A Report On

"Predicting Mortality in ICU using ML"

Submitted to the

Department of Computer Applications

In partial fulfilment of the Course

Master of Computer Applications

Under the guidance of

Mr. Praveenkumar K S

BY

Akhil Shibu

(Reg no: SGI20MCA-2004)

DEPARTMENT OF COMPUTER APPLICATIONS

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520

2020-2022

SNGIST
GROUP OF INSTITUTIONS

Experience Learning

13.07.2022

Project Completion Certificate

This is to certify that **Mr. Akhil Shibu (Reg No: SGI20MCA-2004) S4 M.C.A** degree student of **SNGIST Group of Institutions, N. PARAVUR** has successfully completed his main project on the topic **"Predicting mortality in ICU using Machine Learning"** from our college during March - June 2022.

HOD (MCA)

Principal

Sree Narayana Guru Institute of Science and Technology (SNGIST)
Manjaly P.O., North Paravur, Ernakulam Dist., Kerala-683520, India.
Ph: 0484-2887000 · Fax: +91 484 2440299 · E-mail: sngist@sngist.org

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520

BONAFIDE CERTIFICATE

Certified that the Project Work entitled

"Predicting Mortality in ICU Using ML"

is a bonafide work done by

Akhil Shibu

In partial fulfilment of the requirement for the Award of

MASTER OF COMPUTER APPLICATIONS

Degree From

APJ Abdul Kalam Technological University, Thiruvananthapuram

(2020- 2022)

Head of Department　　　　　　　　　　　　　　　　Project Guide

Submitted for the Viva-Voce Examination held on…………………………

External Examiner1
(Name & Signature)

External Examiner2
(Name & Signature)

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520

SNGIST

CERTIFICATE

This is to certify that the project entitled **"Predicting Mortality in ICU Using ML"** has been successfully carried out by **Akhil Shibu** **(**Reg no: SGI20MCA-2004) in partial fulfilment of the Course **Master of Computer Applications.**

Dr. Kavitha C R

Date: **HEAD OF DEPARTMENT**

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520

CERTIFICATE

This is to certify that the project entitled "**Predicting Mortality in ICU Using ML**" has been successfully carried out by **Akhil Shibu** (Reg no: SGI20MCA-2004) in partial fulfilment of the course **Master of Computer Applications** under my guidance.

Mr Praveenkumar KS

Date: **INTERNAL GUIDE**

SNGIST GROUP OF INSTITUTIONS

North Paravur- 683520

DECLARATION

I, **Akhil Shibu**, hereby declare that the project work entitled "**Predicting Mortality in ICU Using ML**" is an authenticated work carried out by me under the guidance of **Mr. Praveenkumar K S** for the partial fulfilment of the course **MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SNGIST GROUP OF INSTITUTIONS, North Paravur,** affiliated to **APJ ABDUL KALAM UNIVERSITY, THIRUVANANTHAPURAM.** I understand that detection of any such copying is liable to be punished in any way the college deems fit.

Date:

Place: Manjaly   Akhil Shibu

ACKNOWLEDGEMENT

I express my sense of gratitude and sincere thanks to Almighty for his abiding presence and the abounding grace of unseen hand yet tangible guidance all through the formation of this project.

I am thankful to **Prof. Dr. SAGINI THOMAS MATHAI, Principal**, SNGIST Group of Institutions, North Paravur, for her kind support in all aspect during my study.

I sincerely thank **Dr. KAVITHA C.R, HOD,** Department of Computer Applications, SNGIST Group of Institutions, North Paravur, for her encouragement and inspiration of this project work.

I especially thank my Internal guide **Mr. Praveenkumar K S, Assistant Professor**, Department of Computer Applications, SNGIST Group of Institutions, for her guidance and constant supervision as well as for providing necessary information regarding the project.

I want to thank the Department of Computer Applications for giving me the permission to prepare the project on the topic "**Predicting Mortality in ICU Using ML**".

Akhil Shibu

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

Providing an accurate prediction of mortality risk for patients in the health care system as early as possible could help improve care quality and reduce costs. The recent adoption of Electronic Health Records (EHRs) has created an opportunity to improve prediction accuracy by obtaining more detailed data and applying more advanced algorithms. In this work, we applied Machine Learning algorithm to estimate the mortality risk of patients admitted to a single institution's Intensive Care Units (ICUs). Unlike the prior studies that utilize a rich set of features usually available at discharge. we used commonly available data in EHRs at the time of admission.

In this project, we are experiencing Model Deployment (web Page). Here uses different algorithms like Decision tree, Random forest, Extra trees, KNN, Adaboost, XG boost to predict a better model .To build the proposed model, Random Forest is used with accuracy 98 % .Random Forest is a supervised learning classifier that contains a number of decision trees on various subsets of the dataset. The greater number of trees in the forest leads to high accuracy and prevent the problem of overfitting. It is based on the concept of ensemble learning.

Scrum framework in agile methodology was adopted for the development of this project. Agile is a group of software development methodologies or frameworks based on iterative development, with the support of a functional team. This methodology focuses on the frequent delivery of working software.

1.INTRODUCTION

Patients in intensive care unit (ICU) are continuously monitored for medical abnormalities, disease prognosis and potential complications due to their critical health status. They require more specialized care and are time-sensitive to medical responses by doctors and healthcare professionals. There exist challenges in providing ICU patients with timely intervention as they require more individualized care and attention. Healthcare professionals often use case-based reasoning, relying on knowledge accumulated from similar past medical cases to diagnose and treat their current patient . As such, the sharing of knowledge and understanding of key risk factors are more likely to help increase the speed and effectiveness of diagnosis for a new patient.

Hospitals currently utilize severity score systems such as SOFA, APACHE to predict ICU mortality. However, these criteria/rubric-based score systems which bin and compare different strata of patient populations can have limited predictive accuracy at the individual level.Advances in machine learning methodologies have reported classification results with accuracy of above 80%. This study aims to put forth a white-box feature selection and classifier model which can clearly pinpoint key health indicators that can better predict individual patient's in-hospital mortality.

Further, the demand for ICU hospital beds worldwide has been increasing, with ICU costs having risen to nearly 22% of hospital costs and 5% of the total healthcare cost. Despite the increasing demand for ICU hospital beds worldwide, the availability of hospital beds (less than 10% of hospital beds), medical staff and equipment remain limited resources [6,7]. Decisions made by the management of ICU wards in hospitals can directly impact a patient's survival rate. With limited manpower, equipment, supplies, and bed/ward availability in hospitals, resource allocation is often an issue that hospitals face. The objectives of this study are to identify risk factors associated with patients who are non-survivors, as well as survivors in ICU using Machine Learning. By understanding these factors, the hospital can better allocate available resources and attention to where care is needed most. This can also be aimed at helping doctors and healthcare professionals plan effective timely intervention, tailoring specific level of care for an individual ICU patient.

The technique that can be used is the grading system using machine learning. In this study I use diabetes mellitus dataset form MIT's GOSSIS (Global Open -Source Severity of Illness Score) initiative. It contain 85 features from seven feature categories with 91713 data values. It has only a single target" hospital death". The target values are classified into 0's and 1's. 0's indicate non-survived and 1's indicate patients is survived.

