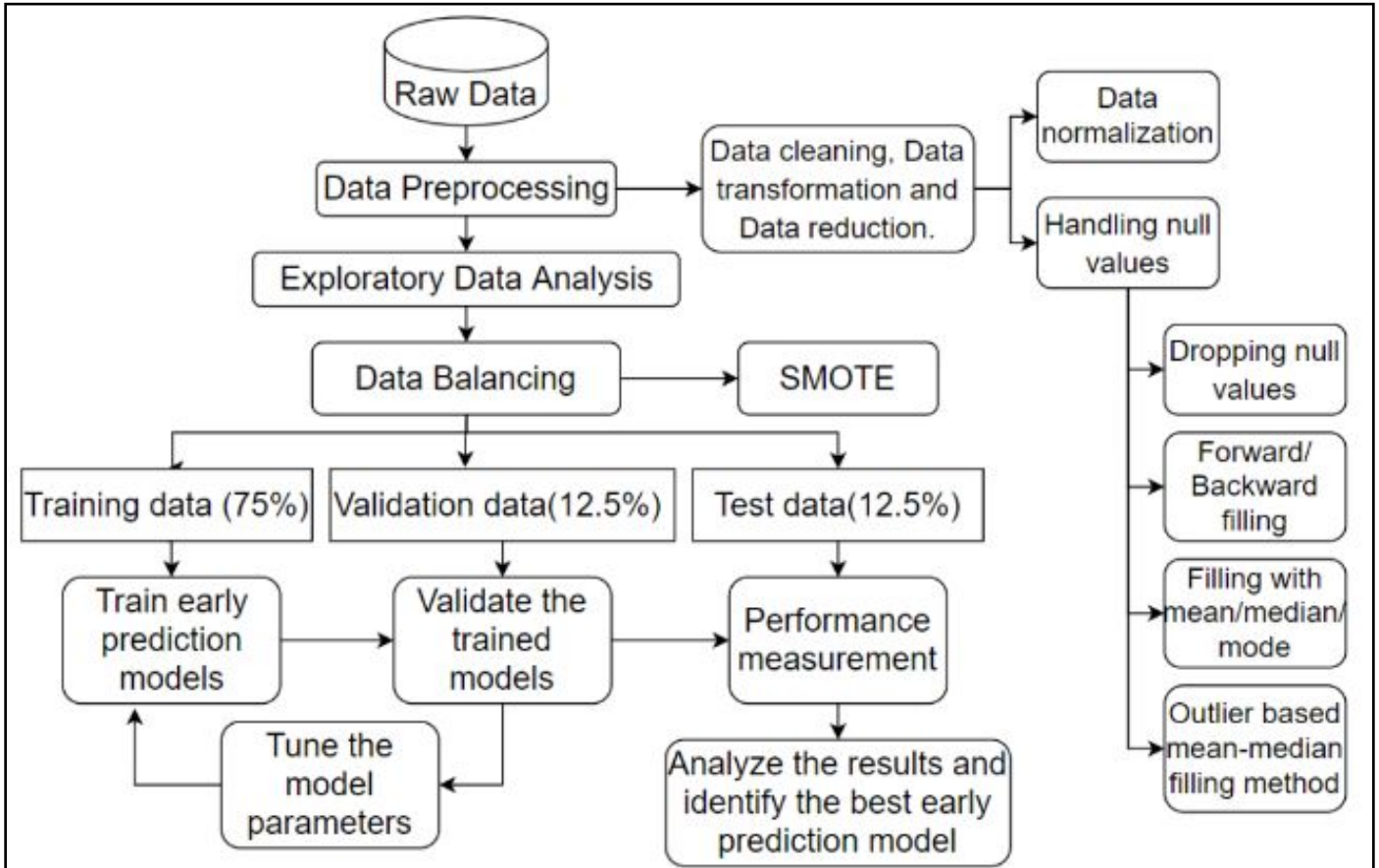
METHODOLOGY FOR FRONT-END



METHODOLOGY FOR FRONT-END

- PKL File:
  - Serialized object format in Python.
  - Stores pre-trained machine-learning models.
  - Holds the backend-trained model for real-time predictions.
- Stream-lit Application:
  - Open-source Python library for ML web apps.
  - Interface for user interaction with the model.
  - Fetches the model from the PKL file for predictions.
- Create/Update Functionality:
  - Allows users to input data.
  - Essential for making predictions.
- GitHub Repository:
  - Hosts source code and files.
  - Supports version control and collaboration.
- Deploy in Stream-lit Cloud:
  - Uses Stream-lit Sharing for deployment.
  - Makes the model accessible through a web-based interface.

