DETECTION OF SEPSIS USING VITAL SIGNS

A PROJECT REPORT

Submitted by

ALAPATI DEVI ANUSHA (111620243003)
PUNURU DIVYA HARSHTHA (111620243041)
VADLAMUDI LAKSHMI TEJASWINI (111620243056)
BATTEPATI DHARANI (111620243006)

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution)
R.S.M. Nagar, Puduvoyal 601206

DECEMBER 2023

R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

R.S.M. Nagar, Puduvoyal 601206

BONAFIDE CERTIFICATE

Certified that this project report **"DETECTION OF SEPSIS USING VITAL SIGNS"** is the bonafide work of **ALAPATI DEVI ANUSHA (111620243003), PUNURU DIVYA HARSHITHA (111620243041), VADLAMUDI LAKSHMI TEJASWINI (111620243056), BATTEPATI DHARANI (111620243006)** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. B. PRATHUSHA LAKSHMI B.E, M.E, Ph.D.	**MS.A.AKHILA**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Artificial Intelligence and Data Science,	Artificial Intelligence and Data Science,
R.M.K. College Of Engineering and Technology,	R.M.K. College Of Engineering and Technology,
R.S.M. Nagar,	R.S.M. Nagar,
Puduvoyal-601206.	Puduvoyal-601206.

ACKNOWLEDGEMENT

We earnestly portray our sincere gratitude and regards to our beloved **Founder Chairman Thiru. R. S. MUNIRATHINAM** for giving us the infrastructure for conducting the project work and our Chairperson **Tmt. MANJULA MUNIRATHINAM** for her blessings. We also thank our **Vice Chairman, Thiru. R. M. KISHORE** and our **Director, Thiru. R. JOTHI NAIDU,** for their constant support and affection showntowards us throughout the course.

We are extremely thankful to our **Principal**, **Dr. K.RAMAR,** for being the source of inspiration in this college.

We reveal our sincere thanks to our **Professor** and **Head of the Department**, Artificial Intelligence and Data Science**, Dr. B. PRATHUSHA LAXMI,** for her commendablesupport and encouragement for the completion of our project.

We wish to record our thanks to our project supervisor **MS.A.AKHILA** for his valuable guidance and support during each stage of our project.

We take this opportunity to extend our thanks to all the faculty members of Department of Artificial Intelligence and Data Science, parents and friends for their care and support during the crucial times of the completion of our project.

ABSTRACT

Sepsis, a life-threatening condition, demands early detection for effective intervention. This study investigates the potential of vital signs as predictive markers for sepsis onset. Through extensive literature review and statistical analysis, we assess the correlation between key vital signs (such as heart rate, respiratory rate, temperature, and blood pressure) and the development of sepsis. Data from diverse patient cohorts are analyzed using machine learning algorithms to establish predictive models. Our findings reveal significant associations between specific vital sign deviations and the onset of sepsis, suggesting their utility as early indicators. Furthermore, the developed predictive models demonstrate promising accuracy, sensitivity, and specificity in identifying impending sepsis based solely on vital sign patterns. Implementing these models in clinical settings could aid healthcare providers in timely intervention, potentially reducing sepsis-related morbidity and mortality rates. This research emphasizes the critical role of continuous vital sign monitoring in proactive sepsis management strategies.

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE
	ABSTRACT	ii
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vi
1	**INTRODUCTION**	1
	1.1 GENERAL	1
	1.2 INTRODUCTION	1
	1.3 PROBLEM STATEMENT	2
	1.4 OBJECTIVE	2
	1.5 EXISTING SYSTEM	2
2	**PROPOSED METHODOLOGY**	
	2.1 GENERAL	5
	2.2 METHODOLOGY	5
	2.3 REQUIREMENTS	6
	2.3.1 SOFTWARE REQUIREMENTS	6
	2.3.1 HARDWARE REQUIREMENTS	6

	2.4 MODULES GENERAL	6
	2.5 DESIGN	13
	2.6 IMPLEMENTATION CODE	19
	2.7 SCREEN SHOTS	28
	2.8 TESTING	33
3	**CONCLUSION**	37
	REFERENCES	38

CHAPTER 1

INTRODUCTION

1.1 GENERAL

This chapter gives introduction to the problem, also includes the information about the existing system and their disadvantages proposed system and its advantages along with future enhancement.