2.1. Existing System

Hospitals currently utilize severity score systems such as SOFA, APACHE to predict ICU mortality. However, these criteria/rubric-based score systems which bin and compare different strata of patient populations can have limited predictive accuracy at the individual level.Advances in machine learning methodologies have reported classification results with accuracy of above 80%

2.2. Definition of Problem

Providing an accurate prediction of mortality risk for patients in the health care system as early as possible could help improve care quality and reduce costs. The recent adoption of Electronic Health Records (EHRs) has created an opportunity to improve prediction accuracy by obtaining more detailed data and applying more advanced algorithms. In this work, we applied Machine Learning algorithm to estimate the mortality risk of patients admitted to a single institution's Intensive Care Units (ICUs). Unlike the prior studies that utilize a rich set of features usually available at discharge. we used commonly available data in EHRs at the time of admission.

2.3. Proposed System

The demand for ICU hospital beds worldwide has been increasing, with ICU costs having risen to nearly 22% of hospital costs and 5% of the total healthcare cost. Despite the increasing demand for ICU hospital beds worldwide, the availability of hospital beds (less than 10% of hospital beds), medical staff and equipment remain limited resources [6,7]. Decisions made by the management of ICU wards in hospitals can directly impact a patient's survival rate. With limited manpower, equipment, supplies, and bed/ward availability in hospitals, resource allocation is often an issue that hospitals face. The objectives of this study are to identify risk factors associated with patients who are non-survivors, as well as survivors in ICU using Machine Learning. By understanding these factors, the hospital can better allocate available resources and attention to where care is needed most. This can also be aimed at helping doctors and healthcare professionals plan effective timely intervention, tailoring specific level of care for an individual ICU patient.

2.4. Objective of the project

The objectives of this study are to identify risk factors associated with patients who are non-survivors, as well as survivors in ICU .By understanding these factors, the hospital can better allocate available resources and attention to where care is needed most.This can also be aimed at helping doctors and healthcare professionals plan effective timely intervention, tailoring specific level of care for an individual ICU patient.

2.5. Scope of the project

Our system can be used both by doctors and medical assistants to predict whether the patient will survive or not by monitoring in ICU for 24hours and collecting the EHR records and using that records into our model The UI is designed to be as simple as possible so that the user can easily use the system.The workflow of the software is easy to be understood by the users so that it is easily able to operate the software and have a good user interface. Also satisfying all the user requirements

2.6. Hardware Software requirements

Hardware

- Processor - Intel core i3 - 3220 (3.3 Ghz) or above
- Ram - 4 GB or aboveStorage - 512 GB or above
- Other - Keyboard, Mouse

Software

- Windows 10 or above
- IDE -VS CodeTool Kit – Anaconda
- Jupyter notebook
- Front end – HTML, CSS, JS, Python, Django

3.METHODOLOGY

3.1. Scrum

Scrum is one of the agile frameworks to manage software development. Scrum is an iterative incremental framework for managing complex software development processes. In scrum the project is divided into multiple short time boxes of 2 to 4 weeks duration called a sprint. The main focus of this scrum framework is to deliver the working increment as fast as possible on a team collaboration. The result of each sprint in the scrum will be a fully tested and approved piece of software. In each of the sprint meetings the product owner decides which user stories should be selected for the upcoming sprint. the fixing of the outstanding bugs will also be included. After all work is done the sprint ends with a sprint review meeting in which the current status of the project will be presented to the product owner.

3.2. Scrum Roles

Product Owner

Mr Shameer K S, Senior Faculty, was the product owner of this project, and acted as spokesperson for the customer and he defined features of the product based on each backlog item or each specific request of the customer. He is responsible for the prioritisation of features according to the market value, and he decides the release date for the product, and he is responsible for the profitability of the product. The product owner will also make changes in the features for the improvement of the product and the priority of the features are also decided by the product owner after each sprint.

Scrum Master

Mr Praveenkumar K S, Associate Professor of MCA was the Scrum master for this project. The Scrum master is responsible for making sure whether the scrum team properly follows the scrum practices, and for solving the issues and barriers in the progress of the team and project. As a scrum master, she should protect the team from external interferences, and ensure that the Scrum process is going well.

Scrum Team

The scrum team consists of a group of people developing the software product. In this project, the scrum team consists of Mr Shameer K S, the product owner, Mr Praveenkumar K S, the Scrum master as well as the project supervisor and Akhil Shibu, Developer. There is no personal responsibility in Scrum, the whole team fails or succeeds as a single entity

.3.3. Sprint Planning Meeting

Most of the time the sprint planning meetings took place as planned, but sometimes the product owner was unavailable for the meeting. In such cases, the meeting simply needed to be rescheduled one or two days later. These extra days would come in advantage for cleaning up what the team has developed since the last scrum and in creative implementations.

3.4. Daily Scrum Meeting

Our daily Scrums took place at 10.00am. The team members will arrive as early as 9.00am and work until then, but if they did arrive before the meeting started it did not matter.

3.5 Sprint Review Meeting

Our review meetings were always held on Fridays. The product owner will visit the project team with any other interested parties, and the team will demonstrate the developed features on a live system and answer the questions that arose during the demo. Usually, we would spend one or two days before the demo checking if everything was working and run tests on the developed product.

3.6 Product Backlog

User story	**Story point**	**Priority**
As a user I want to enter parameters and view prediction result So that to know water is potable	8	1
As a user I want to get a portable document of my result	8	2

As a user I want to verify a given result is genuine or not	2	3
As a user I want to Sign in with google	3	4
As a user I want to register using email and password	2	5
As a user I want to login using email and password	1	6
As an admin I want to login So that I can access the system	2	7
As an admin I want to view users	3	8
As an admin I want to view message	3	9
As an admin I want to view records	3	10
As an admin I want to remove inactive users	3	11
As an admin I want to remove fake message.	5	12
As an admin I want to remove old records of prediction	5	13

3.6.1 User Stories

User type	**Epic**	**User story**
Admin	Login	As an admin I want to login So that I can access the system
	My account	As an admin I want to view registered users, messages and predictionsAs an admin I want to delete inactive users, fake messages and old predictions
User	Registration	As a user I want to register with email, username and password
	Login	

	My account	As a user I want to login with username and passwordAs a user I want to sign in using google
		As a user I want to predict mortaslity in icu.As a user I want to print prediction resultAs a user I want to verify a give result is genuine or not

4. MILESTONES

4.1. Sprint 1

The first meeting was held on 31st January 2022. The team consisted of Mr. Praveenkumar K S, the product owner and Mr.Akhil Shibu, the developer. The requirement collection was mainly done in this sprint which took nearly 3 weeks. And the sprint ended on 21st February 2022.

4.2. Sprint 2

Second sprint started on 28th February 2022 in which the database design was started and the user interface of the system was also designed and the appropriate deep learning model was chosen.

4.3. Sprint 3

Third sprint started on 7th March 2022 in which one of the main modules, that is the module for mortality prediction and user module’s development was started. It also includes the development of sub modules.

4.4 Sprint 4

Fourth sprint started on 21st march 2022 in which the doctor and admin module was developed.

4.5 Sprint 5

Fifth sprint started on 18th April 2022 in which testing and validation are done for each module

5. MODULE DESCRIPTION

5.1. Login

There are two types of users to the system, the admin and user Each of the users have their own username and password to access the system. This module is responsible for the dealing of login of all types of users.

5.2. Admin

The module which is responsible for the activities of admin.

5.2.1. User log

Module responsible for viewing and removing of users

5.2.2. Record log

Module responsible for viewing and removing of prediction records

5.3. Mortality Prediction

Module which is responsible for the mortality prediction. In the module a model which uses Random Forest is used for the classification .

