Enhancing ICU Patient Outcomes Trough Machine Learning and Ensemble Technique

By
Meraj Ahmed
193-15-2964

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the
Requirements forthe **Degree of Bachelor of Science in**
Computer Science and Engineering

Supervised by

Tapasy Rabeya
Lecturer (Senior
Scale)
Department of Computer Science and
EngineeringDaffodil International
University

Co-Supervised by

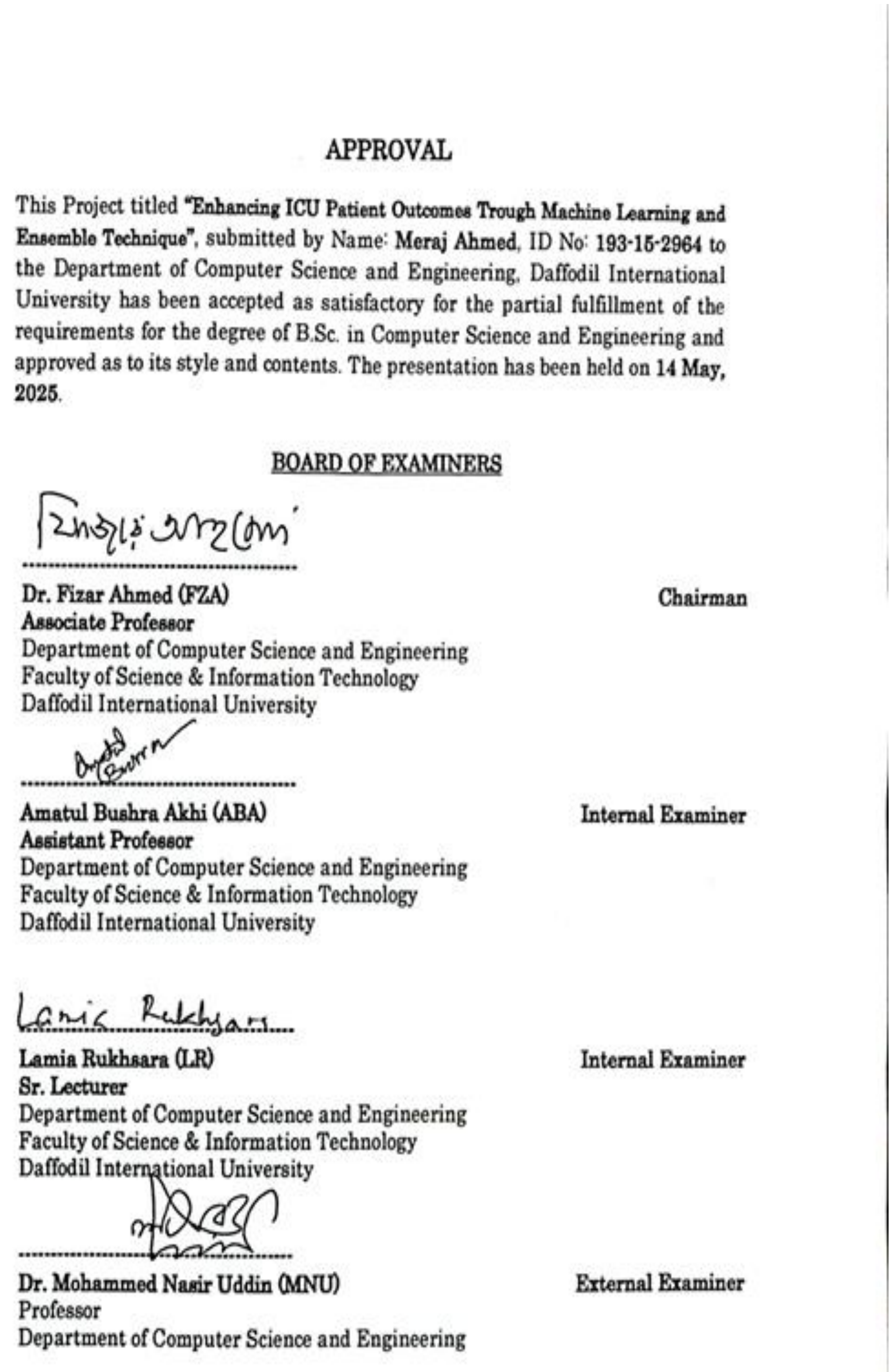
Nafiz Ahmed
Emon

Lecturer
Department of Computer Science and
EngineeringDaffodil International
University

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

May 14, 2025



APPROVAL

This Project titled **"Enhancing ICU Patient Outcomes Trough Machine Learning and Ensemble Technique"**, submitted by Name: **Meraj Ahmed**, ID No: **193-15-2964** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **14 May, 2025**.