1.2 INTRODUCTION

Sepsis, a life-threatening condition stemming from an exaggerated response to infection, poses a critical challenge in healthcare. Rapid detection of sepsis onset is pivotal for effective intervention and improved patient outcomes. Vital signs—such as heart rate, respiratory rate, temperature, and blood pressure—serve as fundamental parameters routinely assessed in clinical settings. Recent technological advancements have fueled interest in leveraging these routinely measured vital signs as potential indicators for early identification of sepsis. This project focuses on exploring the intricate relationship betweendeviations in vital signs and the incipient stages of sepsis.

Successfully identifying early warning signs through these models could equip healthcare providers with a proactive tool for timely interventions. Ultimately, this approach seeks to mitigate the burden of sepsis-related complications and mortality rates, emphasizing the crucial role of real-time monitoring and predictive analytics in optimizing patient care strategies.

1.3 PROBLEM STATEMENT

Sepsis is an deadly disease that occurs due to the severe response by the body towards the disease. Early prediction of sepsis can avoid severe consequences. Therefore, detection of sepsis is done.

1.4 OBJECTIVE

The primary aim is early sepsis detection via vital signs data, crucial for life-saving interventions before its onset. Predicting the % likelihood of sepsis occurrence using lab values, vital signs, and patient data on a website is the core objective.

1.5 EXISTING METHODS

The Sequential Organ Failure Assessment (SOFA) score assesses organ dysfunction in sepsis patients based on several physiological parameters. The quick Sepsis-related Organ Failure Assessment (qSOFA) identifies patients at risk of poor outcomes using altered mental status, low blood pressure, and rapid breathing rate. Machine learning models like Random Forest or Logistic Regression analyze vital signs and clinical data to predict sepsis onset.

1.6 LITERATURE SURVEY

Title: Early detection of sepsis using machine learning techniques.

Author: Federica Briano

Year: 2021

Description: Sepsis, a leading global cause of mortality, prompts a surge in interest in machine learning for event prediction in healthcare. This perspective addresses key aspects in using machine learning models for early sepsis detection: (i) the impact of varying sepsis definitions on model development; (ii) considerations on feature selection and availability; (iii) evaluating model performance and clinical relevance of output. While acknowledging AI's growing role in healthcare, pitfalls must be cautiously navigated. A multidisciplinary approach is crucial for refining machine learning's application in sepsis recognition, potentially enhancing medical decision- making for this intricate syndrome in the future.

Title: A Machine Learning Model for Early Prediction and Detection of Sepsis in Intensive Care units.

Author: Pushpendra Singh

Year: 2022

Description :

Sepsis poses a significant healthcare challenge, impacting ICU patients and escalating mortality rates due to delayed treatment and resource constraints. Manual and biomarker-based models exist but lack full automation and accuracy. This paper proposes a machine learning-based early sepsis detection model using ICU patient data.Various models (RF, LR, SVM, NB, ensemble of these, XGBoost, and a proposed ensemble) are evaluated using clinical data. The proposed ensemble, combining SVM, RF, NB, LR, and XGBoost, shows superior performance with a balanced accuracy of

0.96 compared to other models (RF: 0.90, LR: 0.73, SVM: 0.93, NB: 0.74, ensemble: 0.94, XGBoost: 0.95). Experimental results affirm the superiority of the proposed ensemble model for early sepsis prediction.