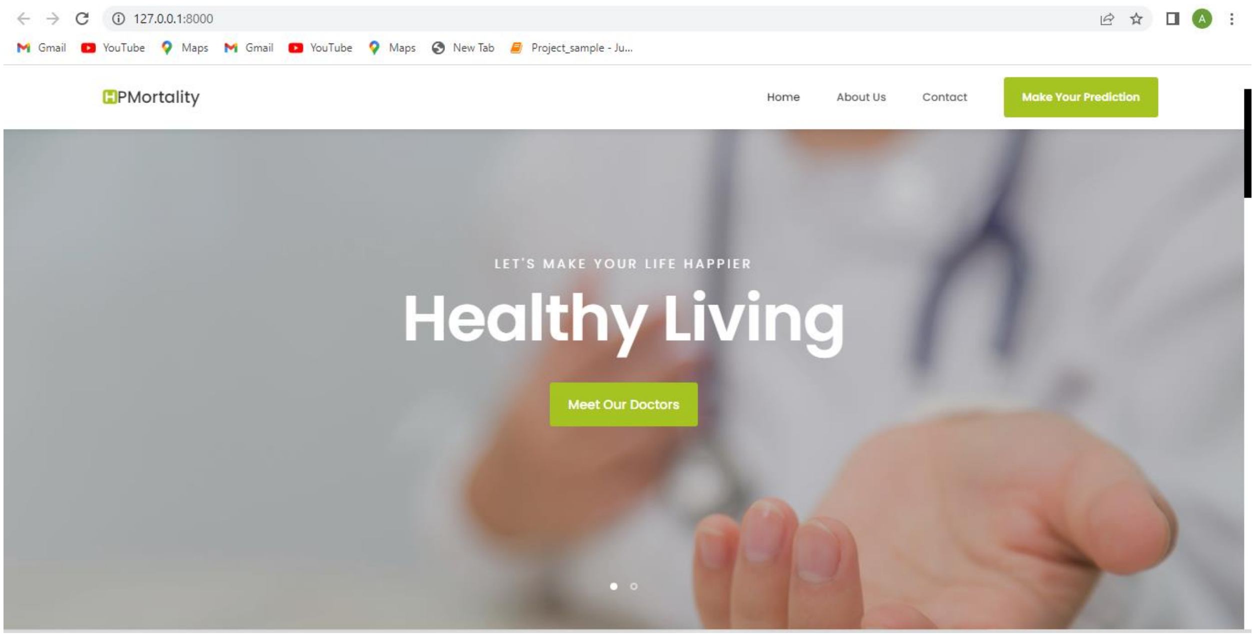
5.4 User

This module is responsible for user to input the parameters and predict there mortality prediction, print and verify.