BOARD OF EXAMINERS

....

Dr. Fizar Ahmed (FZA) — **Chairman**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

....

Amatul Bushra Akhi (ABA) — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Lamia Rukhsara
....

Lamia Rukhsara (LR) — **Internal Examiner**
Sr. Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

....

Dr. Mohammed Nasir Uddin (MNU) — **External Examiner**
Professor
Department of Computer Science and Engineering

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Tapasy Rabeya, Lecturer (Senior Scale)**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Tapasy Rabeya
Lecturer (Senior Scale)
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Nafiz Ahmed Emon
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Meraj

Meraj Ahmed
Student ID: 193-15-2964
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has as- sisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Tapasy Rabeya**, **Lecturer (Senior Scale)**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Machine Learning** to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to **Dr. Sheak Rashed Haider Noori** the Head of the Department of Computer Science and Engineering, for her kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

In order to identify the mortality risk and estimate the death rate of ICU patients, machine learning (ML) and ensemble learning approaches are utilized to analyse a variety of patient data and provide an accurate prognosis. The mortality rate of ICU patients nowadays is very high. If we can identify the reason as early as possible then we can start diagnosis as early as possible. First, relevant attributes such as test results, symptoms, and demographic data are taken out of patient files. ML algorithms like logistic regression, decision trees, and support vector machines classify patients into low- and high-risk categories during the mortality risk identification stage. Through model aggregation, ensemble techniques like random forests and gradient boosting improve predictive performance. Regression models such as ridge regression, neural networks, and linear regression evaluate the probability of death within given time frames to predict mortality rates. These estimates are then improved using ensemble learning strategies like stacking or bagging. This abstraction, which improves patient outcomes in the ICU, enables healthcare workers to quickly identify at-risk patients and make well-informed decisions through careful feature engineering, model hyperparameter tuning, and cross-validation. The best accuracy comes with logistic regression (90%) and 2nd highest accuracy comes with both random forest and KNN which is 89%. The results of xgboost and adaboost are 87.07% and 84%.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The reason, objectives, approach and expected outcomes for using both machine learning and ensemble methods to predict ICU mortality are discussed in this chapter. Having an approach that finds risk early helps raise critical care and lower death rates.

1.1 Introduction

By combining machine learning (ML) and ensemble learning, staff members in ICUs focus on improving patient outcomes. In critical care units, advance equipment is employed by the staff. Because the ICU mortality rate is quite high, early detection of risk factors helps guide prompt interventions. With the help of machine learning and ensemble models, analyzing a lot of patient data produces a more accurate prognosis that works to extend life and reduce the numbers of deaths among patients. These algorithms study complex patterns in the data to provide useful information about how a patient will respond and guide early care. Therefore, business forecasts are made even better using random forests and gradient boosting ensemble methods. Using more than one design increases the ability of the algorithms to predict results correctly and helps patients achieve better outcomes. Regression models measure the chances of death during a specified period. Forms of these techniques are ridge regression, neural networks and linear regression. To help plan patient care, these models use patient data and history to approximate the likelihood of patient death. The mortality rate is estimated with greater accuracy when stacking and bagging are combined with ensemble learning. These methods improve the dependability of mortality predictions by removing bias from each model and using them together. By using abstracted machine learning and several techniques, healthcare teams can quickly find at-risk patients and act on their behalf. When advanced analytics are used, medical professionals use feature engineering to improve hyperparameter optimization which helps them deliver better care and strengthen outcomes. A comprehensive study on the quality of models reveals that logistic regression performs with the greatest accuracy, outperforming All in all, this work explains how the merging of ML and ensemble strategies had a major impact on how ICU patients are managed. Analyzing data can help healthcare professionals spot those who need extra care, start support early and improve ICU patient care. If these techniques keep getting better, people working in critical care medicine have reason to be optimistic about the future.

1.2 Motivation

Using machine learning and ensemble learning in ICUs helps to alert doctors early, choose the best treatment, organize needed resources and save on healthcare spending. Thanks to advanced procedures, researchers get access to more information which makes their decisions more reliable and their treatments for patients more uniform. By using many types of patient information, these models can observe new patterns and patterns between conditions. This lets us learn a lot more about how our bodies function. They make sure patients are protected and safe by producing a notice if any of their monitor readings change. Besides using these models to track their own workings and remain compliant, hospitals are urged to be responsible, transparent and impartial. Applying machine learning and ensemble learning approaches in intensive care units will improve patient care and lower the time needed for healthcare.