Title: Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare

Author:Kim Hout Goh

Description :

Sepsis, a major hospital cause of death, poses challenges in early detection due to its resemblance to less severe conditions. The SERA AI algorithm, using structured and unstructured clinical data, predicts sepsis onset with high accuracy (AUC 0.94, sensitivity 0.87, specificity 0.87) 12 hours in advance. Compared to physician predictions, SERA increases early sepsis detection by up to 32% and decreases false positives by up to 17%. Mining unstructured clinical notes notably enhances accuracy compared to using only clinical measures, improving early warning for sepsis onset 12-48 hours in advance.improve the algorithm’s accuracy compared to using only clinical measures for early warning 12 to 48 hours before the onset of sepsis.

CHAPTER 2

PROPOSED METHODOLOGY

2.1 GENERAL

This chapter gives elaborate details of the proposed approach and the different aspects involved in the proposed approach.

2.2 METHODOLOGY

Utilize a dataset comprising vital signs, laboratory values, and patient information for feature selection and preprocessing. Implement machine learning algorithms such as xboost for predictive modeling, validating the model's performance through cross-validation and evaluation metrics. Deploy the finalized model for real-time sepsis prediction based on incoming patient data.

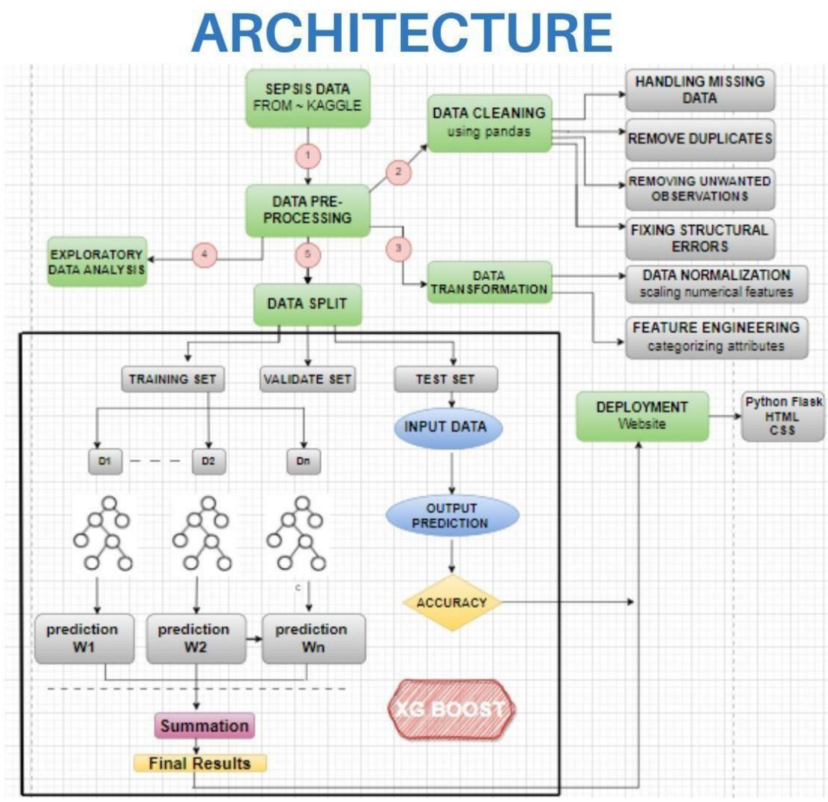

Fig 2.1 : System Architecture

2.3 REUIREMENTS

To implement this project the below software and hardware components are required

2.3.1 Software Requirements

- Language: Python
- IDE: Jyupter NoteBook, vscode

2.3.2 Hardware Requirements

- PC with a minimum of 4GB RAM
- Windows 7 or higher Operating System

2.4 MODULES

- **Pandas**: Offers data manipulation and analysis tools in Python, facilitating handling of structured data through dataframes and operations like filtering, cleaning, and merging datasets efficiently.

- **NumPy**: Provides support for numerical operations and arrays in Python, offering a powerful library for mathematical computations and operations on multidimensional arrays and matrices.