METHODOLOGY FOR BACK-END



METHODOLOGY FOR BACK-END

- Data is collected, including patient information, vital signs, and lab test results, for analysis.
- Raw data undergoes preprocessing, including cleaning, transformation, handling missing values, and encoding.
- Critical features, such as heart rate and blood pressure, are selected for sepsis detection.
- Ensemble learning methods like Random Forest, AdaBoost, and Gradient Boosting are used for sepsis diagnosis.
- Model performance is optimized through hyperparameter tuning, employing techniques like grid search and random search.
- Model performance is assessed using evaluation metrics like accuracy and log loss.

METHODOLOGY FOR BACK-END

- A user-friendly interface is created using Streamlit for data input, predictions, and results viewing.
- User-provided patient data is preprocessed within the application before being used by the machine-learning models.
- Trained models are integrated into the Streamlit app to enable real-time predictions.
- The Streamlit app is initiated through a terminal command, facilitating user interaction.
- Users can obtain real-time predictions for sepsis detection through the app's interface.
- Source code and files are hosted in a GitHub repository, allowing version control and collaboration.
- Deployment is carried out through Streamlit Sharing, providing access via a web-based interface.

MODULES DESCRIPTION

- **Data Loading and Preprocessing Module:**
  - Responsible for loading the raw sepsis dataset.
  - Performs data preprocessing, which may include data cleaning, imputing missing values, and normalizing the data.
- **Exploratory Data Analysis Module:**
  - Provides insights into the dataset by visualizing statistics.
  - Helps understand data distribution, relationships, and class imbalances.
  - Generates visualizations like pie charts, bar graphs, and count plots to illustrate data characteristics.
- **Data Resampling Module:**
  - Balances the dataset by resampling to address class imbalances.
  - Uses techniques like upsampling or downsampling to ensure an equal representation of sepsis and non-sepsis cases.

MODULES DESCRIPTION

- **Data Splitting Module**:
  - Divides the dataset into training and testing sets.
  - Ensures that the data used for model training and evaluation is separate.
- **Label Encoding Module**:
  - Encodes the target labels for machine learning models.
  - Converts the binary labels (e.g., 0 and 1 for sepsis and non-sepsis) into a format suitable for model training.
- **Machine Learning Models Module**:
  - Utilizes various machine learning algorithms, including MLP, AdaBoost, GradientBoosting, GaussianNB, LinearDiscriminantAnalysis, and QuadraticDiscriminantAnalysis, RandomForest, ExtraTrees, BaggingClassifier, and LogisticRegression.
  - Trains these models on the preprocessed data to predict sepsis.

MODULES DESCRIPTION

- **Model Evaluation Module:**
  - Evaluates the performance of each machine learning model using metrics such as accuracy and log loss.
  - Compares the results to determine which model performs best on the dataset.
- **Streamlit Front-End Module:**
  - Builds a user-friendly web interface using Streamlit.
  - Allows users to input data and view model predictions and visualizations.
  - Ensures the app is responsive and presents results in real time.

IMPLEMENTATION

- **Importing Libraries:**
  - Import essential Python libraries, including NumPy, Pandas, Matplotlib, Seaborn, and others for data analysis and visualization.
- **Read the Dataset:**
  - Load the sepsis dataset from a CSV file named "sepsis_data.csv."
  - Display the first and last five rows of the dataset using dataset.head() and dataset.tail().

```
dataset.columns
```

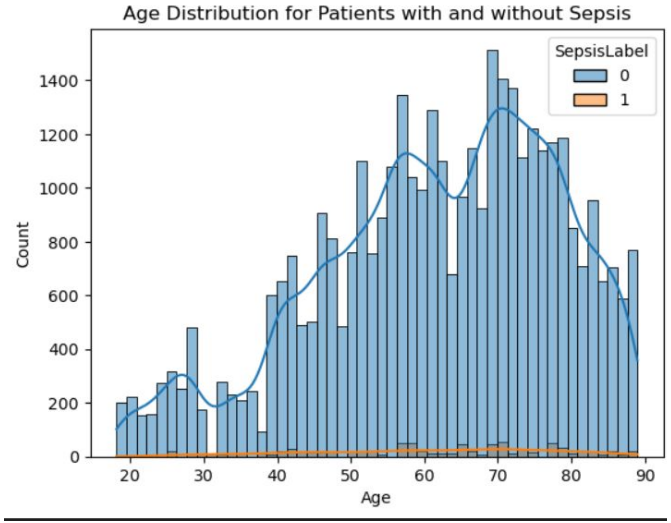
```
Index(['HR', 'O2Sat', 'Temp', 'SBP', 'MAP', 'DBP', 'Resp', 'EtCO2',
       'BaseExcess', 'HCO3', 'FiO2', 'pH', 'PaCO2', 'SaO2', 'AST', 'BUN',
       'Alkalinephos', 'Calcium', 'Chloride', 'Creatinine', 'Bilirubin_direct',
       'Glucose', 'Lactate', 'Magnesium', 'Phosphate', 'Potassium',
       'Bilirubin_total', 'TroponinI', 'Hct', 'Hgb', 'PTT', 'WBC',
       'Fibrinogen', 'Platelets', 'Age', 'Gender', 'Unit1', 'Unit2',
       'HospAdmTime', 'ICULOS', 'SepsisLabel'],
      dtype='object')
```