1.3 Objectives

With advanced machine learning and ensemble learning, we hope to find out which factors might cause patient death sooner within the intensive care unit. This study will introduce random forests, gradient boosting, stacking and bagging to predictive models like logistic regression, decision trees and support vector machines as it looks at different clinical data. To get accurate results in critical care and lower mortality, important features need to be chosen, parameters should be set and the process needs to be tested on new batches of data.

1.4 Methodology

In this study, multiple machine learning and ensemble learning models are applied to examine the likelihood of a patient dying in the ICU. To begin, medical data is taken out and prepared, including lab results, a list of symptoms and demographic data about the patient. After that, patients are divided into those with a high or low risk level based on their characteristics. A variety of machine learning methods such as support vector machines, logistic regression and decision trees, are used for carrying out classification. These models, Random Forest and Gradient Boosting, improve accuracy by combining a set of models. Using methods such as ridge regression, linear regression and neural networks, experts find estimates for death rates. Stacking and bagging make it possible to improve the forecasts even more. Guaranteeing that results are correct, the model building steps use cross-validation, feature engineering and adjusting hyperparameters. The best predicted accuracy was obtained by logistic regression, with 90%, followed by XGBoost at 87.07% and AdaBoost at 84%. The models Random Forest and KNN followed, achieving results of 89% accuracy.

1.5 Project Outcome

The objective of this study is to give medical practitioners a way to estimate with accuracy how likely it is that a patient in the intensive care unit will die. By applying machine learning and ensemble techniques on patient data, the system discovers critical cases promptly so that faster medical action can be taken. As a result, doctors will succeed in earlier detection of people who might develop complications early and will be able to create useful predictions to support their choices. An improvement in the number of lives saved by spotting symptoms early. A trustworthy system helpful in supporting decisions that fits all types of hospitals. With this system, critical care units might decrease how long procedures take and spend, saving the most important resource — lives.

Chapter 2

Background

How using machine learning plays a role in forecasting ICU patient mortality is discussed here. It stresses that using both quick risk spotting and combined methods can make the prognosis and treatment better in critical care for patients.

2.1 Introduction

Many seriously sick people rely on treatment in ICUs, yet not everyone survives. When we find patients may be in danger of death, doctors can work more effectively to save them. Because patient information is stored digitally, their laboratory test results can be investigated for patterns that mean something. At present, more hospitals are using machine learning technology to help predict if ICU patients are going to die. Floridas surveillance system uses logistic regression, decision trees and support vector machines for this classification. Using Random Forest and Gradient Boosting, you can improve a model's accuracy by combining it with other models. The regression methods ridge regression, neural networks and linear regression also try to predict the possibility of a death during different time spans. If models are grouped together by bagging and stacking, predictions tend to be more trustworthy. Using the results from data, health professionals can decide on the best ways to care for their patients.

2.2 Literature Review

Previously simple scoring was used to predict mortality; today, more complex data-based approaches are used. Initially, training datasets used logistic regression and SVMs to compare the demographic information, vital signs and laboratory test values of survivors with those of non-survivors during their first full day in the hospital. Such prediction methods usually achieved mid-80% results which confirmed that important markers like age, blood pressure and lab results can divide risk levels well. Because tree based models handle non linear effects well and fight overfitting, researchers chose to use them for their studies. These trend models provided easy to understand predictions; however, they were susceptible to instability.

Averaging results from numerous trees built based on bootstrap samples and select features produced consistent outcomes and often raised accuracy by several percentage points as compared to individual trees. To increase performance, methods were designed to gradually match new models to the errors observed earlier. By using AdaBoost, any errors were corrected more by improving incorrect classifications. With gradient boosting, the algorithm aims to find solutions by directly reducing the loss. Implemented as XGBoost, this technique adds regularization and parallel computing, so its accuracy measured by AUC improved to 0.90 or higher among large ICU patients. Besides being used for classification, regression-based models can assess the chances of death over different time limits.

While careful tuning and larger datasets are necessary to prevent overfitting, neural networks in particular are capable of learning intricate patterns across a wide range of characteristics. Multiple algorithm ensemble frameworks have been studied in more recent work. By training models in parallel, bagging (like Random Forest) lowers variance, but boosting (like XGBoost, AdaBoost) lowers bias by concentrating on challenging situations one after the other. This is furthered by stacking, which trains a meta learner to combine the predictions of multiple base models, frequently producing slight increases in predictive strength.