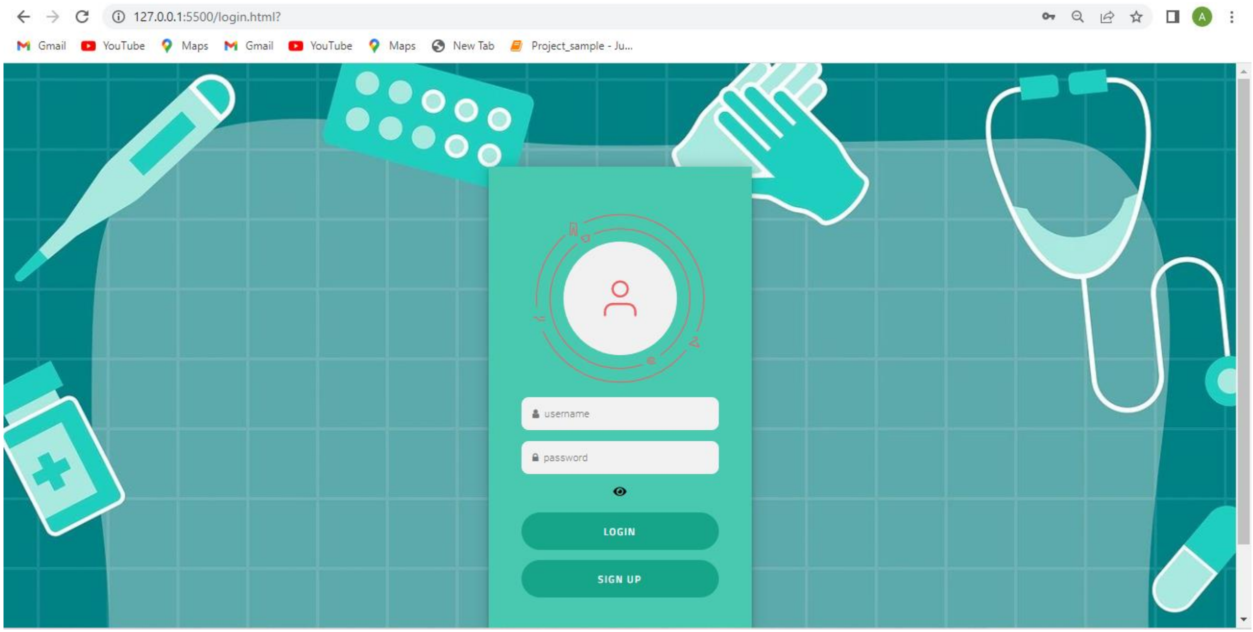
SYSTEM DESIGN

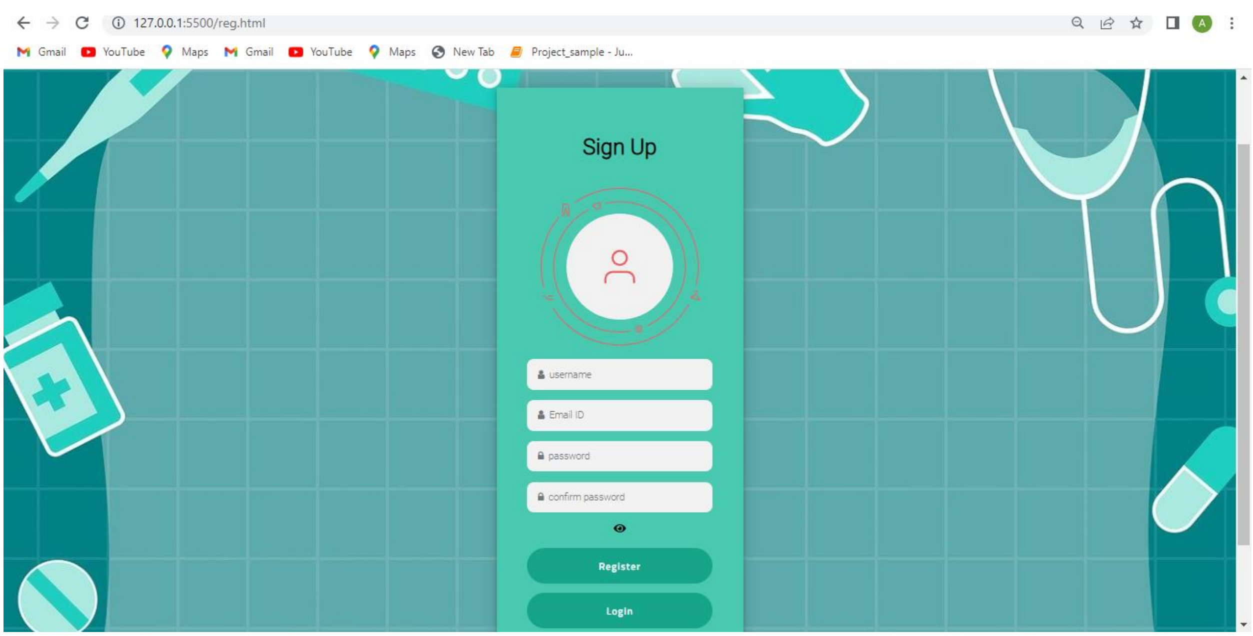
6.1. UI Design

6.1.1 Index Page

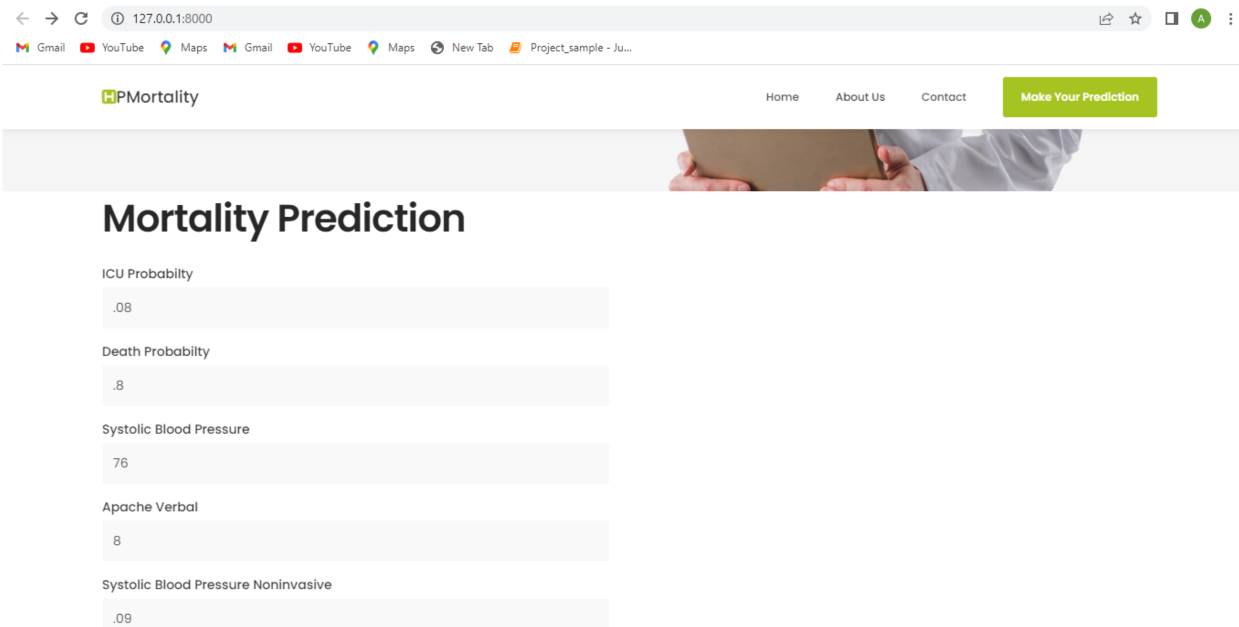


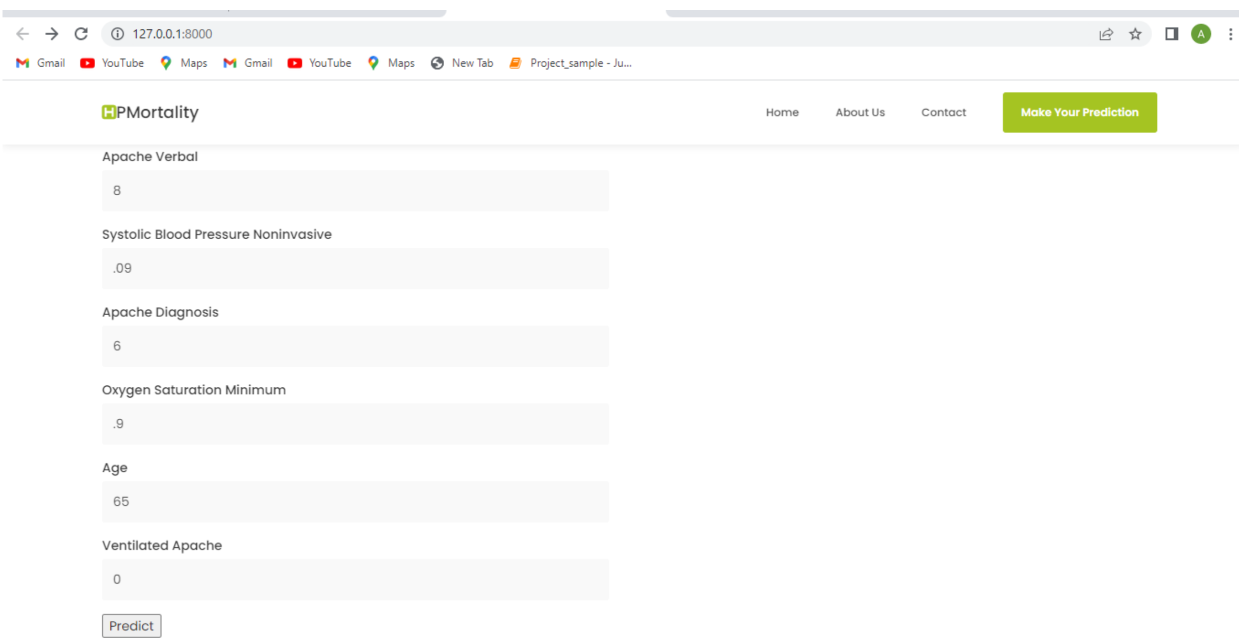
6.1.2 Login and register



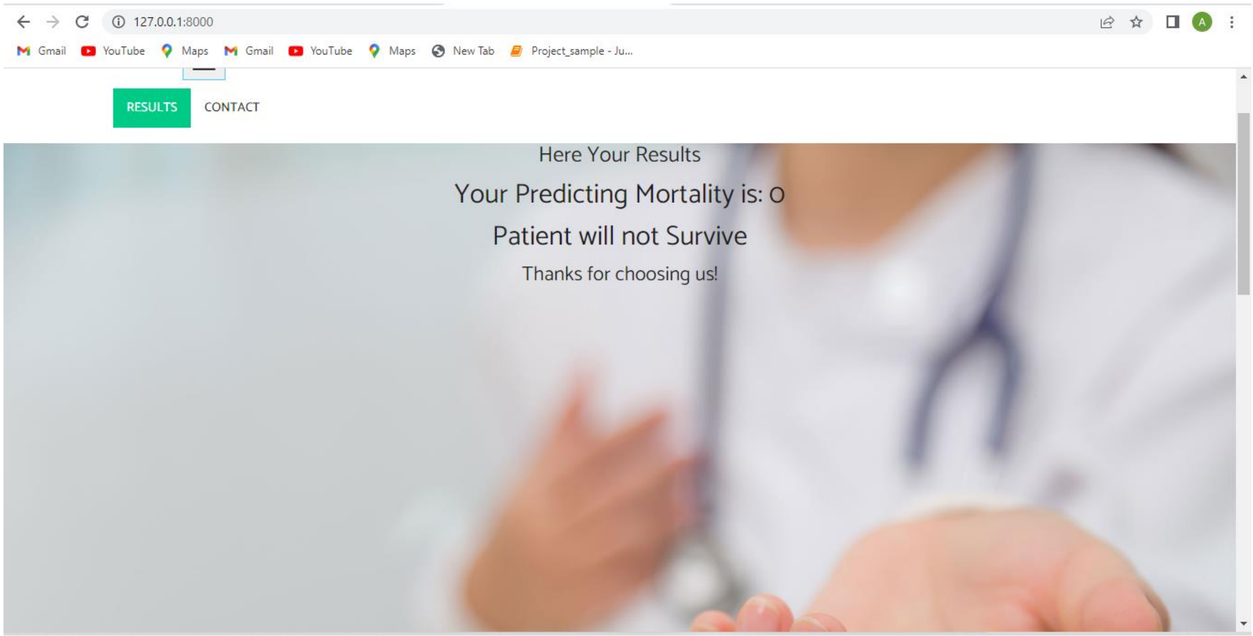


6.1.3 User severity check



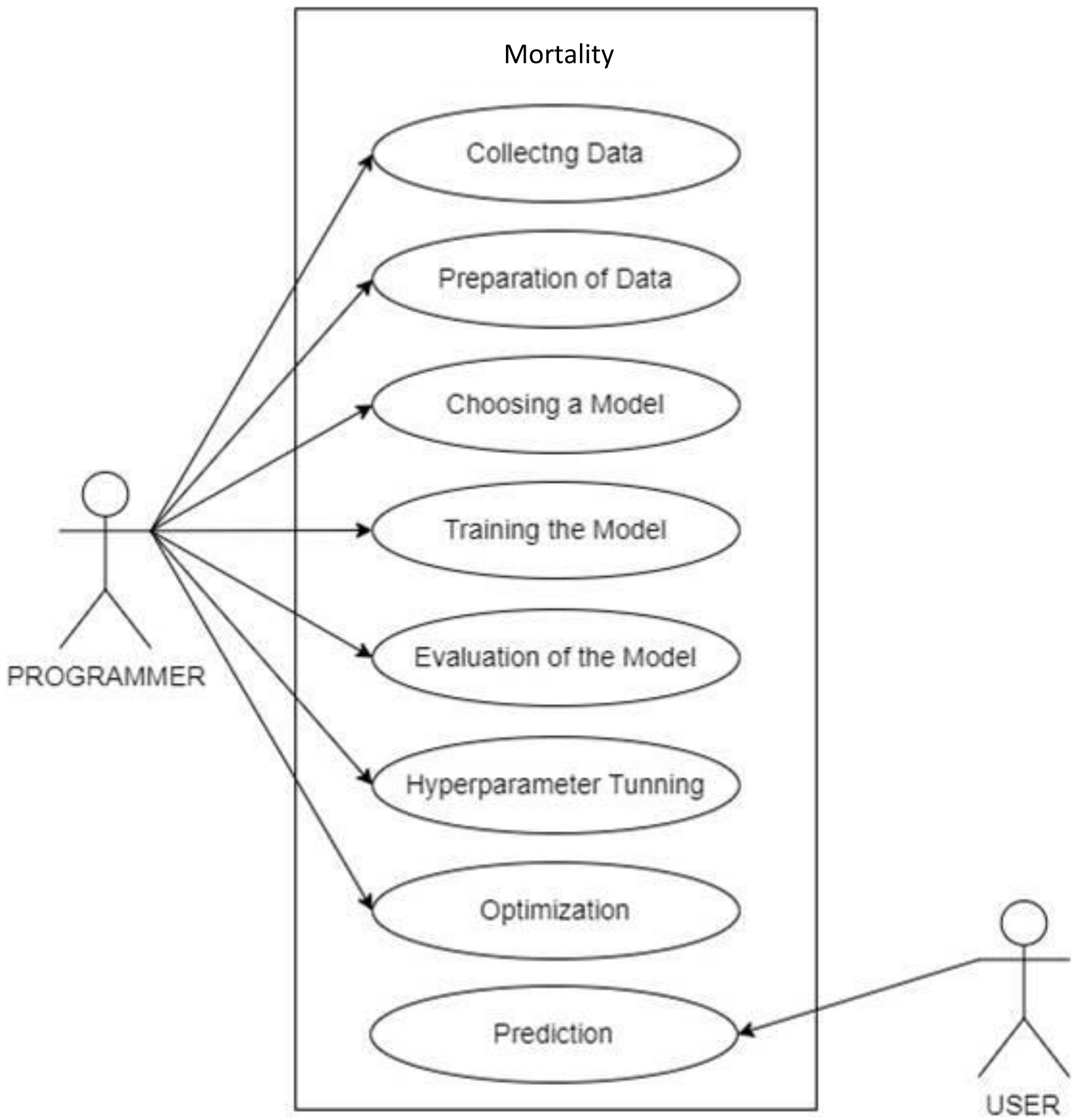


6.1.4 Output page