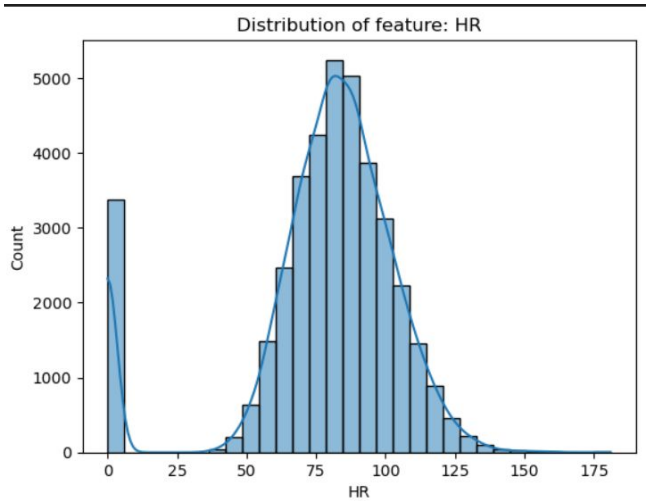
```
## Check the missing values
null_values = dataset.isnull().mean()*100
null_values = null_values.sort_values(ascending=False)
null_values
```

```
HR                    0.0
Glucose               0.0
Magnesium             0.0
Phosphate             0.0
```

IMPLEMENTATION

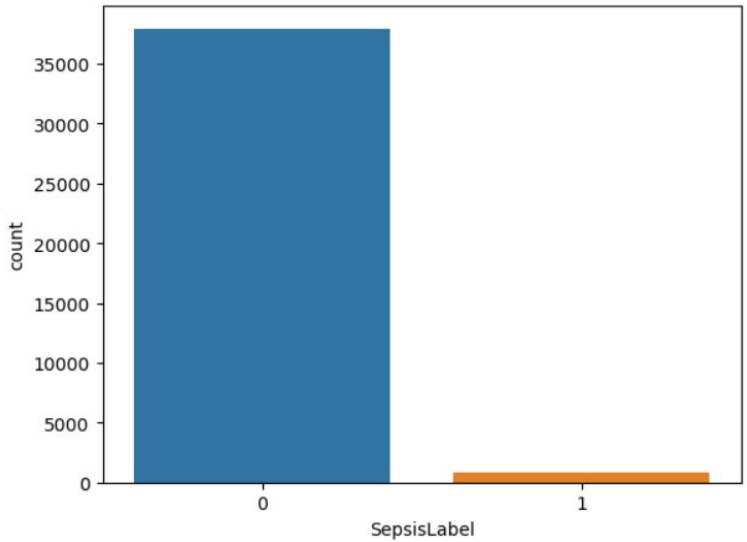
- **Exploratory Data Analysis (EDA):**
  - Examine the dataset to understand the distribution of the "SepsisLabel" class.
  - Visualize the class distribution using pie charts and bar graphs.

