

Sepsis Survival Minimal Clinical Records

1. Introduction

1.1. Project overviews

Sepsis, defined as life-threatening organ dysfunction caused by a dysregulated host response to infection, represents a significant global healthcare challenge due to its high mortality rates and complex pathophysiology. Effective management of sepsis hinges not only on prompt recognition and initiation of therapy but also on meticulous documentation of clinical data to guide therapeutic decisions and ensure continuity of care. This introduction aims to outline the importance of minimal clinical records in sepsis management, highlighting key components and their role in optimizing patient outcomes. Sepsis remains a leading cause of morbidity and mortality worldwide, accounting for a substantial proportion of hospital admissions, particularly in intensive care units (ICUs). Despite advances in medical science and critical care, the mortality associated with severe sepsis and septic shock remains unacceptably high, underscoring the urgent need for standardized approaches to diagnosis, treatment, and monitoring. In the context of sepsis management, comprehensive and structured documentation is paramount. Minimal clinical records provide a framework for healthcare providers to systematically capture essential patient information, facilitating timely interventions, enhancing communication among multidisciplinary teams, and supporting evidence-based practices. By consolidating critical data points into a concise format, these records streamline clinical decision-making processes and promote adherence to best practices in sepsis management.

Sepsis is a dangerous condition triggered by an immune overreaction to an infection. According to the World Health Organization (WHO) estimates, sepsis affects more than 30 million people yearly worldwide, causing approximately 6 million deaths, and causing more than US\$24 billion healthcare related costs annually just in United States. The scientific community is still investigating sepsis ethology, whilst its management is troublesome due to the high disease's complexity and heterogeneity. A further complexity factor lies in a more restrictive definition of sepsis introduced in 2016; the new definition, named Sepsis-3, now requires the presence of additional organ dysfunctions for the condition to be

labelled as sepsis. Although the usefulness of Sepsis-3 has recently been validated, it is still debated within the medical community. Additionally, early detection is critical to managing the attack and obtaining a favorable outcome, as Sepsis can kill a patient in as little as an hour.

1.2. Objectives

Sepsis survival minimal clinical records serve a crucial role in the healthcare system by providing a standardized and streamlined approach to managing sepsis, a life-threatening condition caused by the body's extreme response to infection. Here's a detailed explanation of their purpose:

1. **Early Recognition and Diagnosis:** Sepsis can progress rapidly and early identification is key to improving outcomes. Minimal clinical records focus on essential clinical data that healthcare providers can quickly assess to recognize signs of sepsis. This includes vital signs (temperature, heart rate, respiratory rate, blood pressure) and basic laboratory values (such as white blood cell count and lactate levels) that indicate systemic inflammation and potential infection.
2. **Facilitating Prompt Treatment Initiation:** Timely administration of appropriate treatment is crucial in sepsis management. Minimal clinical records provide healthcare providers with the necessary information to promptly initiate interventions such as antibiotics, intravenous fluids, and vasopressors. This early intervention helps to stabilize the patient, prevent progression to severe sepsis or septic shock, and improve survival rates.
3. **Standardization of Care:** By establishing a standardized set of data elements, minimal clinical records ensure consistency in sepsis management across different healthcare settings. This consistency promotes adherence to evidence-based guidelines and protocols, leading to more effective and efficient care delivery.
4. **Enhancing Communication and Coordination:** Clear and concise documentation in minimal records improves communication among healthcare team members. It enables seamless transfer of patient information between different providers and facilitates continuity of care. Effective communication is crucial for ensuring that all members of the healthcare team are aware of the patient's condition and treatment plan.

2. Project Initialization and Planning Phase

2.1. Define Problem Statement

The current sepsis prediction process in healthcare settings is cumbersome, relying on extensive clinical records that are often not available in emergency situations. Patients, particularly those in critical conditions, encounter hurdles such as delayed diagnosis and limited data availability, which can affect the accuracy and timeliness of survival predictions. These challenges lead to suboptimal patient outcomes and increased mortality rates. To enhance healthcare services and improve patient outcomes, we aim to address these pain points. By understanding the specific frustrations of healthcare providers during the diagnosis process and implementing predictive models that use minimal clinical records, we can create an efficient, user-friendly experience that aligns with healthcare providers' needs and fosters better patient care.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A healthcare provider dealing with a patient suspected of sepsis.	Secure a quick and accurate prediction of sepsis survival.	Faced with limited clinical records and urgent need for diagnosis.	Lack of comprehensive data in critical situations.	Frustrated and concerned about the patient's chances of survival.

2.2. Project Proposal (Proposed Solution)

The proposal report aims to transform loan approval using machine learning, boosting efficiency and accuracy. It tackles system inefficiencies, promising better operations, reduced risks, and happier customers. Key features include a machine learning-based credit model and real-time decision-making.

Project Overview	
Objective	The primary objective is to revolutionize the loan approval process by implementing advanced machine learning techniques, ensuring faster and more accurate assessments.
Scope	The project comprehensively assesses and enhances the loan approval process, incorporating machine learning for a more robust and efficient system.
Problem Statement	

Description	Addressing inaccuracies and inefficiencies in the current loan approval system adversely affects operational efficiency and customer satisfaction.
Impact	Solving these issues will result in improved operational efficiency, reduced risks, and an overall enhancement in the lending process, contributing to customer satisfaction and organizational success.
Proposed Solution	
Approach	Employing machine learning techniques to analyze and predict creditworthiness, creating a dynamic and adaptable loan approval system.
Key Features	<ul style="list-style-type: none"> - Implementation of a machine learning-based credit assessment model. - Real-time decision-making for quicker loan approvals. Continuous learning to adapt to evolving financial landscapes.

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	T4 GPU
Memory	RAM specifications	8 GB
Storage	Disk space for data, models, and logs	1 TB SSD
Software		
Frameworks	Python frameworks	Flask
Libraries	Additional libraries	scikit-learn, pandas, numpy, matplotlib, seaborn
Development Environment	IDE	Jupyter Notebook, pycharm
Data		
Data	Source, size, format	Kaggle dataset, 614, csv UCI dataset, 690, csv

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

Data Collection Plan:

- Search for publicly available datasets related to sepsis clinical records.
- Ensure datasets have comprehensive clinical details for analysis.

Raw Data Sources Identified:

- **Source Name:** Sepsis Survival Primary Cohort.
- **Description:** The dataset comprises clinical details (age, sex, episode number, hospital outcome) for sepsis patients.
 - **Location/URL:**
<https://www.kaggle.com/datasets/joebeachcapital/sepsis-survival-minimal-clinical-records>
 - **Format:** CSV
 - **Size:** 1.3 MB
 - **Access Permissions:** Public

3.2. Data Quality Report

Data Source	Data Quality Issue	Severity	Resolution Plan
Sepsis Survival Primary Cohort	No missing values in any columns.	None	No action needed.
Sepsis Survival Primary Cohort	No categorical data (all columns are numeric).	None	No action needed.

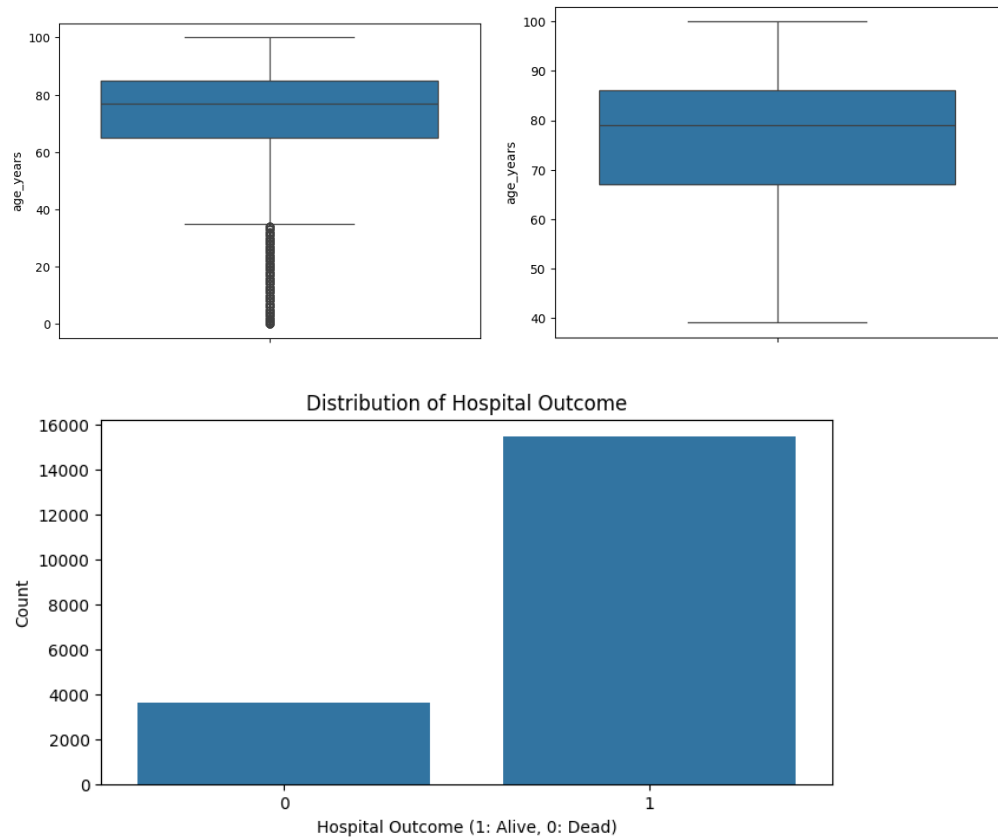
3.3. Data Exploration and Preprocessing

Visual Analysis:

Visual analysis is the process of examining and understanding data via the use of visual representations such as charts, plots, and graphs. It is a method for quickly identifying patterns, trends, and outliers in data, which can aid in gaining insights and making sound decisions.