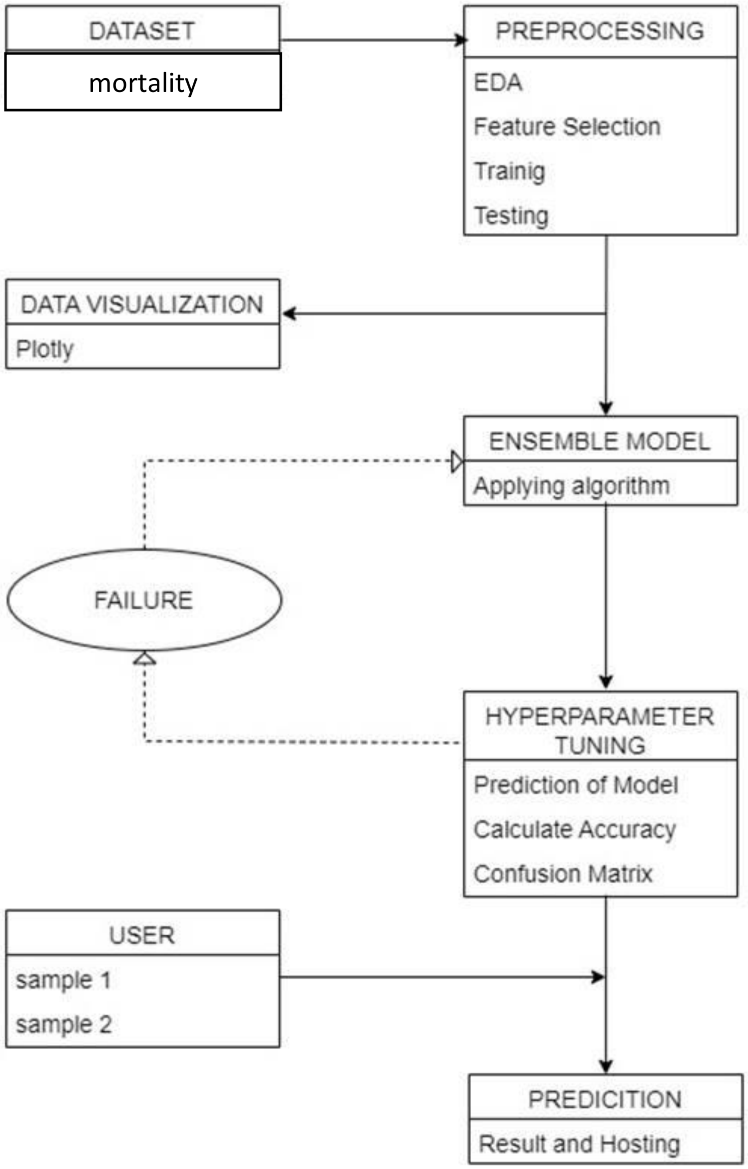


6.2. UML Diagram

6.2.1. Use case Diagram



6.2.2. Class Diagram



6.3. Database design

6.3.1. User registration table

Field name	Data Type	Constraints	Description
Id	Int (11)	Primary key	User id
User name	Varchar (20)	Not null	User name
Email	Varchar (50)	Not null	User e-mail
password	Varchar (20)	Not null	User password