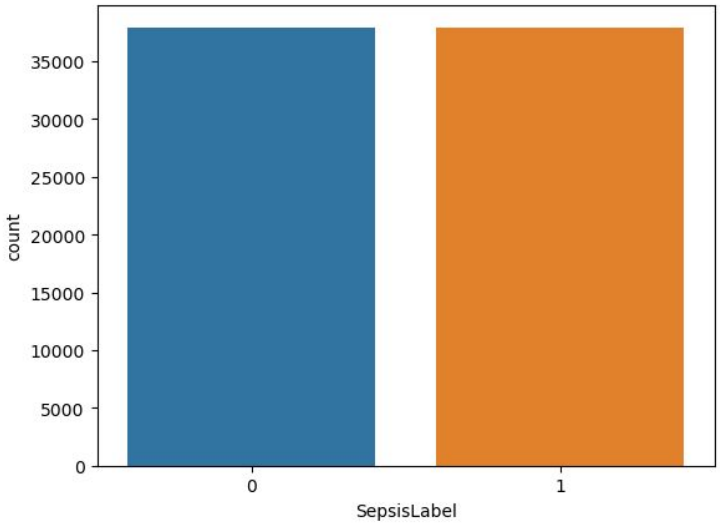




IMPLEMENTATION

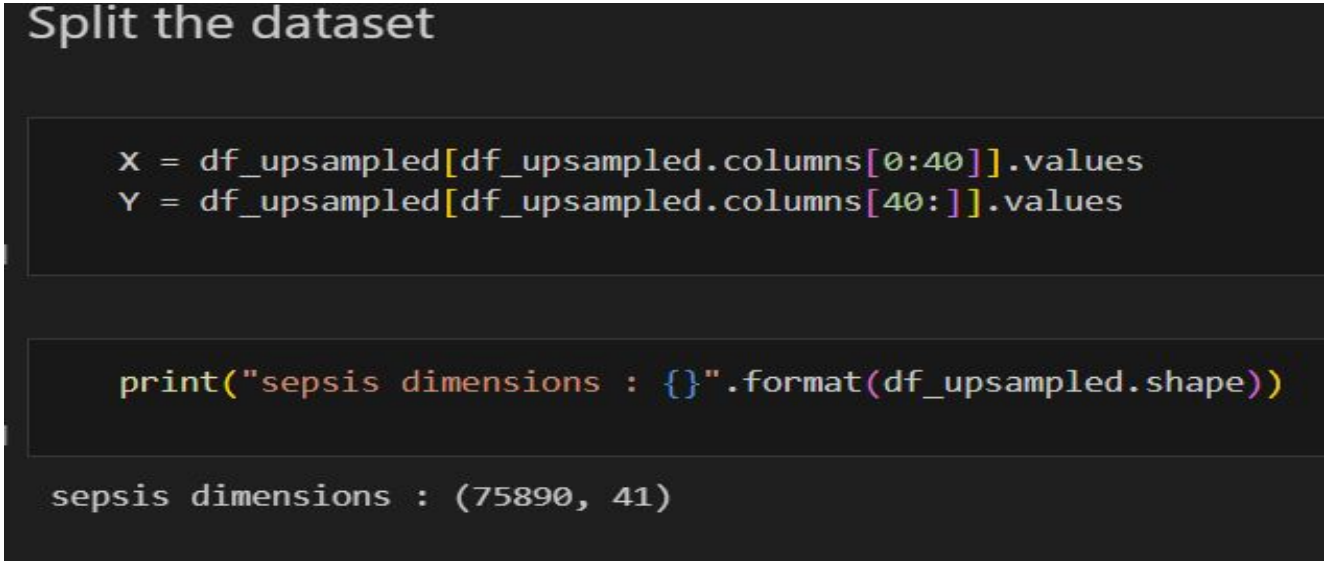
- **Re-sample the Data:**
  - Address class imbalance by upsampling the minority class, resulting in a balanced dataset.





IMPLEMENTATION

- **Split the Dataset:**
  - Split the dataset into training and testing sets using a split ratio.
  - Display the dimensions of the training and testing datasets.



IMPLEMENTATION

- **Normalize the Labels (LabelEncoder):**
  - Apply LabelEncoder from scikit-learn to encode the labels (0 and 1) to facilitate model training.

```
#Printing dimensions of sepsis dataset only with label column
print("sepsis dimensions only label : {}".format(Y.shape))
```

```
sepsis dimensions only label : (75890, 1)
```

Normalize the labels-LabelEncoder

```
from sklearn import preprocessing
labelencoder_Y = preprocessing.LabelEncoder()
Y = labelencoder_Y.fit_transform(Y)
```

IMPLEMENTATION

- **Training the ML Model:**
  - Train various machine learning classifiers, including MLPClassifier, AdaBoostClassifier, GradientBoostingClassifier, GaussianNB,LinearDiscriminantAnalysis, and Quadratic Discriminant Analysis.
  - Evaluate each classifier's performance using accuracy and log loss metrics.

```python
from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier, BaggingClassifier, GradientBoostingClassifier, AdaBoostClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, log_loss
import pandas as pd

# Define classifiers
classifiers = [
    RandomForestClassifier(),
    ExtraTreesClassifier(),
    BaggingClassifier(),
    GradientBoostingClassifier(),
    AdaBoostClassifier(),
    LogisticRegression(),
    GaussianNB(),
    DecisionTreeClassifier(),
    MLPClassifier(
        activation='tanh',
        solver='lbfgs',
        early_stopping=False,
        hidden_layer_sizes=(40, 10, 10, 10, 10, 2),
        random_state=1,
        batch_size='auto',
        max_iter=13000,
        learning_rate_init=1e-5,
        tol=1e-4,
    )
]
```

```
RandomForestClassifier
****Results****
Accuracy: 99.9407%
Log Loss: 0.00951316609950457
==============================
ExtraTreesClassifier
****Results****
Accuracy: 99.9605%
Log Loss: 0.008299058581697005
==============================
BaggingClassifier
****Results****
Accuracy: 99.8155%
Log Loss: 0.0128399591000116
==============================
GradientBoostingClassifier
****Results****
Accuracy: 91.3954%
Log Loss: 0.31111460107876754
==============================
AdaBoostClassifier
****Results****
Accuracy: 79.8261%
Log Loss: 0.6731792683873681
```