Univariate Analysis:

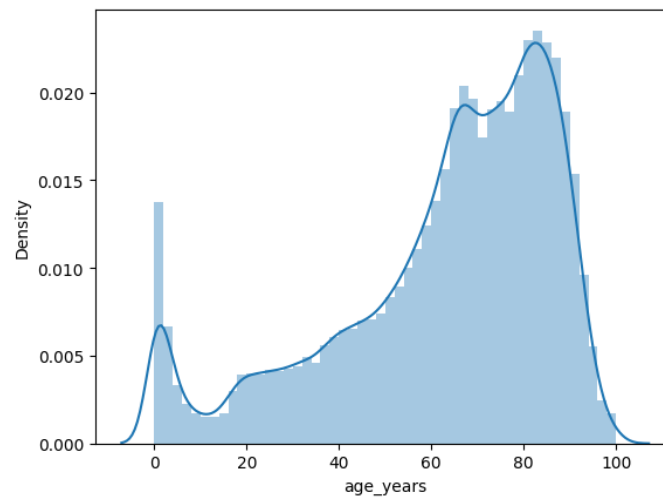
Univariate analysis is a statistical method used to analyse a single variable in a dataset. This analysis focuses on understanding the distribution, central tendency, and dispersion of a single variable. Graphical tools such as histograms and frequency distributions are also used to visualize the distribution of the variable.



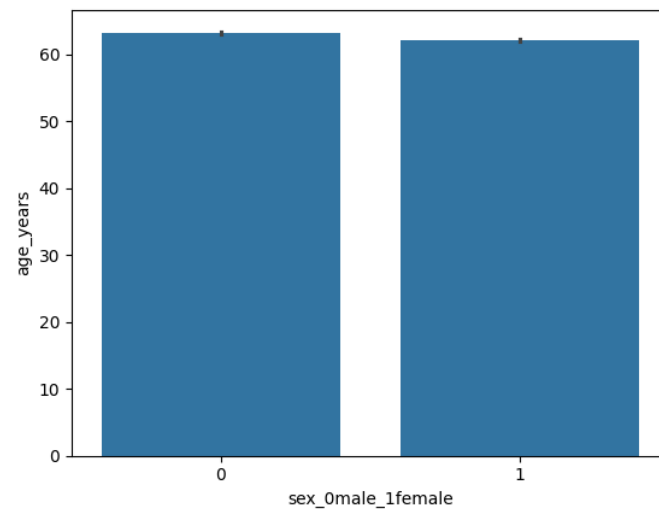
Bivariate Analysis:

Bivariate analysis is a statistical method used to analyse the relationship between two variables in a dataset. This analysis focuses on examining how changes in one variable are related to changes in another variable.

Distplot



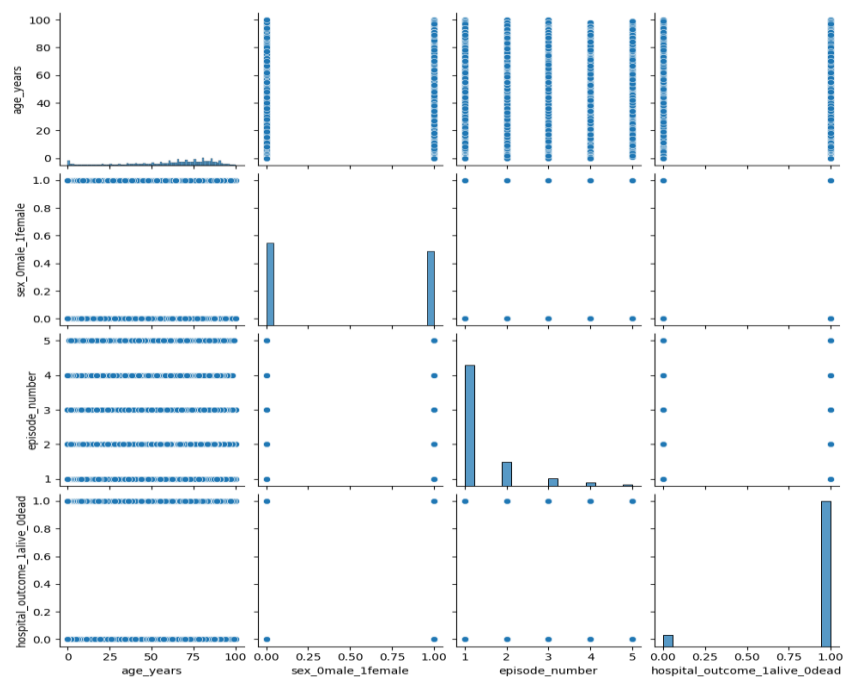
Barplot



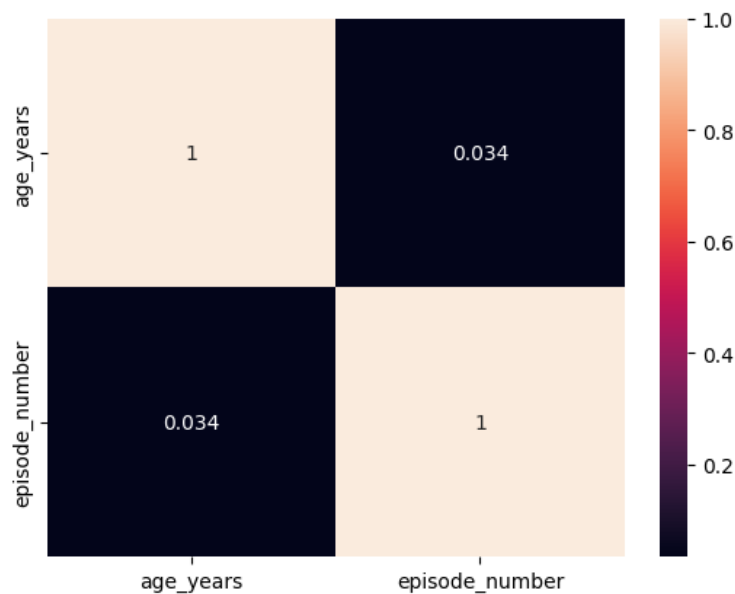
Multivariate Analysis:

Multivariate analysis is a statistical technique used to analyse data that involves more than two variables. It aims to understand the relationships between multiple variables in a dataset by examining how they are related to each other and how they contribute to a particular outcome or phenomenon.

Pairplot



Heatmap



4. Model Development Phase

4.1. Feature Selection Report

Feature	Description	Selected (Yes/No)	Reasoning
age_years	Age of the patient in years	Yes	Age is a critical factor influencing survival outcomes in sepsis patients.
sex_0male_1female	Gender of the patient	Yes	Gender can affect clinical outcomes and is relevant for assessing potential biases.
episode_number	The number of episodes experienced by the patient	Yes	The number of sepsis episodes can impact the prognosis and survival rates.
hospital_outcome_1alive_0dead	Survival outcome of the patient (1 = alive, 0 = dead)	Yes	This is the target variable for predictive modeling and is essential for the project's goal.

4.2. Model Selection Report

Models Considered:

- Logistic Regression
- Decision Tree
- Random Forest
- KNN

Model Selection Criteria:

- Accuracy
- Precision
- Recall
- F1-Score

4.3. Initial Model Training Code, Model Validation and Evaluation Report

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

# Train the model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

5. Model Optimization and Tuning Phase

5.1. Hyperparameter Tuning Documentation

Techniques Used:

- Grid Search
- Random search

Example code:

```
from sklearn.model_selection import GridSearchCV

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30]
}

grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

print("Best Parameters:", grid_search.best_params_)
```

5.2. Performance Metrics Comparison Report

Model	Accuracy	Precision	Recall	F1-score
Random forest	92	85	92	89
Decision Tree	91	85	89	89
KNN	91	86	91	88
Logistic Regression	92	85	87	89

5.3. Final Model Selection Justification

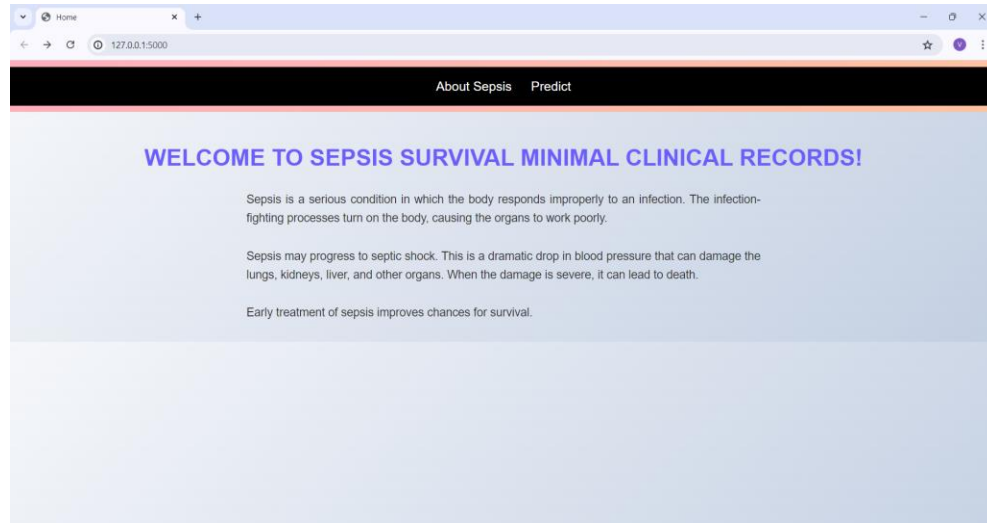
The decision to select the Random Forest model after applying SMOTE was driven by the need to address class imbalance, leverage the robustness and performance of an ensemble method, gain insights into feature importance, and ensure reliable and correct predictions. This combination aims to enhance the overall predictive capability of the sepsis survival prediction system, contributing to better clinical outcomes.