Simple scoring systems have given way to sophisticated data-driven techniques for predicting death in intensive care unit patients. Early approaches used statistical models such as logistic regression and support vector machines (SVMs) to assess patient demographics, vital signs, and laboratory results from the first 24 hours of admission in order to distinguish survivors from non-survivors. These methods frequently produced accuracies in the mid-80% range, proving that age, blood pressure, and important laboratory markers could reliably stratify risk. Researchers used tree-based models to prevent overfitting and capture nonlinear interactions. Although they provided clear rule-based predictions, single decision trees were unstable. Neural networks in particular may learn complex patterns across a wide variety of properties, but overfitting must be avoided with careful tuning and larger datasets. More recent work has examined several algorithm ensemble frameworks. Bagging (like Random Forest) reduces variance by training models in parallel, whereas boosting (like XGBoost, AdaBoost) reduces bias by focusing on difficult scenarios sequentially. This is enhanced by stacking, which teaches a meta learner to aggregate predictions from several base models, often leading to modest gains in predictive power.

2.2.1 Similar Applications

In healthcare and a few other areas, using machine learning and ensembles to identify risks and accurately predicts outcomes works well. Estimating the early stages of sepsis is an important task handled by ensemble learning in intensive care. Responding right away is important because sepsis can quickly causes a patient to become very sick if it is not found on time. Ensemble models that combiness decision trees, gradient boosting and logistic regression can be used to monitor both vital signs and laboratory values and so help identify patients at risks of sepsis. This early spotting enables doctors to take action before the illness puts your dog's life at risk.

If patients with heart failure remain at the hospital for more than one visit after being sent home, hospitals are penalized. With models like Random Forests, XGBoost and voting classifiers, clinicians predict whether a patient will be readmitted to the hospital by reviewing their hospital history, what drugs they take and the results of their tests. As a result, individualized care after discharge and focused discharge planning help to lower re-admissions. Ensemble techniques are being applied to forecast surgical problems. Every once in a while, infections or bleeding may occur after big surgery. To forecast possible in-surgery difficulties, physicians use ensemble models which bring together the useful aspects of support vector machines, neural networks and gradient boosting. As a result, decisions made during surgery are likely to benefit the patient.

Ensemble learning allows doctors to identify how diabetes or chronic kidney disease might unfold over the course of an individual's life. Grouping information from patients' backgrounds, activities and test findings helps healthcare professionals locate individuals who may see their disease advance rapidly. As a result, they are better able to personalize treatment and monitoring which helps manage the condition over the long term.

Ensemble approaches are found in many areas besides healthcare. A good example is using them in finance to examine credit risk, allowing organizations to estimate if a borrower might be unable to pay back a loan. To help reduce downtime and cuts in maintenance fees, some manufacturing companies build predictive models that can foretell equipment failure ahead of time. In a similar way, using ensembles in retail helps predict customer churn which enables businesses to act in advance to maintain top clients. Using these techniques might greatly enhance important decisions in several industries because they combine various models to make predictions more accurate.

2.2.2 Related Research

Iwase.S, et.al showed a Prediction algorithm for ICU mortality and length of stay using machine learning. In a study conducted at Chiba University Hospital, machine learning was used to accurately predict the length of ICU stay and mortality in 12,747 patients. AUCs for mortality and ICU stay duration were 0.945 and 0.881–0.889, respectively, using random forest classifiers. Lactate dehydrogenase (LDH) was shown to be a critical variable in the prediction process and patient classification based on mortality risk. The dataset, which displayed similar patient backgrounds and results, was divided into 80% training and 20% test cohorts[1].

Yun.K,et.al proposed Prediction of Mortality in Surgical Intensive Care Unit Patients Using Machine Learning Algorithms. With the addition of new factors, earlier ICU mortality prediction models like APACHE II and SAPS II developed into APACHE IV and SAPS III. With 43 variables used, machine learning produced results with high accuracy in outcome categorization, with decision trees outperforming neural networks (AUC = 0.96 vs. 0.80). Dopamine dosage, serum albumin levels, and prenatal nutritional intake have been shown to be important indicators of SICU patient death[2].