RESULTS

Classifier	Accuracy (%)	Log Loss
RandomForestClassifier	99.94	0.0095
ExtraTreesClassifier	99.96	0.0083
BaggingClassifier	99.82	0.0128
GradientBoostingClassifier	91.40	0.3111
AdaBoostClassifier	79.83	0.6732
LogisticRegression	69.19	0.5859
GaussianNB	57.77	2.1210
DecisionTreeClassifier	99.70	0.1024
MLPClassifier	94.85	0.1786

RESULTS

1. **RandomForestClassifier**, **ExtraTreesClassifier**, and **BaggingClassifier** demonstrate remarkable accuracy above 99%, suggesting strong predictive power. They also exhibit low log loss, making them highly suitable for sepsis prediction.
2. **DecisionTreeClassifier** is another high-accuracy model, but it has a moderate log loss. It can be a practical choice if you prioritize interpretability.
3. **GradientBoostingClassifier** offers decent accuracy, but it comes with a log loss of 0.3111, indicating room for improvement.
4. **AdaBoostClassifier** achieves moderate accuracy but has a relatively higher log loss, which may be a drawback in critical medical applications.
5. **LogisticRegression**, while providing reasonable accuracy, has a moderate log loss. It can be an acceptable choice, considering the trade-off between accuracy and log loss.

RESULTS

6. **GaussianNB** has the lowest accuracy and the highest log loss, making it less suitable for sepsis prediction in this context.
7. **MLPClassifier** offers a balanced performance with good accuracy and log loss, making it a viable choice for sepsis prediction.

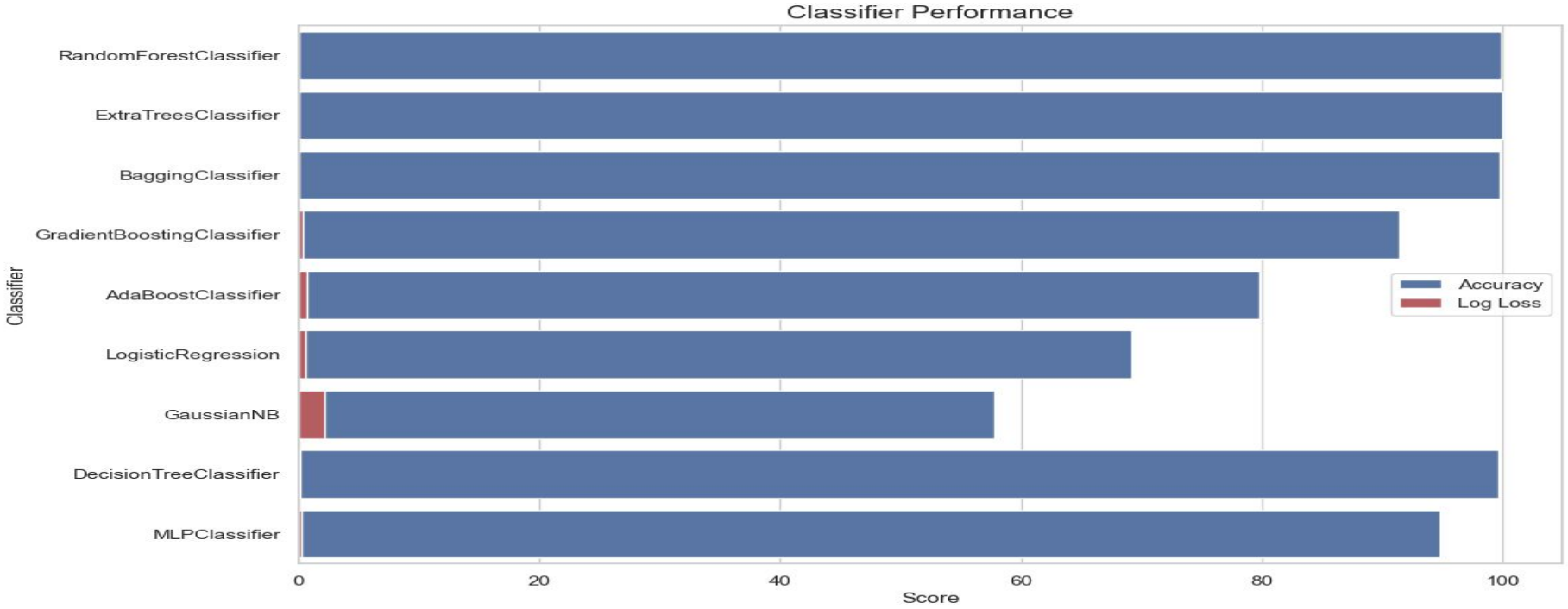
- In the final phase of the project, we will select the top 5 classifiers based on their accuracy performance, and then generate predictions using majority voting from five models.
- The final predicted class will be determined by the majority consensus among these classifiers.
- This approach leverages the combined decision-making power of the top-performing models to enhance the reliability of the final prediction for sepsis detection in the project.