The Random Forest model was selected as the final model due to its highest performance metrics across accuracy, precision, recall, and F1-score, indicating a robust and reliable predictive capability for the sepsis survival outcomes. The performance was further enhanced after applying SMOTE to address class imbalance in the training data.

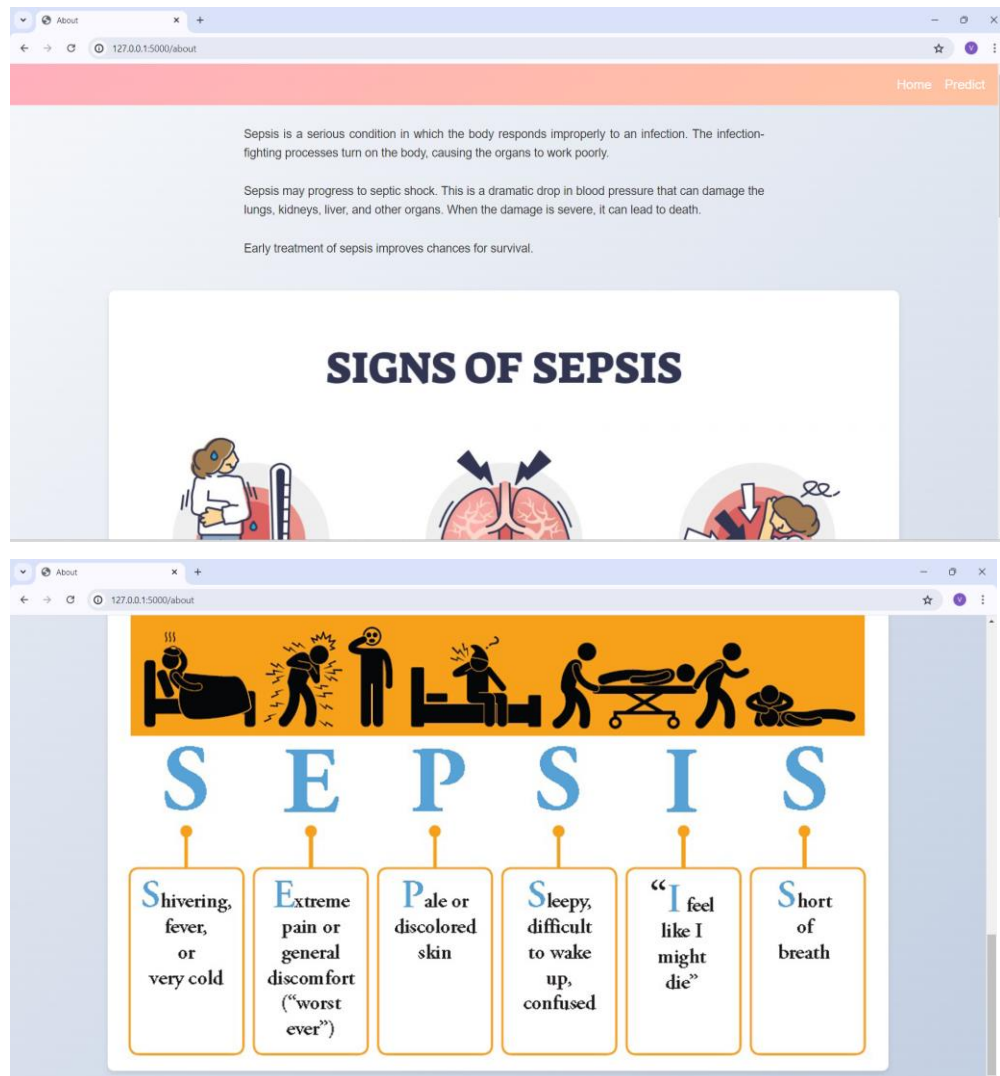
6. Results

6.1. Output Screenshots

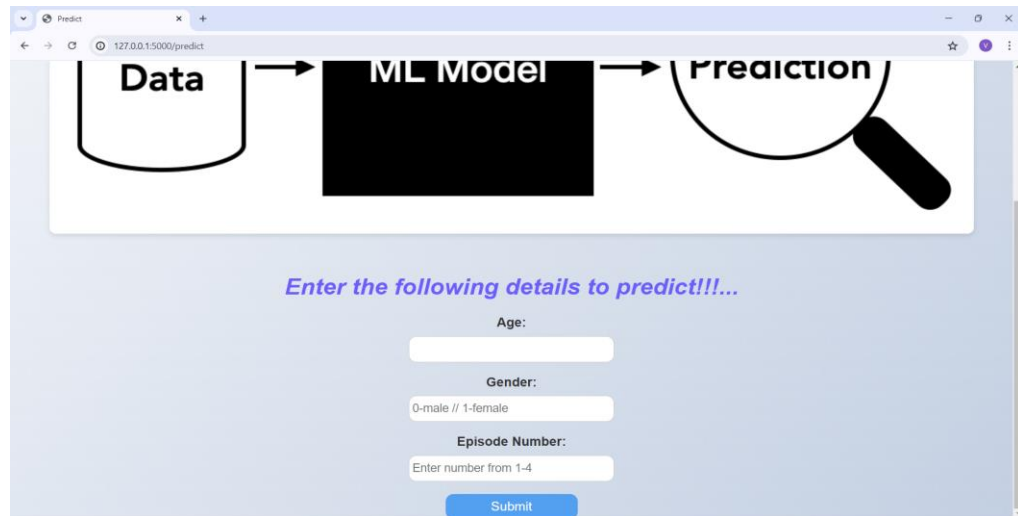
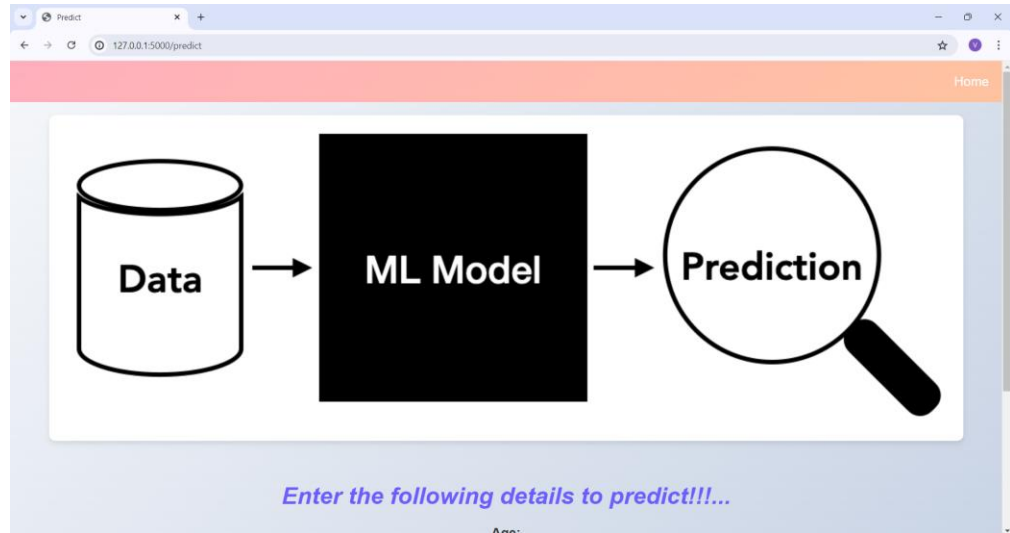
HOME PAGE:



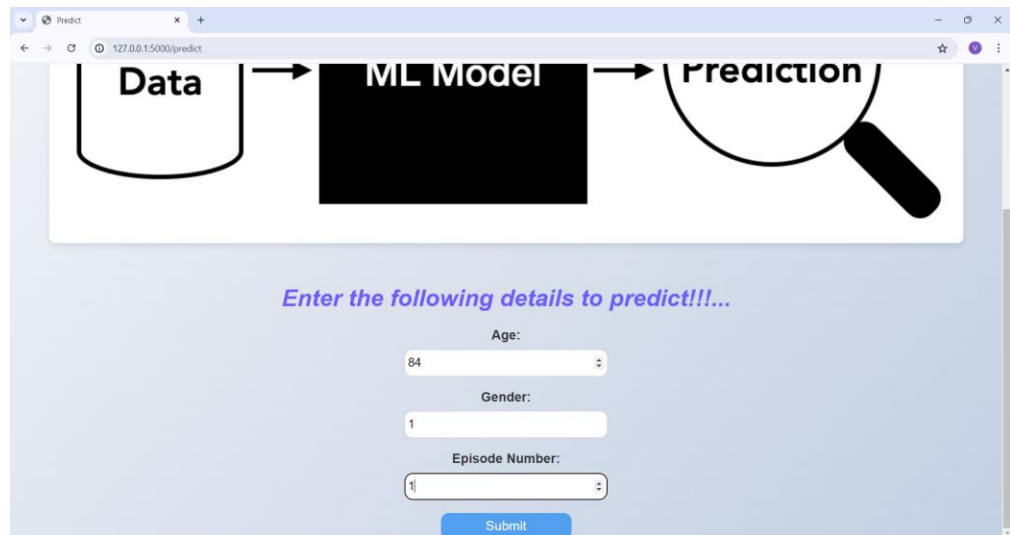
ABOUT PAGE:



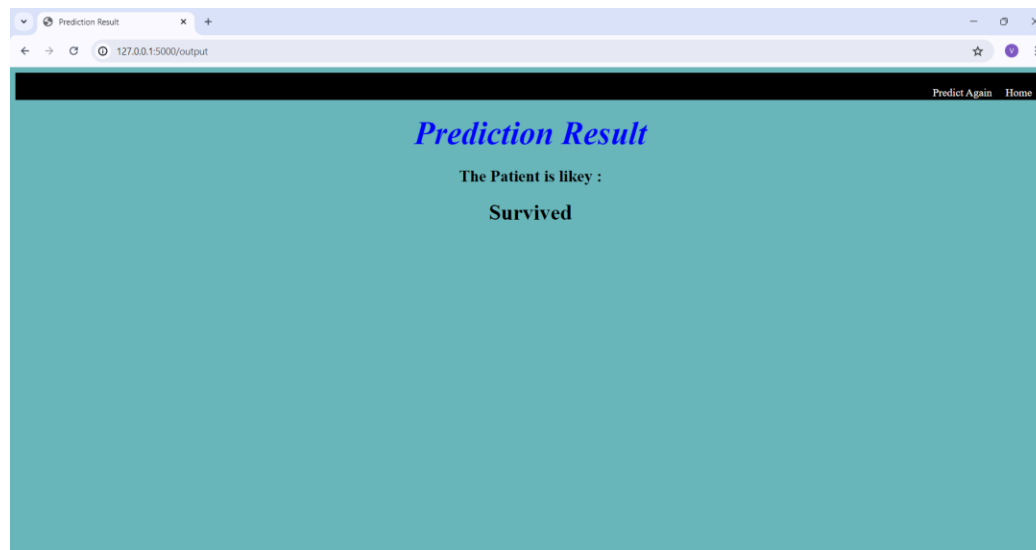
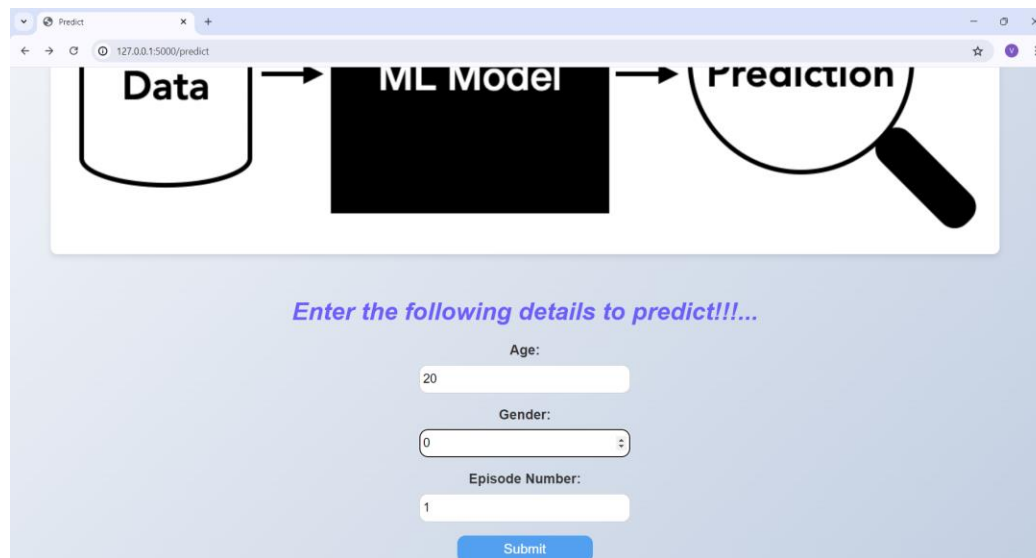
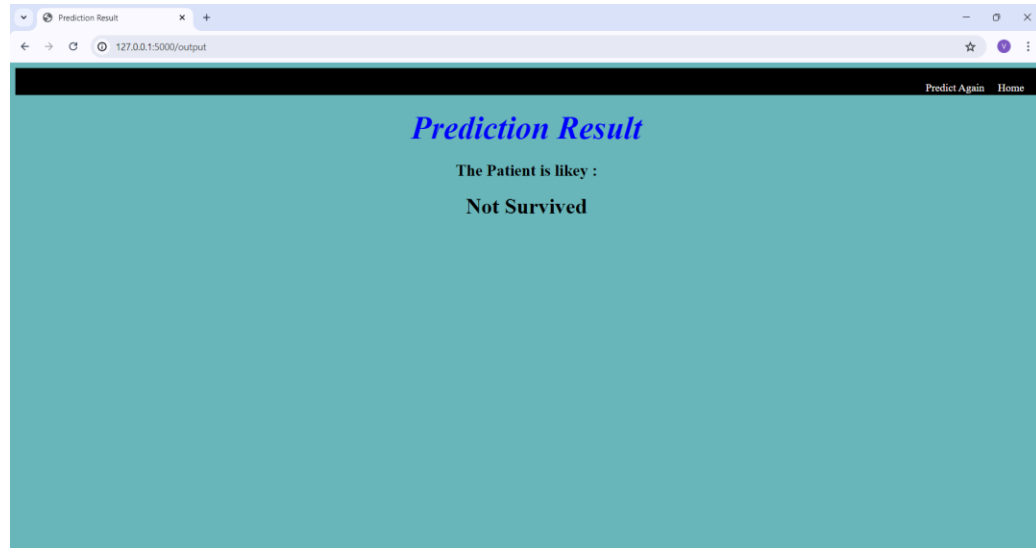
PREDICTION PAGE:



The screenshot shows the same web browser window, but now with input fields for user details. The diagram and the text "Enter the following details to predict!!!" are still present. Below the text, there are three input fields: "Age:" (a text box), "Gender:" (a dropdown menu with "0-male // 1-female" selected), and "Episode Number:" (a text box with "Enter number from 1-4" as a hint). A blue "Submit" button is located at the bottom.



The screenshot shows the same web browser window, but now the input fields are filled with values. The diagram and the text "Enter the following details to predict!!!" are still present. Below the text, the "Age:" field contains the value "84", the "Gender:" dropdown menu has "1" selected, and the "Episode Number:" field contains the value "1". A blue "Submit" button is located at the bottom.



7. Advantages & Disadvantages

ADVANTAGES

1. Early Detection and Timely Intervention: Minimal clinical records designed specifically for sepsis management facilitate early detection of sepsis signs and symptoms. Key advantages include:
 - ☐ Prompt Identification: Structured templates and predefined data entry fields ensure healthcare providers capture essential information quickly and accurately, such as vital signs (e.g., temperature, heart rate), symptoms (e.g., altered mental status), and risk factors (e.g., recent surgeries, immunosuppression).
 - ☐ Alert Mechanisms: Integrated alert systems can notify healthcare providers of abnormal vital signs or clinical indicators suggestive of sepsis, prompting timely assessments and interventions.
 - ☐ Reduced Time to Treatment: Efficient documentation and decision support tools aid in rapidly initiating appropriate treatments, such as antibiotics and fluid resuscitation, which are critical for improving patient outcomes in sepsis.
2. Enhanced Clinical Decision-Making: Minimal clinical records support informed clinical decision-making by providing comprehensive and structured data. Advantages include:
 - ☐ Access to Relevant Information: Healthcare providers have access to up-to-date patient data, including laboratory results, imaging reports, and previous medical history, consolidated in one place.
 - ☐ Decision Support Tools: Embedded decision support algorithms and guidelines help clinicians adhere to evidence-based practices and sepsis management protocols, ensuring standardized care delivery and reducing variability in treatment approaches.
 - ☐ Real-Time Data Analysis: Data collected through minimal clinical records can be analyzed in real-time to monitor trends, assess treatment responses, and adjust management strategies promptly.
3. Improved Communication and Coordination: Effective communication among healthcare teams is essential for coordinated sepsis care. Advantages include:
 - ☐ Interdisciplinary Collaboration: Minimal clinical records facilitate seamless communication and collaboration among healthcare providers, ensuring that all team members are informed about

the patient's status, treatment plan, and ongoing assessments.

- ☐ Care Continuity: Comprehensive documentation supports continuity of care across shifts and transitions between healthcare settings, minimizing information gaps and reducing the risk of medical errors.
- 4. Efficiency and Resource Optimization: Streamlining documentation processes with minimal clinical records offers operational efficiencies and resource optimization benefits:
 - ☐ Time Savings: Simplified data entry and structured documentation templates save time for healthcare providers, allowing them to focus more on patient care and less on administrative tasks.