Zhang.G,et.al described redicting sepsis in-hospital mortality with machine learning: a multi-center study using clinical and inflammatory biomarkers. The SHAP technique improved the interpretability of the model, and this study demonstrated the promise of machine learning in early outcome prediction for patients with sepsis. Twelve important clinical factors were used to create models from 3535 eligible sepsis patients; eXtreme Gradient Boosting (XGBoost) performed best (AUC = 0.94, F1 score = 0.937). Age, AST, invasive ventilation, and serum urea nitrogen were identified using feature importance analysis as critical predictors, and SHAP analysis helped to clarify model predictions[3].

Lue.C,et.al proposed A machine learning-based risk stratification tool for in-hospital mortality of intensive care unit patients with heart failure. Individualised treatment decisions were aided by a machine learning-based mortality risk assessment that used ICU clinical data and surpassed standard risk scores and linear regression. The model outperformed traditional techniques, as evidenced by its AUC of 0.831 and good calibration; its external validation yielded an AUC of 0.809. Decision curve study confirmed that risk stratification demonstrated specificity over a range of death rates, from very low to very high[4].

Meyer.H,et.al showed Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. We collected data on 12,616 patients with 15,615 ICU admissions for the model's development. The model development dataset included 11 492 patients with 14 190 ICU hospitalisations. Of these, we assigned about 20% of patients (2299 patients with 2825 admissions) to the holdout test dataset (figure 2). Using information from their initial ICU stay, the table displays the baseline characteristics of the patients in the training dataset and holdout test dataset. Within the development dataset, 4816 (41·9%) of the participants were female, with a median age of 65 years (IQR 52–75). Ninety days after ICU admission, 1815 (15·7%) patients died in the ICU, 3389 (29·5%) in the hospital, and 3802 (33·1%) in the ICU[5].

Kong.g, et. al showed Using machine learning methods to predict in-hospital mortality of sepsis patients in the ICU. Using MIMIC III data, this study created machine learning models to forecast the probability of in-hospital mortality for sepsis patients in intensive care units. Using the current SAPS II scoring method, four models (LASSO, RF, GBM, and LR) were assessed; GBM performed the best (AUROC 0.845, Brier score 0.104). When compared to SAPS II, the machine learning models showed superior calibration and accuracy, especially when it came to detecting high-risk patients[6].

Veith.N, et.al proposed Machine Learning-based Prediction of ICU Patient Mortality at Time of Admission. This study used more than 58,000 admissions from the MIMIC III database and machine learning techniques to forecast the mortality of ICU patients. The generated models outperform previous research, which frequently concentrate on certain patient subsets, using just patient administrative data available at admission. These models show promise for early death prediction that may be applied to all patients 12 in the intensive care unit. In order to increase accuracy, future research will concentrate on improving models and investigating new, easily-acquired characteristics at admission[7].

Nuno.B, et.al showed Developing machine learning models for prediction of mortality in the medical intensive care unit. In order to predict ICU mortality, this study compared machine learning models (XGB, BN, NB) to traditional scoring methods (SAPS, SOFA, LODS, OASIS). It found that XGB performed the best (AUROC 0.919, Brier score 0.072). The ML models performed far better in terms of accuracy, calibration, and discriminating than the conventional systems. In the MICU, XGB and BN in particular showed notable advancements in patient deterioration monitoring[8].

Lorenzoni.G, et.al proposed COVID-19 ICU mortality prediction: a machine learning approach using SuperLearner algorithm. Using predetermined clinical characteristics, this study created machine learning models to predict ICU mortality in COVID-19 patients. Age was the most important predictor across all models, with training accuracy ranging from 0.72 to 0.90 and cross-validation performance from 0.75 to 0.85. The models provide a trustworthy resource for COVID-19 ICU patients' mortality prediction[9].

Huang.B, et.al described Mortality prediction for patients with acute respiratory distress syndrome based on machine learning: a population-based study. The MIMIC-III and eICU-CRD databases were used in this study to create a machine learning-based mortality prediction model for ARDS patients. With AUROCs up to 0.905, the random forest (RF) model greatly outperformed the existing scoring systems (SAPS-II, APPS, OSI, and OI), with platelet count and lactate level emerging as important predictors. When it came to ARDS patients' in-hospital, 30-day, and 1-year mortality predictions, the RF model proved to have better accuracy and calibration[10].