DISCUSSION

- Early sepsis detection is crucial for healthcare, and our ensemble learning models show potential in reducing mortality and improving patient outcomes.
- Model generalization and real-world testing are essential to ensure practicality and reliability in clinical settings, considering variations across patient populations and healthcare facilities.
- Interpretability of ensemble models, often seen as "black box," is a concern in healthcare. We need to address this by exploring methods for model transparency.
- Ethical and regulatory considerations, including patient privacy and data security, must be addressed when implementing sepsis prediction models in healthcare.
- Collaboration with healthcare authorities is vital for seamless integration into clinical workflows while safeguarding patient rights and data privacy.
- Successful integration into healthcare practices requires careful planning, collaboration, and ongoing refinement.

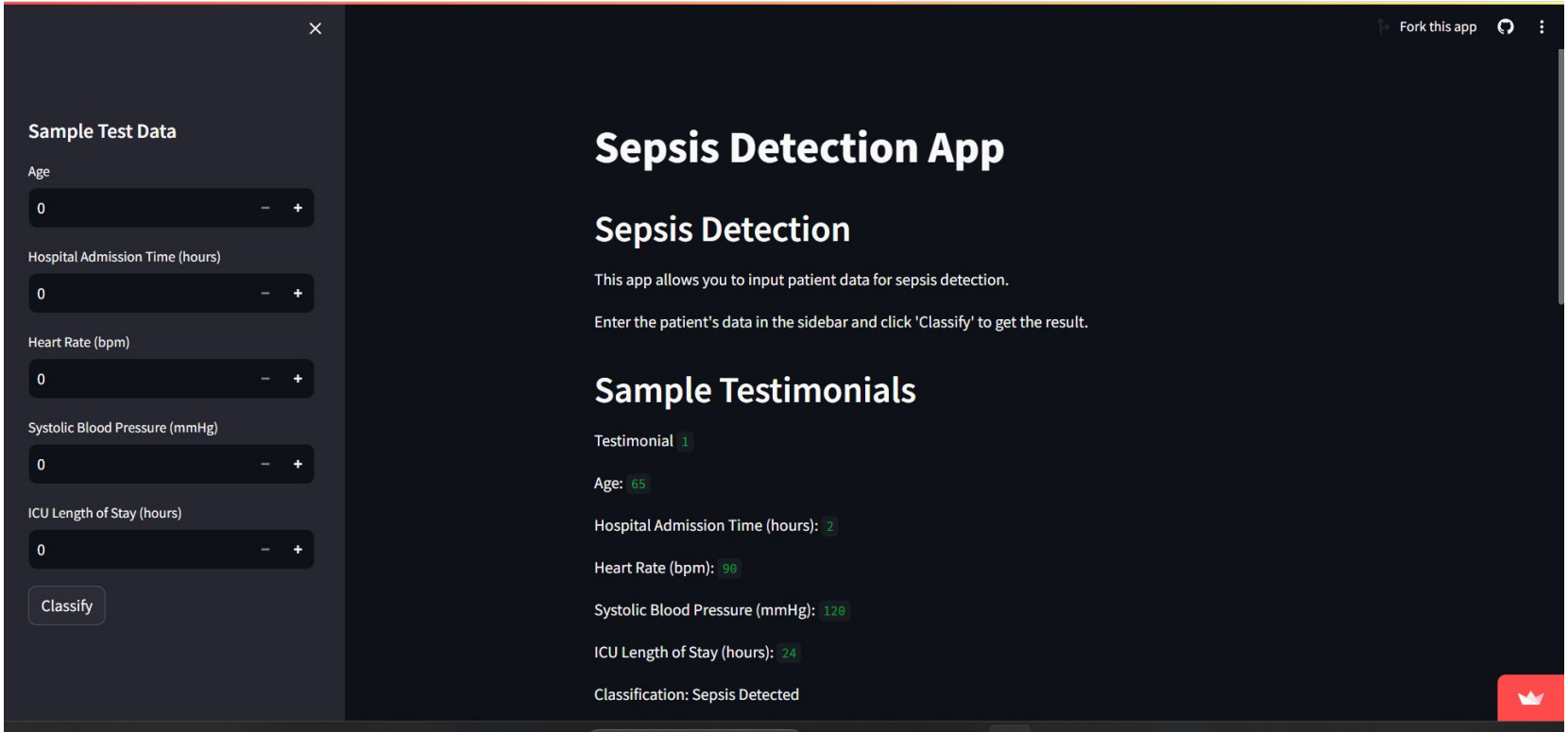
OUTPUT SCREENSHOTS

Evaluated various machine learning classifiers to predict sepsis based on the balanced dataset. The evaluation metrics include accuracy and log loss.