- **Matplotlib**: A versatile plotting library in Python enabling the creation of various types of visualizations like line plots, histograms, and scatter plots for data exploration and presentation.

- **Seaborn**: Built on top of Matplotlib, Seaborn offers enhanced statistical graphics, providing a high-level interface for creating attractive and informative statistical graphics.

- **Scikit-learn**: A comprehensive machine learning library in Python, offering tools for various tasks like classification, regression, clustering, and model selection through a consistent API and user-friendly interface.

2.3.3 Technology Requirements

2.3.3.1 Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

Xgboost algorithm

XGBoost stands as an optimized and efficient machine learning algorithm designed to enhance gradient boosting. This algorithm employs a unique ensemble learning method using decision trees, iterating and improving on previous models' weaknesses by focusing on misclassified instances. XGBoost is known for its speed, accuracy, and ability to handle large datasets, leveraging techniques like regularization to prevent overfitting andparallel computing for faster processing. Its popularity stems from its effectiveness in various tasks such as classification, regression, and ranking.

Why did we use Xgboost?

1. **High Accuracy:** XGBoost is known for its exceptional predictive performance. In sepsis detection where early and accurate identification is crucial, XGBoost's ability to handle complex relationships within the data and its robustness in handling large datasets make it a suitable choice for achieving high accuracy.

2. **Handling Imbalanced Data:** Sepsis detection datasets often exhibit class imbalance, with fewer instances of sepsis cases compared to non-sepsis cases. XGBoost's capability to handle imbalanced data through weighted classes or sample weights helps in training models that are more sensitive to detecting the rare but critical instances of sepsis.

3. **Regularization Techniques:** XGBoost offers built-in regularization techniques that help prevent overfitting and enhance generalization performance. This is crucial in healthcare applications like sepsis detection, where the model needs to generalize well to new, unseen patient data.

2.3.3.2 WHAT IS STREAMLIT?

Streamlit, on the other hand, is a powerful Python library facilitating the creation of interactive web applications for data exploration and visualization. It simplifies the development process by enabling users to convert data scripts into shareable web apps swiftly. With Streamlit, developers can create intuitive user interfaces directly from Python scripts, allowing seamless integration of data visualizations, machine learning models, and user inputs. Its simplicity and flexibility make it an attractive tool for rapid prototyping, enabling both data scientists and developers to create user-friendly applications without extensive web development .

Why did we use Streamlit?

1. **Easy Integration of Models:** Streamlit simplifies the integration of machine learning models into a user-friendly interface. For sepsis detection, this means healthcare professionals can interact with the predictive model seamlessly, inputting patient data and receiving predictions without needing expertise in coding or complex tools.

2. **Real-time Data Input:** In clinical settings, Streamlit's capability to handle real-time data input allows for immediate predictions based on the most current patient information. This feature is vital in the context of sepsis detection, where timely predictions can significantly impact patient outcomes.

3. **User-Friendly Interface:** Streamlit's user-friendly nature makes it accessible to healthcare professionals, enabling them to use the application without extensive technical knowledge. This ease of use is crucial in healthcare applications where quick and intuitive interfaces are paramount.

2.5 USE CASE DIAGRAM

The image presents a use case diagram centered around a hospital setting, focusing on sepsis prevention and patient care. It begins with an emphasis on health and the hospital's commitment to preventing sepsis through the slogan "WE CARE FOR YOURHEALTH - ARE YOU SEPSISIFIED?" The starting point is a healthy state, which branches into monitoring various vital signs such as respiratory rate, temperature, heart rate, pulse oximetry, and mean arterial pressure.

The diagram depicts the process of regularly checking these vital signs and health indicators in patients. Ultimately, the outcome of this use case is determining the patient's likelihood of developing severe sepsis. The diagram underlines the significance of early detection and intervention in preventing sepsis, as highlighted by the prominence of the word "SEPSIS" in the final section. Overall, it illustrates how hospitals utilize vital signs and health data to gauge the risk of severe sepsis and underscores the critical role of proactive monitoring and intervention in ensuring patienthealth.