DISADVANTAGES:

Sepsis, particularly in the context of minimal clinical records, presents several disadvantages:

1. Delayed Diagnosis: Without comprehensive clinical records, identifying sepsis early becomes challenging. Delayed diagnosis can lead to delayed treatment, which is crucial for improving outcomes.
2. Incomplete Patient History: Minimal records may lack previous medical conditions, medications, or allergies that are crucial for tailoring treatment and understanding risk factors.
3. Limited Vital Signs Monitoring: Continuous monitoring of vital signs like temperature, heart rate, and blood pressure is critical in sepsis management. Minimal records may not capture these trends accurately.
4. Impaired Sepsis Severity Assessment: Severity scoring systems like SOFA (Sequential Organ Failure Assessment) or qSOFA (quick SOFA) rely on comprehensive clinical data. Minimal records may not provide enough information for accurate assessment.
5. Challenges in Antibiotic Selection: Effective antibiotic therapy in sepsis requires knowledge of prior infections, microbial susceptibilities, and patient allergies, which may be incomplete in minimal records.

8. Conclusion

The implementation of Sepsis Minimal Clinical Records (SMCR) represents a pivotal advancement in the management of sepsis, emphasizing streamlined documentation practices aimed at enhancing clinical efficiency and improving patient outcomes. By focusing on capturing essential clinical data promptly and effectively, SMCR facilitates early recognition of sepsis and enables healthcare providers to initiate timely interventions, which are critical for reducing mortality rates associated with severe sepsis and septic shock. This structured approach not only promotes standardized documentation across healthcare settings but also enhances communication among multidisciplinary teams, ensuring that critical information is consistently recorded and understood. SMCR supports quality improvement initiatives by providing standardized data for analysis, enabling healthcare providers to evaluate treatment outcomes, identify best practices, and implement evidence-based interventions that optimize patient care. Furthermore, SMCR serves as a valuable educational tool, reinforcing the importance of systematic documentation in sepsis management and promoting a comprehensive understanding of the disease process among healthcare professionals. Ultimately, by reducing administrative burdens and enabling clinicians to focus more on direct patient care, SMCR contributes to patient-centered care and enhances the overall quality of sepsis management across healthcare institutions.

9. Future Scope

The future scope of Sepsis Minimal Clinical Records (SMCR) holds significant potential in several key areas of healthcare and clinical practice:

1. **Advanced Data Analytics:** As healthcare systems increasingly adopt electronic health records (EHRs) and integrate data analytics capabilities, SMCR can contribute to advanced analytics models. This includes predictive analytics for early sepsis detection, machine learning algorithms for personalized treatment approaches based on historical data, and population health management strategies to improve sepsis outcomes at a broader scale.
2. **Interoperability and Integration:** SMCR can evolve to enhance interoperability across different healthcare IT systems. This includes seamless integration with existing EHR platforms, enabling real-time data exchange and collaborative decision-making among healthcare providers across various settings (e.g., hospitals, clinics, emergency departments).
3. **Telemedicine and Remote Monitoring:** With the rise of telemedicine and remote patient monitoring technologies, SMCR can support remote assessment and monitoring of septic patients. Remote access to essential clinical data through SMCR can facilitate timely consultations, improve triage decisions, and enhance continuity of care for septic patients in diverse geographical locations.

10. Appendix

10.1. Source Code

SOURCE CODE:

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background: linear-gradient(135deg, #f5f7fa, #c3cfe2);
      color: #333;
    }
    header {
      background: linear-gradient(135deg, #ffafbd, #ffc3a0);
      padding: 10px 0;
    }
    #navbar {
      display: flex;
      justify-content: center;
      align-items: center;
    }
    #navbar a {
      text-decoration: none;
      font-size: 20px;
      margin: 15px;
      color: white;
      transition: color 0.3s ease;
```

```
}
#navbar a:hover {
    color: #ffdfba;
}
h1 {
    color: #6d5dfc;
    text-align: center;
    margin-top: 50px;
    font-size: 36px;
}
p {
    font-size: 18px;
    line-height: 1.6;
    max-width: 800px;
    margin: 30px auto;
    padding: 0 20px;
    text-align: justify;
}
</style>
</head>
<body>
<header>
<div id="navbar">
    <a href="{{ url_for('about') }}">About Sepsis</a>
    <a href="{{ url_for('predict') }}">Predict</a>
</div>
</header>
<h1><b>WELCOME TO SEPSIS SURVIVAL MINIMAL CLINICAL
RECORDS!</b></h1>
<p>Sepsis is a serious condition in which the body responds improperly to an infection. The
infection-fighting processes turn on the body, causing the organs to work poorly.</p>
<p>Sepsis may progress to septic shock. This is a dramatic drop in blood pressure that can
damage the lungs, kidneys, liver, and other organs. When the damage is severe, it can
lead to death.</p>
<p>Early treatment of sepsis improves chances for survival.</p>
```

</body>

</html>

about.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>About</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background: linear-gradient(135deg, #f5f7fa, #c3cfe2);

color: #333;

}

header {

background: linear-gradient(135deg, #ffafbd, #ffc3a0);

padding: 20px 0;

}

header div {

display: flex;

justify-content: flex-end;

align-items: center;

padding-right: 20px;

}

header a {

color: white;

text-decoration: none;

font-size: 20px;

margin-left: 20px;

transition: color 0.3s ease;

```
    }
    header a:hover {
        color: #ffdfba;
    }
    p {
        font-size: 18px;
        line-height: 1.6;
        max-width: 800px;
        margin: 30px auto;
        padding: 0 20px;
        text-align: justify;
    }
    .content {
        text-align: center;
    }
    .content img {
        width: 80%;
        height: auto;
        margin: 20px 0;
        border-radius: 10px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }
</style>
</head>
<body>
    <header>
        <div>
            <a href="{{ url_for('home') }}">Home</a>
            <a href="{{ url_for('predict') }}">Predict</a>
        </div>
    </header>
    <p>
```

Sepsis is a serious condition in which the body responds improperly to an infection. The infection-fighting processes turn on the body, causing the organs to work poorly.

</p>

<p>

Sepsis may progress to septic shock. This is a dramatic drop in blood pressure that can damage the lungs, kidneys, liver, and other organs. When the damage is severe, it can lead to death.

</p>

<p>

Early treatment of sepsis improves chances for survival.

</p>

<div class="content">

</div>

</body>

</html>

predict.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Predict</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background: linear-gradient(135deg, #f5f7fa, #c3cfe2);

color: #333;

}

header {

background: linear-gradient(135deg, #ffafbd, #ffc3a0);

padding: 20px 0;

```
}  
header div {  
  display: flex;  
  justify-content: flex-end;  
  align-items: center;  
  padding-right: 20px;  
}  
header a {  
  color: white;  
  text-decoration: none;  
  font-size: 20px;  
  margin-left: 20px;  
  transition: color 0.3s ease;  
}  
header a:hover {  
  color: #ffdfba;  
}  
h1 {  
  text-align: center;  
  color: #6d5dfc;  
  font-size: 36px;  
  margin-top: 30px;  
}  
form {  
  text-align: center;  
  margin-top: 20px;  
}  
label {  
  font-size: 20px;  
  margin-bottom: 10px;  
  display: block;  
}  
input[type="number"] {  
  width: 300px;
```



```
        height: 30px;
        margin-bottom: 20px;
        padding: 5px;
        border-radius: 12px;
        border: 1px solid #ccc;
        font-size: 18px;
    }
    input[type="submit"] {
        background-color: rgb(83, 159, 240);
        color: white;
        width: 200px;
        height: 40px;
        font-size: 20px;
        border: none;
        border-radius: 12px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }
    input[type="submit"]:hover {
        background-color: rgb(60, 120, 180);
    }
    .content {
        text-align: center;
        margin-top: 20px;
    }
    .content img {
        width: 100%;
        max-width: 1400px;
        height: auto;
        margin-bottom: 30px;
        border-radius: 10px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }
</style>
```

```

</head>
<body>
  <header>
    <div>
      <a href="{{ url_for('home') }}">Home</a>
    </div>
  </header>
  <div class="content">
    