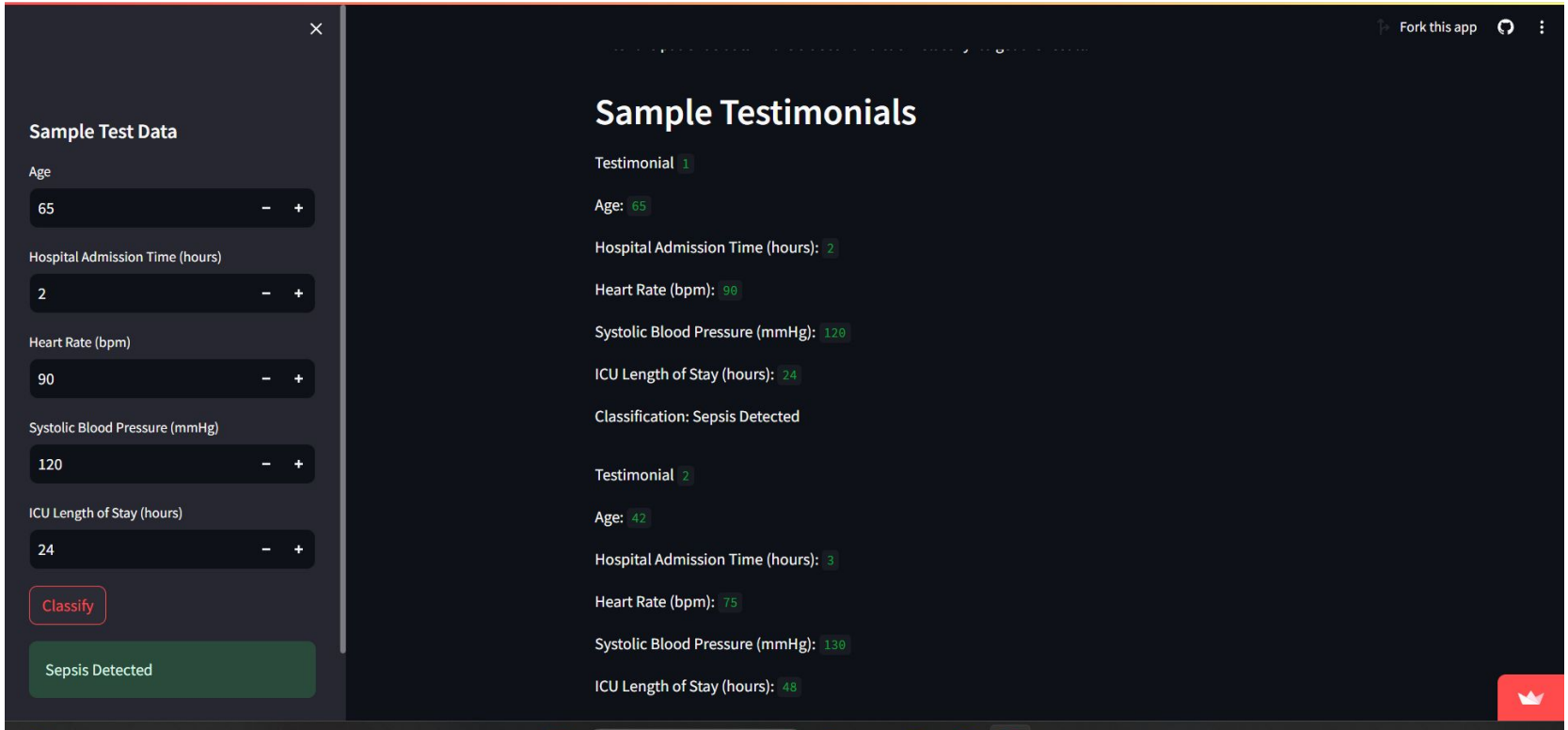
OUTPUT SCREENSHOTS

Front End Implementation



OUTPUT SCREENSHOTS

Front End Implementation



REFERENCES

1. Shankar-Hari, M., Phillips, G. S., Levy, M. L., Seymour, C. W., Liu, V. X., Deutschman & Singer, M. (2016). Developing a new definition and assessing new clinical criteria for septic shock: For the third international consensus definitions for sepsis and septic shock (sepsis-3). Jama, 315(8), 775–787.
2. McGregor C. Improving time to antibiotics and implementing the "Sepsis 6". BMJ Qual Improv Rep. 2014;2(2):u202548.w1443. Published 2014 Jan 14. doi:10.1136/bmjquality.u202548.w1443
3. Reinhart, K., Daniels, R., Kissoon, N., Machado, F. R., Schachter, R. D. & Finfer, S. (2017). Recognizing sepsis as a global health priority—A WHO resolution. New England Journal of Medicine, 377(5), 414-417.
4. Zimlichman, E., Henderson, D., Tamir, O., Franz, C., Song, P., Yamin, C. K. & Bates, D. W. (2013). Health care–associated infections. JAMA Internal Medicine, 173(22), 2039-2046.
5. Rudd, K. E., Johnson, S. C., Agesa, K. M., Shackelford, K. A., Tsoi, D., Kievlan & Lozano, R. (2020). Global, regional, and national sepsis incidence and mortality, 1990–2017: Analysis for the Global Burden of Disease Study. The Lancet, 395(10219), 200-211.
6. Sendak, M. P., Gao, M., Nichols, M., Lin, W., & Buchman, T. G. (2018). Recovery of sepsis patients' health-related quality of life. Critical Care, 22(1), 1-8.
7. B. C. Srimedha, R. Naveen Raj and V. Mayya, "A Comprehensive Machine Learning Based Pipeline for an Accurate Early Prediction of Sepsis in ICU," in IEEE Access, vol. 10, pp. 105120-105132, 2022, doi: 10.1109/ACCESS.2022.3210575.
8. Reyna, M., Josef, C., Jeter, R., Shashikumar, S., Moody, B., Westover, M. B., Sharma, A., Nemati, S., & Clifford, G. D. (2019). Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. PhysioNet.
9. "The PhysioNet/Computing in Cardiology Challenge 2019." Available: https://physionet.org/content/challenge-2019/1.0.0/. Accessed: 10-Jan-2023.