6.3.2. Mortality table

Field name	Data Type	Constraints
id	Int (10)	Primary key
U_id	Int (10)	Foreign key
Hospital death	Int (10)	Not null
Icu death	Int (10)	Not null
Diagnosis	Int (10)	Not null
Age	Int (10)	Not null
SpO2 min	Int (10)	Not null
Sysbp min	Int (10)	Not null
Noninvasive	Int (10)	Not null
Verbal apache	Int (10)	Not null
Ventilated apache	Int (10)	Not null

TESTING

Test cases

Login Page

SL.NO	DESCRIPTION	INPUT	EXPECTED RESULT	PASSING CRITERIA

1	To check whether a user can login using wrong user name or password	Entered login page with wrong Username and Password	Display "incorrect username or password"	Should display "incorrect username or password"
2	To check whether a user can login using correct user name or password	Entered login page with correct Username and Password	For Users and doctor:Covid severity checking page will be loadedFor admin: Dashboard of admin must be loaded	For user and doctor: severity checking page should be loaded.For admin: Dashboard should be loaded.
3	To check whether a user can login with an empty login field	Click login button without entering details	Display "Please fill out this field"	Should displayed "Please fill out this field"

Registration

SL.NO	DESCRIPTION	INPUT	EXPECTED RESULT	PASSING CRITERIA
1	To check whether a person can sign	Clicking signup button without giving values to some fields	Display "Please fill out this field"	Should display "Please fill out this field"

	up with no details			
2	To check whether a person can sign up with correct details	Enter values and click register	Display “successfully registered”	Should displayed “successfully registered” and values must be added to database
3	To ensure strength of password	Enter short password	Display “must contain at least one number and upper or lower case and at least 8 or more characters”	Displayed “must contain at least one number and upper or lower case and at least 8 or more characters”
4	To check whether the same email can be used for different accounts.	Enter already used email	Display “account already exist”	Displayed “account already exist”
2	To check whether a person can reply to a message.	Reply for the message	Reply message	Must be able to reply.

7.1. Test report

Test CaseNo/ ID	No of test case run	No. of test case successful (%)	Pass/Fail	Expected result	Actual Result
TC-01	5	100	Pass	Output screen must be displayed	Output screen is displayed
TC-02	6	100	Pass	Output screen should be displayed and values must be stored DB	Output screen are displayed and values must be stored in DB
TC-03	15	90	Pass	Severity must be classified and displayed	Severity is classified and displayed
TC-04	8	100	Pass	Message must be sent and replied	Message is sent and replied

SYSTEM IMPLEMENTATION

Each of the people who wish to visit the system will land on a home page with the latest updated news which is dynamically updated by the admin.. And an about section is also there to know about the system.The first module deals with the login. A user / doctor / admin can access the system through the login module. Here the user will be able to login to the system by entering the login details and by clicking a button named as login after this the users will go to their corresponding page.The registration is a compulsory step for new users. If a user is new to the system then he must register with the system. Doctors are excluded from this step. Doctors are added by the admin to ensure the genuinity of the doctor. When a doctor is added by the admin, an email with the login details of the doctor will be sent to the doctor by the system.Severity check is the next step for both doctors and normal users after the login. After entering EHR records of patient in ICU and, we can check for the severity by clicking the check severity button. Admin dashboard consists of a potion to add, update or remove the news, doctor.

CONCLUSION & FUTURE ENHANCEMENT

A new web application is developed which consists of a trained machine learningmodel to predic t the mortality of patient admitted to ICU. This model will help both common people and doctors to check the mortality. This will help doctors, hospitals, and medical facilities in their decision m aking about which patients need to get attention first before others, and at the same time, to keep hospitals' resources for high-risk priority patients. From the analysis of mortality using machine l earning algorithms, mainly I used three different models such as decision tree, random forest, and gradient boost. All model perform entirely different. From all three model it found that Random F orest classifier provide a better model with accuracy 98% and F1-Score 98%.We can build a mod el with Random Forest classifier with the most common features apache_4a_icu_death_prob, ap ache_4a_hospital_death_prob, ventilated_apache, d1_sysbp_min Gcs_verbal_apache, d1_sysbp_ noninvasive_min , apache_3j_diagnosis , d1_spo2_min , d1_temp_min,apache_2_diagnosis. It is clear that the model improves accuracy and precision of mortality prediction with this dataset co mpared to existing dataset.

10. APPENDIX

10.1 Appendix A

10.1.1 Sample source code

Model.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from sklearn.metrics import classification_report
warnings.simplefilter('ignore')
data=pd.read_csv('dataset.csv')
data
data.info()
for column in data.columns:
    print(data[column].value_counts())
    print("*"*20)
data['gender']=data['gender'].map({'M':'1','F':'2'},na_action=None)
data.head()
data["icu_admit_source"].value_counts(normalize=True).plot(kind='bar'
sns.catplot(x="icu_admit_source",y="hospital_death",kind="bar",data=data,aspect=1.8)
plt.xticks(rotation=90)
plt.show()
if "icu_admit_type" not in data.columns:
    print("Column not found")
data["icu_type"].value_counts(normalize=True).plot(kind='bar')
sns.catplot(x="icu_type",y="hospital_death",kind="bar",data=data,aspect=1.8)
plt.xticks(rotation=90)
```

```
plt.show()

data[["apache_3j_bodysystem","apache_2_bodysystem"]].head(10)

data=data.drop(['apache_2_bodysystem','apache_3j_bodysystem','ethnicity','bmi_category','Un
named: 83'],axis=1)

data.head()

data.isnull().sum()

feature_na=[feature for feature in data.columns if data[feature].isnull().sum()>0]

feature_na

for feature in feature_na:

   print('{} has {} % missing
values'.format(feature,np.round(data[feature].isnull().sum()/len(data)*100,4)))

data['weight']= data['weight'].fillna(data['weight'].median())

data['bmi']= data['bmi'].fillna(data['bmi'].median())

data['age']= data['age'].fillna(data['age'].median())

data['height']= data['height'].fillna(data['height'].median())

data['gender']= data['gender'].fillna(data['gender'].median())

data['apache_2_diagnosis']=
data['apache_2_diagnosis'].fillna(data['apache_2_diagnosis'].median())

data['apache_3j_diagnosis']=
data['apache_3j_diagnosis'].fillna(data['apache_3j_diagnosis'].median())

data['arf_apache']= data['arf_apache'].fillna(data['arf_apache'].median())

data['gcs_eyes_apache']= data['gcs_eyes_apache'].fillna(data['gcs_eyes_apache'].median())

data['gcs_motor_apache']=
data['gcs_motor_apache'].fillna(data['gcs_motor_apache'].median())