  </div>
  <h1><i>Enter the following details to predict!!!...</i></h1>
  <div style="text-align: center;">
    <form action="{{ url_for('output') }}" method="POST">
      <label for="age"><b>Age:</b></label>
      <input id="age" name="age" type="number" required>
      <br>
      <label for="gender"><b>Gender:</b></label>
      <input id="gender" name="gender" type="number" placeholder="0-male // 1-female"
        required>
      <br>
      <label for="episode_number"><b>Episode Number:</b></label>
      <input id="episode_number" name="episode_number" type="number"
        placeholder="Enter number from 1-4" required>
      <br>
      <input type="submit" value="Submit">
    </form>
  </div>
</body>
</html>

```

output.html

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Prediction Result</title>
</head>
<body style="background-color: #68b6ba">
  <header>
    <div style="background-color: black; padding: 20px;">
      <a href="{{ url_for('home') }}" style="color: white; text-decoration: none; float:
      right;">Home</a>
      <a href="{{ url_for('predict') }}" style="color: white; text-decoration: none; float: right;
      margin-right: 20px;">Predict Again</a>
    </div>
  </header>
  <h1 style="text-align: center;"><font color="blue" size="55px"><i>Prediction
  Result</i></font></h1>
  <div style="text-align: center;">
    <h2>The Patient is likely : <span style="color: red;"></span></h2>
    {% if prediction == 'Dead' %}
      <h1>Not Survived</h1>
    {% elif prediction == 'Alive' %}
      <h1>Survived</h1>
    {% endif %}
  </div>
</body>
</html>
```

```
app.py
import pickle
from flask import Flask, render_template, request
import pandas as pd

app = Flask(__name__)
with open('sepsis_survival.pkl', 'rb') as file:
```

```
model = pickle.load(file)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')

@app.route('/output', methods=['POST'])
def output():
    if request.method == 'POST':
        try:
            age_years = int(request.form['age'])
            sex_0male_1female = int(request.form['gender'])
            episode_number = float(request.form['episode_number'])

            features = [[age_years, sex_0male_1female, episode_number]]
            print(f"Input features: {features}")

            # Create a DataFrame with appropriate feature names
            feature_names = ['age_years', 'sex_0male_1female', 'episode_number']
            features_df = pd.DataFrame(features, columns=feature_names)

            # Print DataFrame to debug input
            print(f"Features DataFrame:\n{features_df}")

            # Predict the outcome
            prediction = model.predict(features_df)[0]
```

```
print(f"Raw prediction: {prediction}")

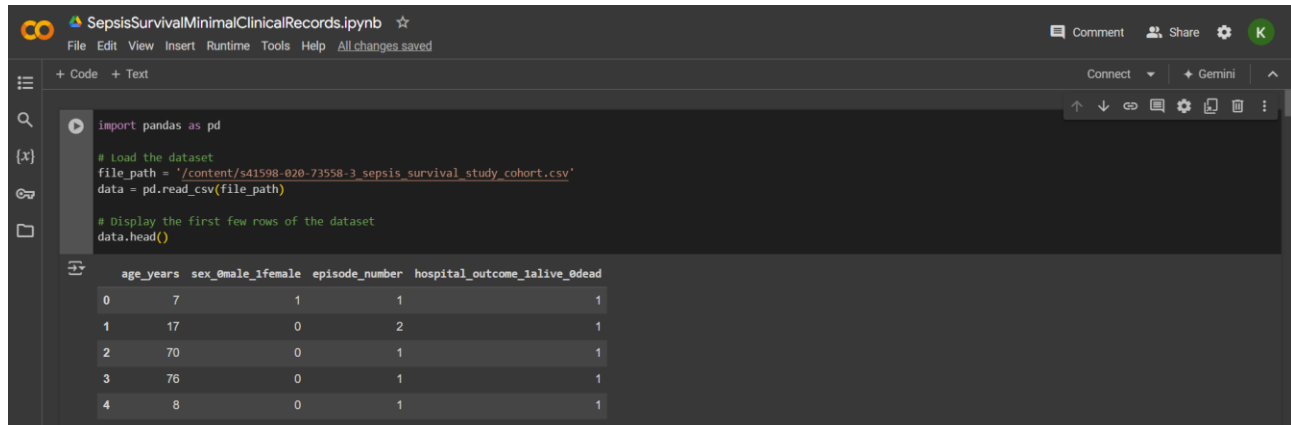
# Handle prediction output
if prediction == 0:
    result = "Dead"
elif prediction == 1:
    result = "Alive"
else:
    result = "Unknown"

print(f"Result: {result}")
return render_template("output.html", prediction=result)

except Exception as e:
    print(f"Error during prediction: {e}")
    return render_template("output.html", prediction="Error during prediction")

if __name__ == '__main__':
    app.run(debug=True)
```

MODEL BUILDING



```
import pandas as pd

# Load the dataset
file_path = '/content/s41598-020-73558-3 sepsis_survival_study_cohort.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
data.head()
```

	age_years	sex_0male_1female	episode_number	hospital_outcome_1alive_0dead
0	7	1	1	1
1	17	0	2	1
2	70	0	1	1
3	76	0	1	1
4	8	0	1	1

```

SepsisSurvivalMinimalClinicalRecords.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
Connect + Gemini

[ ] # Descriptive statistics
data.describe(include='all')

age_years  sex_male_ifemale  episode_number  hospital_outcome_1alive_0dead
count  19051.000000      19051.000000      19051.000000      19051.000000
mean    72.503228          0.448585          1.396252          0.810719
std     18.614470          0.497363          0.783749          0.391742
min       0.000000          0.000000          1.000000          0.000000
25%      65.000000          0.000000          1.000000          1.000000
50%      77.000000          0.000000          1.000000          1.000000
75%      85.000000          1.000000          2.000000          1.000000
max     100.000000          1.000000          5.000000          1.000000

[ ] # Check for missing values
missing_values = data.isnull().sum()
missing_values

age_years          0
sex_male_ifemale  0
episode_number    0
hospital_outcome_1alive_0dead  0
dtype: int64
  
```

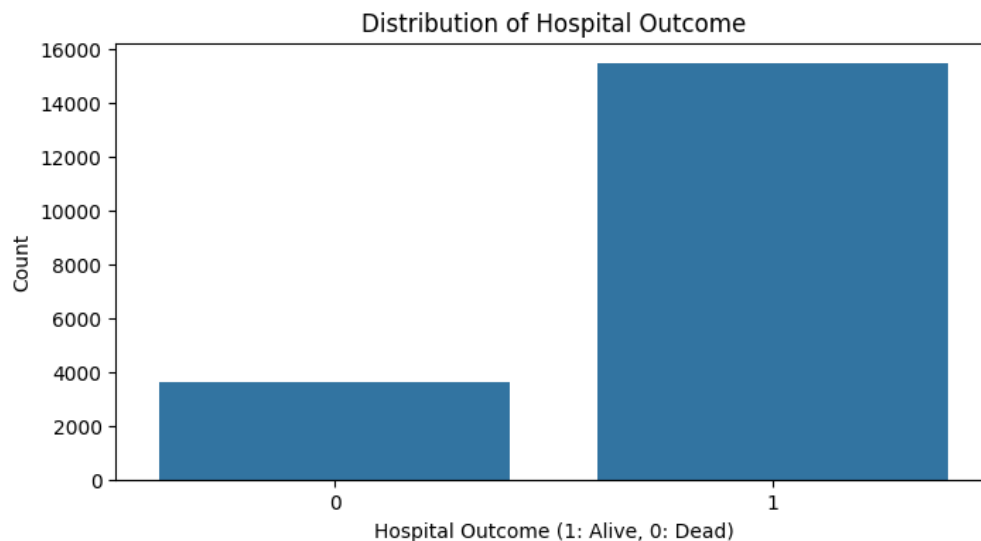
```

import matplotlib.pyplot as plt
import seaborn as sns

# Set up the visualizations
plt.figure(figsize=(8, 4))

# Visualize the distribution of the target variable
sns.countplot(x='hospital_outcome_1alive_0dead', data=data)
plt.title("Distribution of Hospital Outcome")
plt.xlabel("Hospital Outcome (1: Alive, 0: Dead)")
plt.ylabel('Count')

plt.show()
  
```



```

q1=data['age_years'].quantile(0.25)
q3=data['age_years'].quantile(0.75)
iqr=q3-q1
q1,q3,iqr
  
```

```
(65.0, 85.0, 20.0)
```

+ Code + Text

```

[ ] upper_bound=q3+1.5*iqr
    lower_bound=q1-1.5*iqr
    upper_bound,lower_bound
  