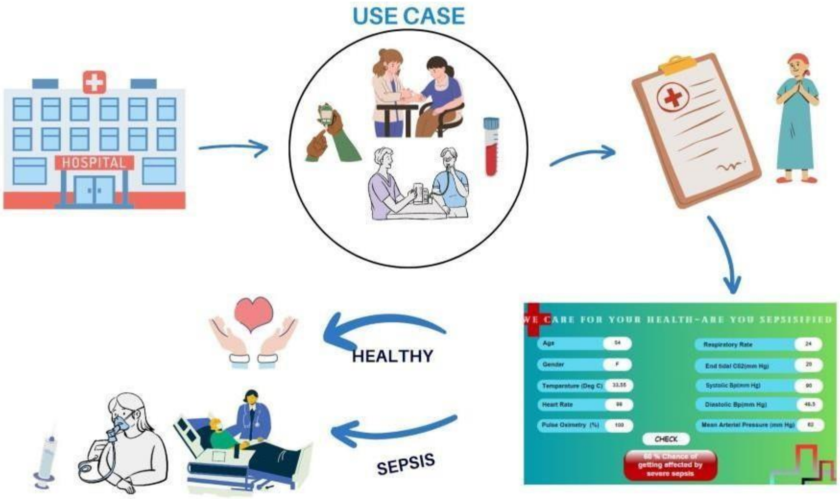

Fig 2.6: Use Case diagram

2.6 IMPLEMENTION CODE

```
import pandas as pd
import numpy as np
import xgboost as xgb
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Convert the data into DMatrix format for XGBoost
dtrain = xgb.DMatrix(X_train_oversampled, label=y_train_oversampled)
dtest = xgb.DMatrix(X_test, label=y_test)

# Define the XGBoost parameters
params = {
    "objective": "binary:logistic",
    "eval_metric": "logloss",
    "eta": 0.1,
    "max_depth": 6,
    "min_child_weight": 1,
    "gamma": 0,
    "subsample": 0.8,
    "colsample_bytree": 0.8,
    "scale_pos_weight": 1,
    "seed": 42
}

# Train the XGBoost model
num_rounds = 100
xgb_model = xgb.train(params, dtrain, num_rounds)

# Make predictions on the test set
y_pred = xgb_model.predict(dtest)
y_pred_binary = [1 if p >= 0.5 else 0 for p in y_pred]

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred_binary)
confusion_mat = confusion_matrix(y_test, y_pred_binary)

# Plot the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_mat, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted Labels')
```

```
plt.ylabel('True Labels')
plt.title('Confusion Matrix')
plt.show()

print("Accuracy:", accuracy)
print("Confusion Matrix:")
print(confusion_mat)

# ... (Your code for the user inputs and predictions)
# ... (Follow the steps outlined earlier for user inputs and predictions)
# ... (Ensure you have the necessary functions for processing user inputs)

# Finally, make predictions using the trained model on the user input data
# Get user input for age
user_age = int(input("Please enter your age: "))

# Get user input for HR
user_hr = int(input("Please enter your heart rate (HR): "))

# Get user input for O2Sat
user_o2sat = float(input("Please enter your oxygen saturation (O2Sat): "))

# Get user input for Temp
user_temp = float(input("Please enter your temperature (Temp): "))

# Get user input for BP
user_sbp = int(input("Please enter your systolic blood pressure (SBP): "))
user_dbp = int(input("Please enter your diastolic blood pressure (DBP): "))

# Get user input for Resp
user_resp = int(input("Please enter your respiration rate (Resp): "))

# Get user input for MAP
user_map = int(input("Please enter your mean arterial pressure (MAP): "))

# Get user input for FiO2
user_fio2 = float(input("Please enter your fraction of inspired oxygen (FiO2): "))

# Perform age categorization using the fe_new_age function
def fe_new_age(age):
    if age >= 60:
        return 'old'
    elif age < 10:
        return 'infant'
```

```
    else:
        return 'adult'

def fe_new_hr(hr, age):
    if (hr >= 70 and hr < 110) and (age < 10):
        return 'normal'
    elif (hr > 60 and hr < 100) and (age >= 10):
        return 'normal'
    elif ((hr < 70) or (age >= 110)) and (age < 10):
        return 'abnormal'
    elif (hr >= 100) and (age >= 10):
        return 'abnormal'
    else:
        return 'Missing'
def fe_new_temp(temp):
    if temp >= 36 and temp < 38:
        return 'normal'
    else:
        return 'abnormal'

def fe_new_bp(sbp,  dbp):
    if sbp < 90 and dbp < 60:
        return 'low'
    elif sbp >= 90 and sbp <= 120 and dbp >= 60 and dbp <= 80:
        return 'normal'
    elif sbp >= 120 and sbp <= 140 and dbp >= 80 and dbp <= 90:
        return 'elevated'
    elif sbp > 140 and dbp > 90:
        return 'high'