data['gcs_unable_apache']=
data['gcs_unable_apache'].fillna(data['gcs_unable_apache'].median())

data['gcs_verbal_apache']=
data['gcs_verbal_apache'].fillna(data['gcs_verbal_apache'].median())

data['heart_rate_apache']= data['heart_rate_apache'].fillna(data['heart_rate_apache'].median())

data['intubated_apache']= data['intubated_apache'].fillna(data['intubated_apache'].median())

data['map_apache']= data['map_apache'].fillna(data['map_apache'].median())
```

```
data['resprate_apache']= data['resprate_apache'].fillna(data['resprate_apache'].median())

data['temp_apache']= data['temp_apache'].fillna(data['temp_apache'].median())

data['ventilated_apache']= data['ventilated_apache'].fillna(data['ventilated_apache'].median())

data['d1_diasbp_max']= data['d1_diasbp_max'].fillna(data['d1_diasbp_max'].median())

data['d1_diasbp_min']= data['d1_diasbp_min'].fillna(data['d1_diasbp_min'].median())

data['d1_diasbp_noninvasive_max']=
data['d1_diasbp_noninvasive_max'].fillna(data['d1_diasbp_noninvasive_max'].median())

data['d1_diasbp_noninvasive_min']=
data['d1_diasbp_noninvasive_min'].fillna(data['d1_diasbp_noninvasive_min'].median())

data['d1_heartrate_max']= data['d1_heartrate_max'].fillna(data['d1_heartrate_max'].median())

data['d1_heartrate_min']= data['d1_heartrate_min'].fillna(data['d1_heartrate_min'].median())

data['d1_mbp_max']= data['d1_mbp_max'].fillna(data['d1_mbp_max'].median())

data['d1_mbp_min']= data['d1_mbp_min'].fillna(data['d1_mbp_min'].median())

data['d1_mbp_noninvasive_max']=
data['d1_mbp_noninvasive_max'].fillna(data['d1_mbp_noninvasive_max'].median())

data['d1_mbp_noninvasive_min']=
data['d1_mbp_noninvasive_min'].fillna(data['d1_mbp_noninvasive_min'].median())

data['d1_resprate_max']= data['d1_resprate_max'].fillna(data['d1_resprate_max'].median())

data['d1_resprate_min']= data['d1_resprate_min'].fillna(data['d1_resprate_min'].median())

data['d1_spo2_max']= data['d1_spo2_max'].fillna(data['d1_spo2_max'].median())

data['d1_spo2_min']= data['d1_spo2_min'].fillna(data['d1_spo2_min'].median())

data['d1_sysbp_max']= data['d1_sysbp_max'].fillna(data['d1_sysbp_max'].median())

data['d1_sysbp_min']= data['d1_sysbp_min'].fillna(data['d1_sysbp_min'].median())

data['d1_sysbp_noninvasive_max']=
data['d1_sysbp_noninvasive_max'].fillna(data['d1_sysbp_noninvasive_max'].median())

data['d1_sysbp_noninvasive_min']=
data['d1_sysbp_noninvasive_min'].fillna(data['d1_sysbp_noninvasive_min'].median())

data['d1_temp_min']= data['d1_temp_min'].fillna(data['d1_temp_min'].median())

data['h1_diasbp_max']= data['h1_diasbp_max'].fillna(data['h1_diasbp_max'].median())

data['h1_diasbp_min']= data['h1_diasbp_min'].fillna(data['h1_diasbp_min'].median())

data['h1_diasbp_noninvasive_max']=
data['h1_diasbp_noninvasive_max'].fillna(data['h1_diasbp_noninvasive_max'].median())
```

```
data['h1_diasbp_noninvasive_min']=
data['h1_diasbp_noninvasive_min'].fillna(data['h1_diasbp_noninvasive_min'].median())

data['h1_heartrate_max']= data['h1_heartrate_max'].fillna(data['h1_heartrate_max'].median())

data['h1_heartrate_min']= data['h1_heartrate_min'].fillna(data['h1_heartrate_min'].median())

data['h1_mbp_max']= data['h1_mbp_max'].fillna(data['h1_mbp_max'].median())

data['h1_mbp_min']= data['h1_mbp_min'].fillna(data['h1_mbp_min'].median())

data['h1_mbp_noninvasive_max']=
data['h1_mbp_noninvasive_max'].fillna(data['h1_mbp_noninvasive_max'].median())

data['h1_mbp_noninvasive_min']=
data['h1_mbp_noninvasive_min'].fillna(data['h1_mbp_noninvasive_min'].median())

data['h1_resprate_max']= data['h1_resprate_max'].fillna(data['h1_resprate_max'].median())

data['h1_resprate_min']= data['h1_resprate_min'].fillna(data['h1_resprate_min'].median())

data['h1_spo2_max']= data['h1_spo2_max'].fillna(data['h1_spo2_max'].median())

data['h1_spo2_min']= data['h1_spo2_min'].fillna(data['h1_spo2_min'].median())

data['h1_sysbp_max']= data['h1_sysbp_max'].fillna(data['h1_sysbp_max'].median())

data['h1_sysbp_min']= data['h1_sysbp_min'].fillna(data['h1_sysbp_min'].median())

data['h1_sysbp_noninvasive_max']=
data['h1_sysbp_noninvasive_max'].fillna(data['h1_sysbp_noninvasive_max'].median())

data['h1_sysbp_noninvasive_min']=
data['h1_sysbp_noninvasive_min'].fillna(data['h1_sysbp_noninvasive_min'].median())

data['d1_glucose_max']= data['d1_glucose_max'].fillna(data['d1_glucose_max'].median())

data['d1_glucose_min']= data['d1_glucose_min'].fillna(data['d1_glucose_min'].median())

data['d1_potassium_max']=
data['d1_potassium_max'].fillna(data['d1_potassium_max'].median())

data['d1_potassium_min']=
data['d1_potassium_min'].fillna(data['d1_potassium_min'].median())

data['apache_4a_hospital_death_prob']=
data['apache_4a_hospital_death_prob'].fillna(data['apache_4a_hospital_death_prob'].median())

data['apache_4a_icu_death_prob']=
data['apache_4a_icu_death_prob'].fillna(data['apache_4a_icu_death_prob'].median())

data['aids']= data['aids'].fillna(data['aids'].median())

data['cirrhosis']= data['cirrhosis'].fillna(data['cirrhosis'].median())

data['diabetes_mellitus']= data['diabetes_mellitus'].fillna(data['diabetes_mellitus'].median())
```

```
data['hepatic_failure']= data['hepatic_failure'].fillna(data['hepatic_failure'].median())
data['d1_temp_max']= data['d1_temp_max'].fillna(data['d1_temp_max'].median())
data['immunosuppression']=
data['immunosuppression'].fillna(data['immunosuppression'].median())
data['leukemia']= data['leukemia'].fillna(data['leukemia'].median())
data['lymphoma']= data['lymphoma'].fillna(data['lymphoma'].median())
data['solid_tumor_with_metastasis']=
data['solid_tumor_with_metastasis'].fillna(data['solid_tumor_with_metastasis'].median())
for feature in feature_na:
    print('{} has {} % missing
values'.format(feature,np.round(data[feature].isnull().sum()/len(data)*100,4)))
data_cat = data.select_dtypes(include = [np.object])
data_cat.columns.
for cat in data_cat:
    print("{}: {}".format(cat,data[cat].nunique()))
from sklearn.preprocessing import LabelEncoder
print('Transform all String features to category.\n')
for usecol in data_cat:
    data[usecol] = data[usecol].astype('str')
    data[usecol] = data[usecol].astype('str')