```

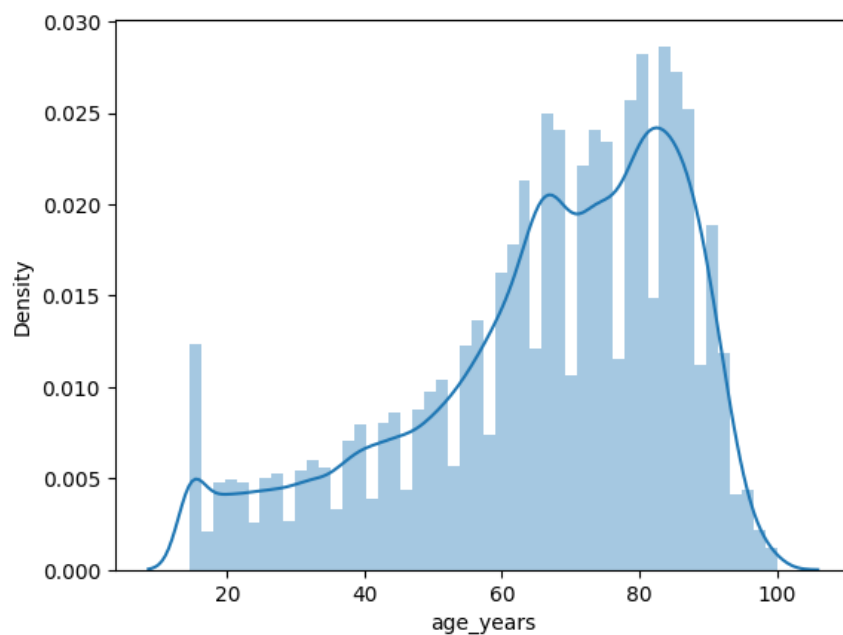
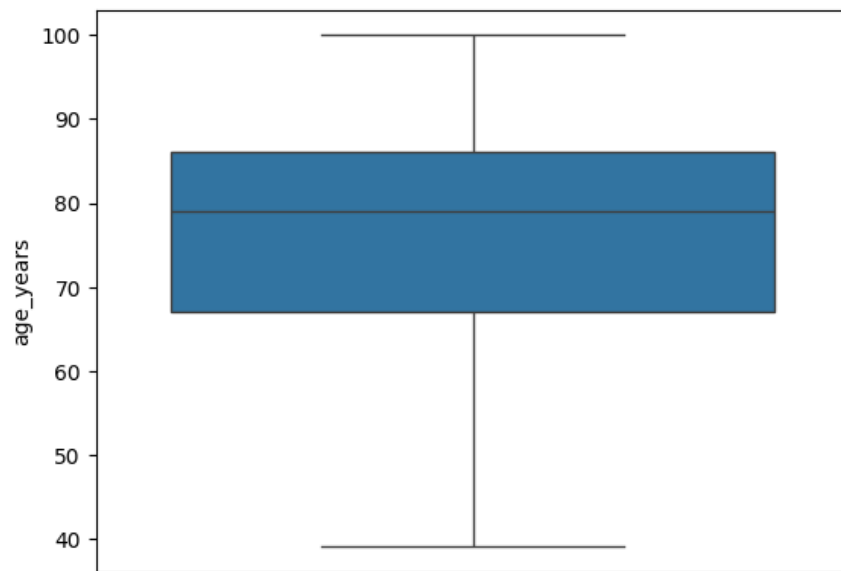
```
(115.0, 35.0)
```

```

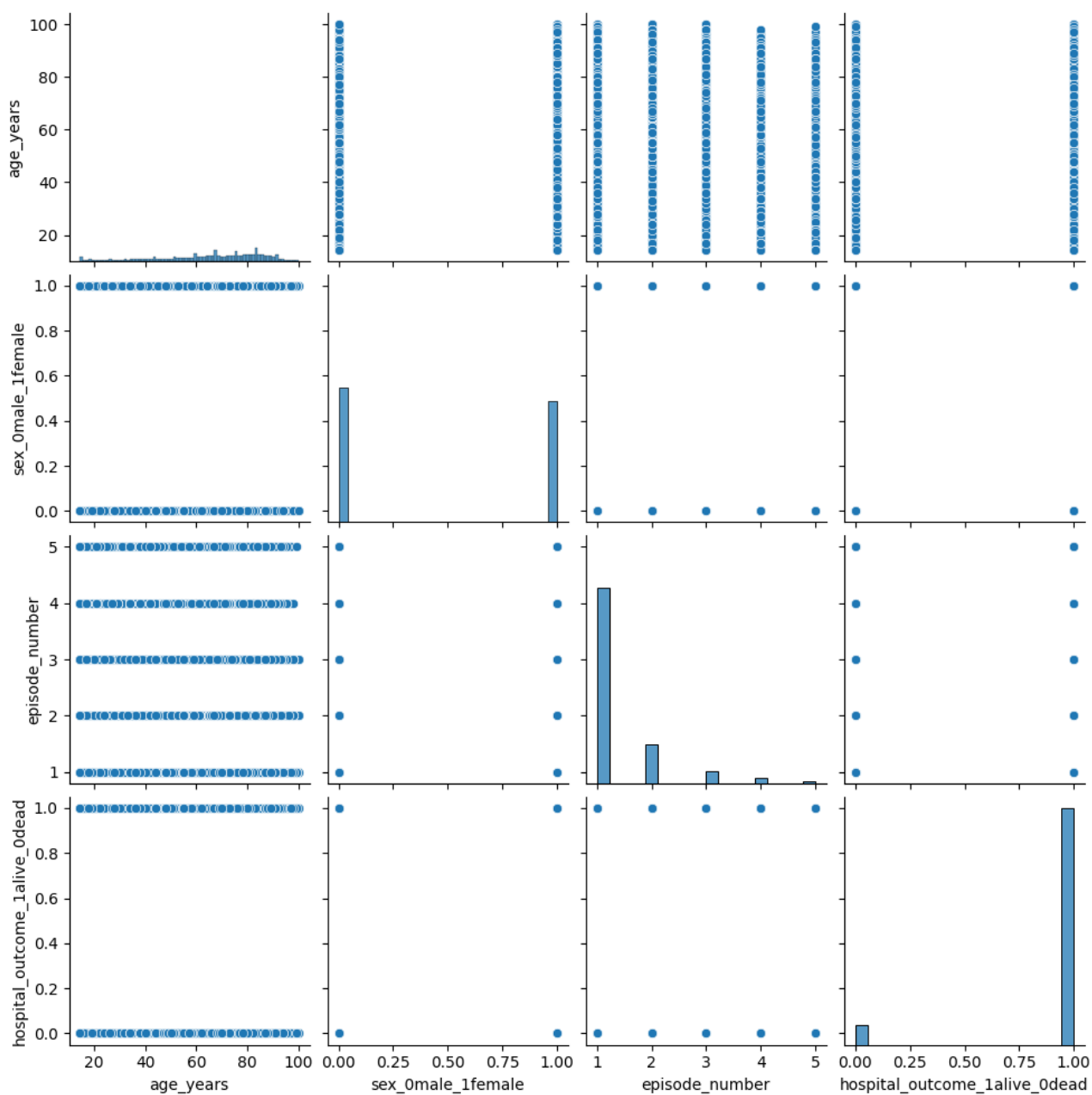
[ ] data1=data[(data['age_years']>=lower_bound) & (data['age_years']<=upper_bound)]
    print("before removing outliers ",len(data))
    print("after removing outliers ",len(data1))
    print("Total Outliers ",len(data)-len(data1))
  
```

```

before removing outliers 19051
after removing outliers 18065
Total Outliers 986
  