    else:
        return 'Missing'

def fe_new_resp(resp, age):
    if (resp >= 30 and resp <= 60) and (age < 1):
        return 'normal'
    elif (resp >= 24 and resp <= 40) and (age >= 1 and age <= 3):
        return 'normal'
    elif (resp >= 22 and resp <= 34) and (age >= 3 and age <= 6):
        return 'normal'
    elif (resp >= 18 and resp <= 30) and (age >= 6 and age <= 12):
        return 'normal'
    elif (resp >= 12 and resp <= 16) and (age >= 12 and age <= 18):
        return 'normal'
    elif (resp >= 12 and resp <= 20) and (age > 18):
```

```
        return 'normal'
    else:
        return 'abnormal'

def fe_new_map(map_value):
    if map_value >= 70 and map_value < 100:
        return 'normal'
    else:
        return 'abnormal'

def fe_new_fio2(fio2):
    if fio2 < 0.8:
        return 'normal'
    else:
        return 'abnormal'
user_age_category = fe_new_age(user_age)
# Perform HR categorization using the fe_new_hr function
user_hr_category = fe_new_hr(user_hr, user_age)

# Perform O2Sat categorization using the fe_new_o2sat function
user_o2sat_category = fe_new_o2sat(user_o2sat)

# Perform Temp categorization using the fe_new_temp function
user_temp_category = fe_new_temp(user_temp)

# Perform BP categorization using the fe_new_bp function
user_bp_category = fe_new_bp(user_sbp, user_dbp)

# Perform Resp categorization using the fe_new_resp function
user_resp_category = fe_new_resp(user_resp, user_age)

# Perform MAP categorization using the fe_new_map function
user_map_category = fe_new_map(user_map)

# Perform FiO2 categorization using the fe_new_fio2 function
user_fio2_category = fe_new_fio2(user_fio2)

# Create a DataFrame to hold the user's input with the same column names as your model's
input
user_gender = 0 # Change this to the actual gender input (0 for male, 1 for female)
user_input_df = pd.DataFrame({
    'new_age_old': [1 if user_age_category == 'old' else 0],
    'new_o2sat_abnormal': [1 if user_o2sat_category == 'abnormal' else 0],
    'new_o2sat_normal': [1 if user_o2sat_category == 'normal' else 0],
```

```
    'new_temp_abnormal': [1 if user_temp_category == 'abnormal' else 0],
    'new_temp_normal': [1 if user_temp_category == 'normal' else 0],
    'new_bp_elevated': [1 if user_bp_category == 'elevated' else 0],
    'new_bp_high': [1 if user_bp_category == 'high' else 0],
    'new_bp_low': [1 if user_bp_category == 'low' else 0],
    'new_bp_normal': [1 if user_bp_category == 'normal' else 0],
    'new_resp_abnormal': [1 if user_resp_category == 'abnormal' else 0],
    'new_resp_normal': [1 if user_resp_category == 'normal' else 0],
    'new_map_abnormal': [1 if user_map_category == 'abnormal' else 0],
    'new_map_normal': [1 if user_map_category == 'normal' else 0],
    'new_fio2_abnormal': [1 if user_fio2_category == 'abnormal' else 0],
    'new_fio2_normal': [1 if user_fio2_category == 'normal' else 0],
    'new_hr_abnormal': [1 if user_hr_category == 'abnormal' else 0],
    'Gender': [user_gender],  # Add the 'Gender' column to the user input DataFrame
})

user_input_dmatrix = xgb.DMatrix(user_input_df)

# Make predictions using the trained model
user_prediction = xgb_model.predict(user_input_dmatrix)

# Output the prediction (You can define a threshold here if it's a binary classification)
print("Severity Score for the User Input:", user_prediction[0])
```