    #Fit LabelEncoder
    le = LabelEncoder().fit(
            np.unique(data[usecol].unique().tolist()+
                    data[usecol].unique().tolist()))
    data[usecol] = le.transform(data[usecol])+1
    data[usecol] = data[usecol].replace(np.nan, 0).astype('int').astype('category')
   data.isnull().sum()
x=data.drop(['hospital_death','hospital_id','encounter_id','icu_id','patient_id'],axis=1)
y=data['hospital_death']
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state=42)
print("Shape of x_train:", x_train.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of x_test:", x_test.shape)
print("Shape of y_test:", y_test.shape)
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state=42)
dt=DecisionTreeClassifier(random_state=42)
dt=dt.fit(x_train,y_train)
pred=dt.predict(x_test)
dt.score(x_test,y_test)
data['hospital_death'].value_counts()
import imblearn
from imblearn.over_sampling import RandomOverSampler
ros = RandomOverSampler(random_state=42)
x_ros, y_ros = ros.fit_resample(x_train, y_train)
print('Original dataset shape', y_train.value_counts())
print('Resample dataset shape', y_ros.value_counts())
class_count_0, class_count_1 = data['hospital_death'].value_counts()
class_0 = data[data['hospital_death'] == 0]
class_1 = data[data['hospital_death'] == 1]
print('class 0:', class_0.shape)
print('class 1:', class_1.shape)
class_0_under = class_0.sample(class_count_1)
test_under = pd.concat([class_0_under, class_1], axis=0)
print("total class of 1 and0:",test_under['hospital_death'].value_counts())
test_under['hospital_death'].value_counts().plot(kind='bar', title='count (target)')
x=data.drop(['hospital_death','hospital_id','encounter_id','icu_id','patient_id'],axis=1)
```

```
y=test_over
x.head()
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_ros, y_ros,test_size=0.2,random_state=42)
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
rf=RandomForestClassifier(n_estimators=10,random_state=10)
rf=rf.fit(x_train,y_train)
ypred=rf.predict(x_test)
print(classification_report(y_test,ypred))
print(confusion_matrix(y_test,rf.predict(x_test)))
from sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
param_grid={
'max_depth':range(1,5),
'min_samples_leaf':range(2,5)
}
n_folds=5
ranfc=GradientBoostingClassifier()
grid_search=GridSearchCV(estimator=gbc,param_grid=param_grid,cv=n_folds,verbose=1)
grid_search.fit(x_train,y_train)
cv_result=pd.DataFrame(grid_search.cv_results_)
cv_result.sort_values('mean_test_score',ascending=False)[0:5]

plt.xlabel('FEATURE IMPORTANCE ')
plt.ylabel('FEATURE NAMES')
important_features=pd.DataFrame({'Features':x_train.columns,'Importance':rf.feature_importa
nces_})
print(important_features.nlargest(20,'Importance'))
```

Index.html

```
% load static %}
<!DOCTYPE html>
<html lang="en">
<head>

     <title>PMortality</title>
<!--

Template 2098 Health

http://www.tooplate.com/view/2098-health

-->
     <meta charset="UTF-8">
     <meta http-equiv="X-UA-Compatible" content="IE=Edge">
     <meta name="description" content="">
     <meta name="keywords" content="">
     <meta name="author" content="Tooplate">
     <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">

     <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
     <link rel="stylesheet" href="{% static 'css/font-awesome.min.css' %}">
     <link rel="stylesheet" href="{% static 'css/animate.css' %}">
     <link rel="stylesheet" href="{% static 'css/owl.carousel.css' %}">
     <link rel="stylesheet" href="{% static 'css/owl.theme.default.min.css' %}">

     <!-- MAIN CSS -->
     <link rel="stylesheet" href="{% static 'css/tooplate-style.css' %}">
```

```
</head>
<body id="top" data-spy="scroll" data-target=".navbar-collapse" data-offset="50">

    <!-- PRE LOADER -->
    <section class="preloader">
        <div class="spinner">

            <span class="spinner-rotate"></span>

        </div>
    </section>



    <!-- HEADER -->
    <header>
        <div class="container">
            <div class="row">

                <div class="col-md-4 col-sm-5">
                    <p>Welcome to Mortality Rate Prediction</p>
                </div>

                <div class="col-md-8 col-sm-7 text-align-right">
                    <span class="phone-icon"><i class="fa fa-phone"></i> 010-060-0160</span>
                    <span class="date-icon"><i class="fa fa-calendar-plus-o"></i> 6:00 AM -
10:00 PM (Mon-Fri)</span>
                    <span class="email-icon"><i class="fa fa-envelope-o"></i> <a
href="#">info@company.com</a></span>
                </div>
```

```
            </div>
        </div>
    </header>



    <!-- MENU -->
    <section class="navbar navbar-default navbar-static-top" role="navigation">
        <div class="container">

            <div class="navbar-header">
                <button class="navbar-toggle" data-toggle="collapse" data-target=".navbar-
collapse">
                    <span class="icon icon-bar"></span>
                    <span class="icon icon-bar"></span>
                    <span class="icon icon-bar"></span>
                </button>

                <!-- lOGO TEXT HERE -->
                <a href="index.html" class="navbar-brand"><i class="fa fa-h-
square"></i>PMortality</a>
            </div>

            <!-- MENU LINKS -->
            <div class="collapse navbar-collapse">
                <ul class="nav navbar-nav navbar-right">
                    <li><a href="#top" class="smoothScroll">Home</a></li>
                    <li><a href="#about" class="smoothScroll">About Us</a></li>
                    <!--<li><a href="#team" class="smoothScroll">Doctors</a></li>-->
                   <!-- <li><a href="#news" class="smoothScroll">News</a></li> -->
                    <li><a href="#google-map" class="smoothScroll">Contact</a></li
```

10.2 Appendix B

10.2.1 Webliography

[1]. https://www.kaggle.com/adityakadiwal/mortality.[accessed 10 April 2022].

[2]. https://www.djangoproject.com/ [accessed on 02 June 2022].

[3]. https://www.python.org/ [accessed on 25 April 2022].

[4]. https://scikit-learn.org/stable/ [accessed on 06 May 2022].

[5]. https://getbootstrap.com/ [accessed on 08 June 2022].

10.2.2 Bibliography

[1] Morid MA, Sheng ORL, Abdelrahman S. PPMF: A patient-based predictive modelling framework for early ICU mortality prediction. arXiv:1704.07499; 2017.

[2] Johnson AE, Pollard TJ, Mark RG. Reproducibility in critical care: a mortality prediction case study. In: Machine Learning for Healthcare Conference; 2017. p. 361–76.

[3] Tan R, Ding S, Pan J, Qiu Y. ICU mortality prediction based on key risk factors identification. In: International Conference on Health Information Science; 2019. p. 89–97.

[4] Sadeghi R, Banerjee T, Romine W. Early hospital mortality prediction using vital signals. Smart Health 2018;9:265–74.

[5] Luo Y, Xin Y, Joshi R, Celi L, Szolovits P. Predicting ICU mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements. In: Thirtieth AAAI Conference on Artificial Intelligence; 2016.