```

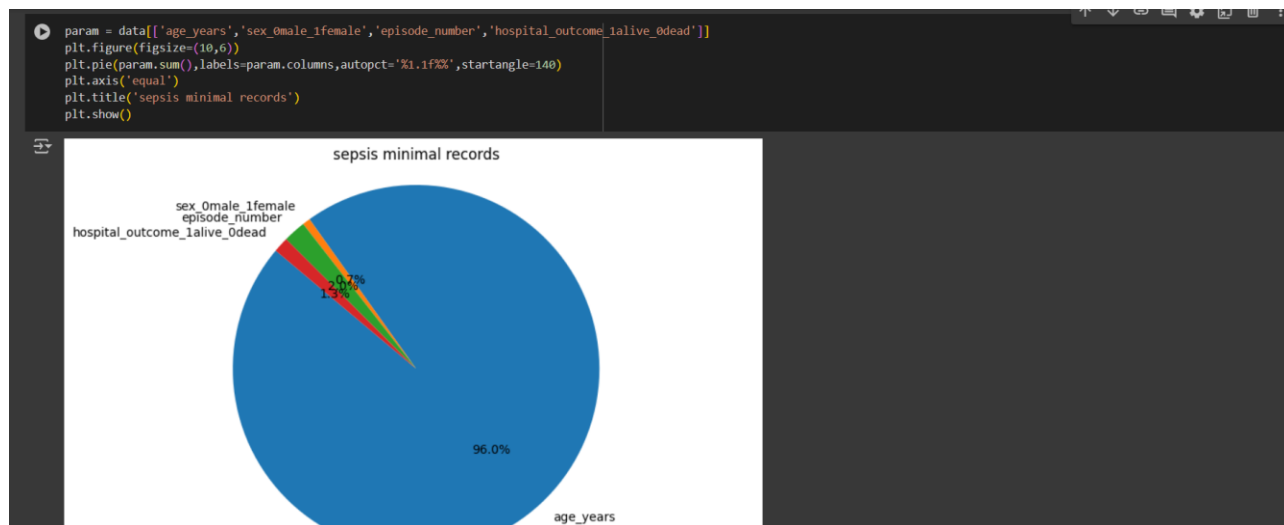


PAIRPLOT



HEATMAP





```
[28] #splitting the data into training data and testing data
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)

[29] #checking the accuracy of the model using LogisticRegression
from sklearn.linear_model import LogisticRegression
log = LogisticRegression()

[30] log.fit(x_train,y_train)

+ LogisticRegression
LogisticRegression()

[31] y_train_pred = log.predict(x_train)
y_test_pred = log.predict(x_test)

[32] from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

[33] train_accuracy = accuracy_score(y_train,y_train_pred)
train_precision = precision_score(y_train,y_train_pred,average='weighted')
train_recall = recall_score(y_train,y_train_pred,average='weighted')
train_f1score = f1_score(y_train,y_train_pred,average='weighted')

[34] test_accuracy = accuracy_score(y_test,y_test_pred)
test_precision = precision_score(y_test,y_test_pred,average='weighted')
test_recall = recall_score(y_test,y_test_pred,average='weighted')
test_f1score = f1_score(y_test,y_test_pred,average='weighted')

print("log train accuracy:",train_accuracy)
print("log test accuracy:",test_accuracy)
print("log train precision:",train_precision)
print("log test precision:",test_precision)
print("log train recall:",train_recall)
print("log test recall:",test_recall)
print("log train f1score:",train_f1score)
print("log test f1score:",test_f1score)

log train accuracy: 0.9226058436874839
log test accuracy: 0.9241736258851173
log train precision: 0.8512015428062939
log test precision: 0.8540908907816449
log train recall: 0.9226058436874839
```

```
[37] knn.fit(x_train,y_train)

[38] y_train_pred = knn.predict(x_train)
     y_test_pred = knn.predict(x_test)

train_accuracy = accuracy_score(y_train,y_train_pred)
train_precision = precision_score(y_train,y_train_pred,average='weighted')
train_recall = recall_score(y_train,y_train_pred,average='weighted')
train_fscore = f1_score(y_train,y_train_pred,average='weighted')

[40] test_accuracy = accuracy_score(y_test,y_test_pred)
     test_precision = precision_score(y_test,y_test_pred,average='weighted')
     test_recall = recall_score(y_test,y_test_pred,average='weighted')
     test_fscore = f1_score(y_test,y_test_pred,average='weighted')

[41] print("knn train accuracy:",train_accuracy)
     print("knn test accuracy:",test_accuracy)
     print("knn train precision:",train_precision)
     print("knn test precision:",test_precision)
     print("knn train recall:",train_recall)
     print("knn test recall:",test_recall)
     print("knn train fscore:",train_fscore)
     print("knn test fscore:",test_fscore)
```

```
[44] y_train_pred = dt.predict(x_train)
     y_test_pred = dt.predict(x_test)

[45] train_accuracy = accuracy_score(y_train,y_train_pred)
     train_precision = precision_score(y_train,y_train_pred,average='weighted')
     train_recall = recall_score(y_train,y_train_pred,average='weighted')
     train_fscore = f1_score(y_train,y_train_pred,average='weighted')

[46] test_accuracy = accuracy_score(y_test,y_test_pred)
     test_precision = precision_score(y_test,y_test_pred,average='weighted')
     test_recall = recall_score(y_test,y_test_pred,average='weighted')
     test_fscore = f1_score(y_test,y_test_pred,average='weighted')

[47] print("dt train accuracy:",train_accuracy)
     print("dt test accuracy:",test_accuracy)
     print("dt train precision:",train_precision)
     print("dt test precision:",test_precision)
     print("dt train recall:",train_recall)
     print("dt test recall:",test_recall)
     print("dt train fscore:",train_fscore)
     print("dt test fscore:",test_fscore)

dt train accuracy: 0.9226058436874839
dt test accuracy: 0.9241736258851173
dt train precision: 0.8512015428062939
dt test precision: 0.8540968907816449
dt train recall: 0.9226058436874839
dt test recall: 0.9241736258851173
dt train fscore: 0.8854665095303385
dt test fscore: 0.887754492933094

[54] from imblearn.over_sampling import SMOTE
     from imblearn.under_sampling import RandomUnderSampler
     from imblearn.pipeline import Pipeline
     from sklearn.ensemble import RandomForestClassifier

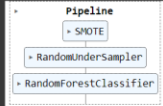
[55] rus = RandomUnderSampler()

[56] smote = SMOTE(sampling_strategy='auto')

[57] rf_classifier = RandomForestClassifier()

pipeline = Pipeline([
    ('smote', smote),
    ('rus',rus),
    ('rf',rf_classifier)
])

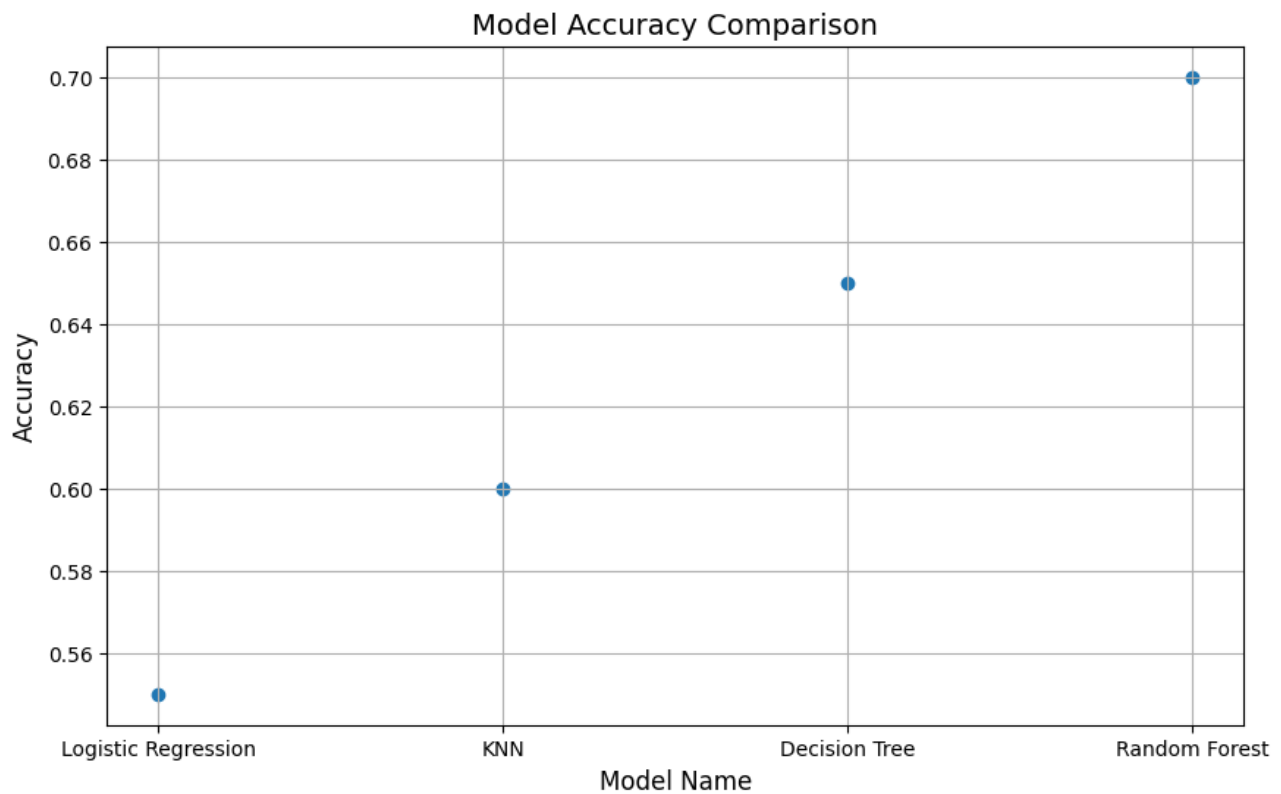
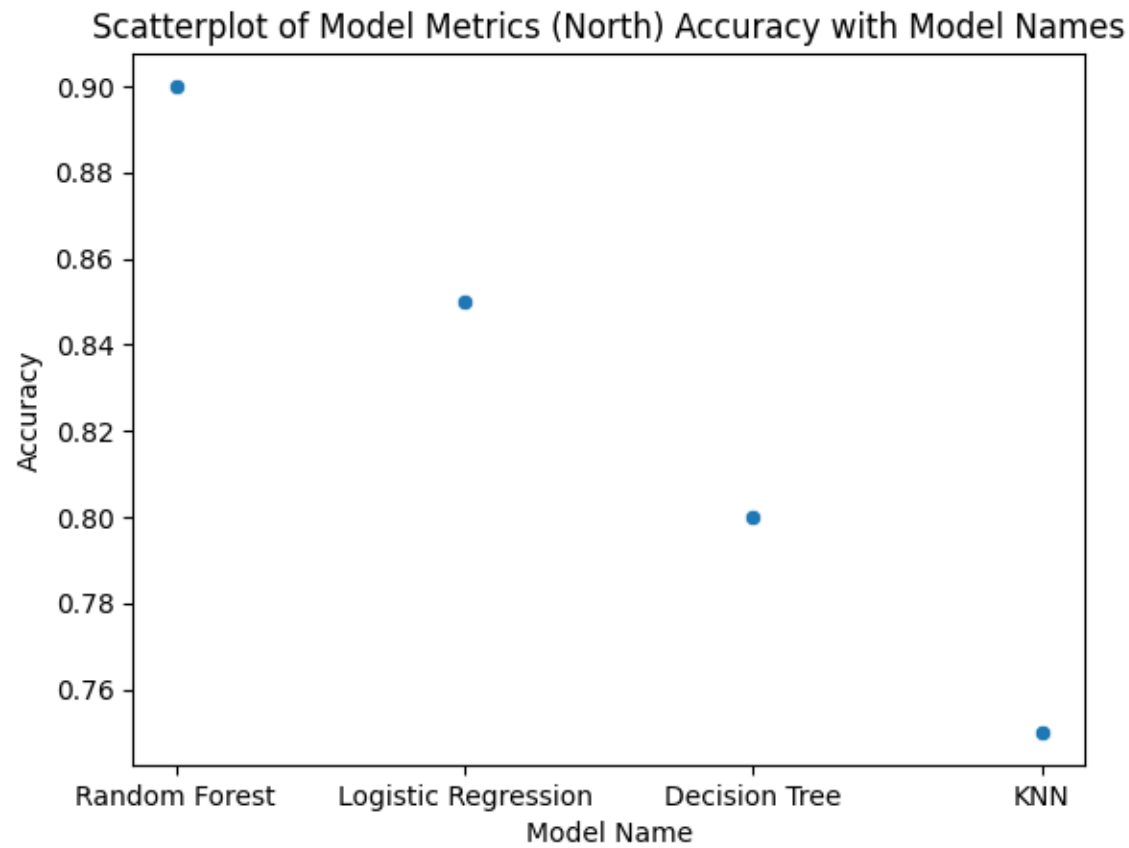
[59] pipeline.fit(x_train,y_train)
```



```

graph TD
    Pipeline --> SMOTE
    SMOTE --> RandomUnderSampler
    RandomUnderSampler --> RandomForestClassifier
  
```

DESCRIPTIVE ANALYSIS



10.2. GitHub & Project Demo Link