2.7 SCREEN SHOTS

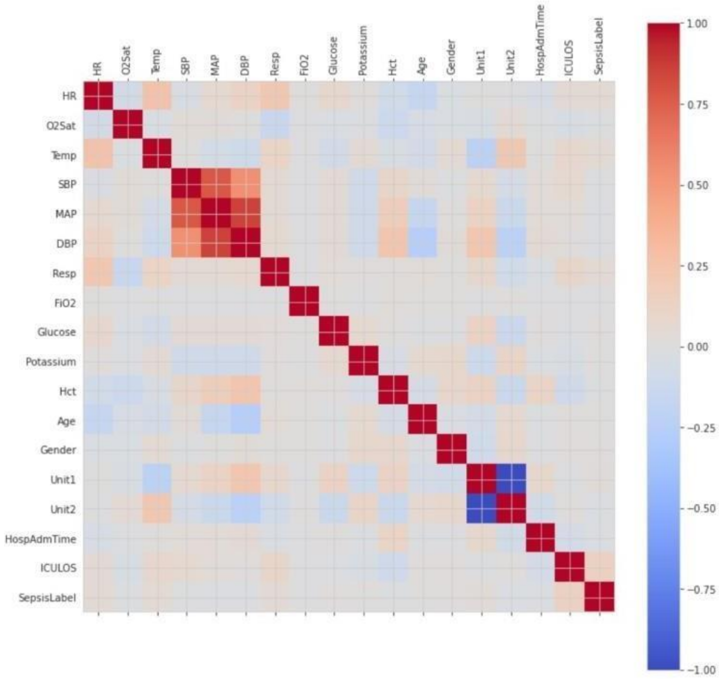

Fig 2.11: Heat map

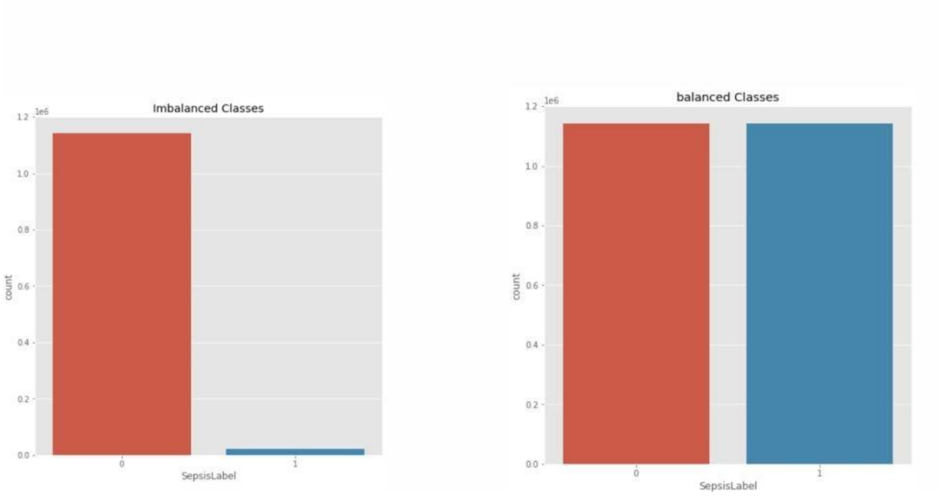

Fig 2.11: Balancing Imbalance Data

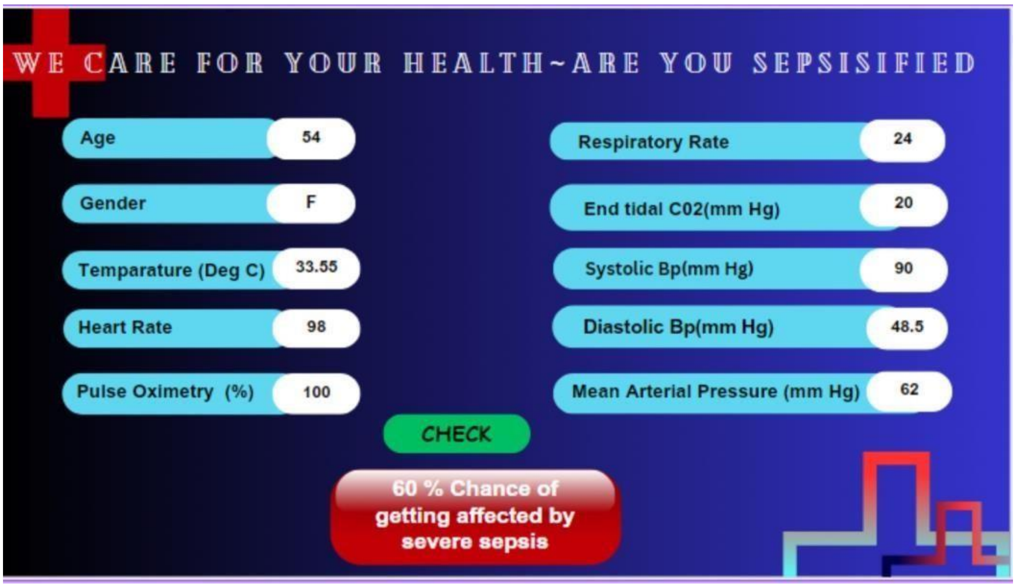

Fig 2.11: Final output screen shot

CHAPTER 3

CONCLUSION

The integration of XGBoost for sepsis detection with Streamlit's web interface presents an innovative healthcare solution. Leveraging XGBoost's robust machine learning capabilities, the system predicts sepsis likelihood based on vital signs and clinical data, offering valuable insights for healthcare providers. XGBoost's prowess in managing complex data relationships and handling imbalanced datasets ensures accurate predictions. Streamlit's user-friendly interface facilitates seamless input of patient data and real-time output, presenting the percentage chance of sepsis occurrence swiftly. This integrated approach empowers clinicians with a prompt and comprehensive tool for early sepsis detection. The fusion of predictive accuracy and intuitive interface streamlines decision- making, potentially improving patient outcomes and reducing sepsis-related complications. Ultimately, this technology holds promise in revolutionizing clinical practices by enabling timely interventions, potentially saving lives.

REFERNCES

[1.] Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., et al. (2016). The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). JAMA - Journal of the American Medical Association, 315(8), 801–810. https://doi.org/10.1001/jama.2016.0287

[2.] Seymour, C. W., Liu, V. X., Iwashyna, T. J., Brunkhorst, F. M., Rea, T. D., Scherag, A., et al. (2016). Assessment of Clinical Criteria for Sepsis: For the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). JAMA - Journal of the American Medical Association, 315(8), 762–774. https://doi.org/10.1001/jama.2016.0288