10. Glickman, S. W., Cairns, C. B., Otero, R. M., Woods, C. W., Tsalik, E. L., Langley, R. J., & Fowler Jr, V. G. (2010). Disease progression in hemodynamically stable patients presenting to the emergency department with sepsis. Academic Emergency Medicine, 17(4), 383–390.

REFERENCES

11. Shapiro, N., Howell, M. D., Bates, D. W., Angus, D. C., Ngo, L., & Talmor, D. (2006). The association of sepsis syndrome and organ dysfunction with mortality in emergency department patients with suspected infection. Annals of Emergency Medicine, 48(5), 583–590.
12. Fleischmann, C., Scherag, A., Adhikari, N. K., Hartog, C. S., Tsaganos, T., Schlattmann & Reinhart, K. (2016). Assessment of global incidence and mortality of hospital-treated sepsis. Current estimates and limitations. American Journal of Respiratory and Critical Care Medicine, 193(3), 259–272.
13. Angus, D. C., Linde-Zwirble, W. T., Lidicker, J., Clermont, G., Carcillo, J., & Pinsky, M. R. (2001). Epidemiology of severe sepsis in the United States: Analysis of incidence, outcome, and associated costs of care. Critical Care Medicine, 29(7), 1303–1310.
14. Mayr, F. B., Yende, S., & Angus, D. C. (2014). Epidemiology of severe sepsis. Virulence, 5(1), 4–11.
15. Liu, V. X., Fielding-Singh, V., Greene, J. D., Baker, J. M., Iwashyna, T. J., Bhattacharya & Escobar, G. J. (2017). The timing of early antibiotics and hospital mortality in sepsis. American Journal of Respiratory and Critical Care Medicine, 196(7), 856–863.
16. Kumar, A., Roberts, D., Wood, K. E., Light, B., Parrillo, J. E., Sharma & Cheang, M. (2006). Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. Critical Care Medicine, 34(6), 1589–1596.
17. Prescott, H. C., & Iwashyna, T. J. (2019). Improving sepsis treatment by embracing diagnostic uncertainty. Annals of the American Thoracic Society, 16(4), 426–429.
18. Henry, K. E., Hager, D. N., Pronovost, P. J., & Saria, S. (2015). A targeted real-time early warning score (trewscore) for septic shock. Science Translational Medicine, 7(299), 299ra122–299ra122.
19. Nemati, S., Holder, A., Razmi, F., Stanley, M. D., Clifford, G. D., & Buchman, T. G. (2018). An interpretable machine learning model for accurate prediction of sepsis in the ICU. Critical Care Medicine, 46(4), 547.