Subudhi.S,et.al proposed Comparing machine learning algorithms for predicting ICU admission and mortality in COVID-19. This study used data from the Mass General Brigham Healthcare database to assess 18 machine learning methods for predicting COVID-19 patient mortality and ICU admission. When it came to forecasting 28-day mortality and 5-day ICU admission, ensemble-based models performed better. Important indicators for ICU admission were CRP, LDH, and O2 saturation; for death, eGFR, neutrophil, and lymphocyte percentages were significant predictors, suggesting their possible use in clinical decision-making in the context of infectious disease outbreaks such as COVID-19[11].

Nile.X,et.al proposed Mortality Prediction in Cerebral Hemorrhage Patients Using Machine Learning Algorithms in Intensive Care Units. This study examines how well many machine learning algorithms predict early mortality in patients with intracerebral haemorrhage (ICH) in comparison to traditional clinical ratings like the ICH score and APACHE II. Machine learning techniques achieved the highest prediction accuracy of 81%, demonstrating their capacity to manage massive data and complicated variable 13 interactions thoroughly. The study emphasises how machine learning can enhance clinical decision-making and predictive accuracy in the management of ICH[12].

Awad.A,et.al showed Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach. This study uses the MIMIC-II database to build machine learning models in order to solve the problem of early mortality prediction in ICU patients. Based on baseline demographic, physiological, vital sign, and laboratory data, the models—Random Forest (RF), Decision Trees (DT), Naive Bayes (NB), and Projective Adaptive Resonance Theory (PART)—were assessed for their ability to predict hospital mortality. Despite missing values, the EMPICU-RF model showed good mortality prediction within the first six hours after ICU admission, suggesting a viable method for early risk assessment in critical care settings[13].

Meyar.H,et.al described Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. Using time-series data, this study examined whether machine learning models could enhance the accuracy of the 90-day mortality prediction for ICU patients in real-time. A recurrent neural network was trained using SAPS III variables and physiological time-series data with hourly resolution using data from four ICUs in Denmark. The model was verified both internally and externally, showed good predictive performance, and used a Shapley additive explanations technique to quantify each feature's influence on predictions, making it easier to understand. In ICU conditions, this dynamic approach provided improved prognostication and interpretability, outperforming conventional static models[14].

Alghatani.K,et.al proposed Predicting Intensive Care Unit Length of Stay and Mortality Using Patient Vital Signs: Machine Learning Model Development and ValidationIn order to create prediction models for ICU patient outcomes, including death and length of stay, this study made use of the MIMIC database. Regression techniques were utilised to predict the precise number of days spent in the intensive care unit (ICU), while six machine learning algorithms were applied for the binary classification of discharge status and length of stay. Baseline demographic and vital sign characteristics, together with an improved quantiles technique with engineering variables, were used to build the models. With an accuracy of almost 89% for death prediction and 65% for duration of stay based on the median ICU stay, the random forest method proved to be the most accurate[15].

2.3 Gap Analysis

Aspect	Current Methods	Identified Gaps	Future Directions
Data Utilization	Basic demographic and vital-sign and lab data	Limited incorporation of genetic, comorbidity and lifestyle information	Integrate richer data sources (medical history, genomics, vitals over time)
Model Complexity	Decision trees, logistic regression	Unable to capture highly nonlinear or temporal patterns	Adopt deep learning (RNN/CNN) and advanced feature engineering
Interpretability	Random Forest, SVM often act as "black boxes"	Loss of clinical trust due to opaque decision logic	Apply explainable AI methods (LIME, SHAP) to clarify model outputs
Real-Time Decision Making	Offline batch analysis of historical records	No continuous, real-time alerts for clinicians	Develop models that run on live ICU streams for immediate intervention
Generalization	Models trained on single-center data	Poor performance when applied elsewhere	Train and validate on multi-center, diverse patient cohorts
Evaluation Metrics	Accuracy, Precision, Recall, AUC	Insensitive to class imbalance and rare events	Use balanced metrics (F1-score, specificity, calibration) and cost-sensitive learning
Ensemble Methods	Random Forest, XGBoost with fixed architectures	Fixed ensembling may underutilize model diversity	Explore hybrid stacking and dynamic ensemble selection for robustness

Table 2.1: Gap Analysis

2.4 Summary

People with critical and serious diseases that must be watched constantly receive treatment in the ICU. Even with new medical equipment, many patients in ICUs die because their condition was not spotted on time or recognized when needed. As healthcare data grows more accessible, it becomes urgent for smart systems to help with clinical choices.

One can analyze and forecast outcomes through efficient methods from machine learning (ML) and ensemble learning. With the help of data from the past, including test outcomes, symptoms and personal details, these approaches can determine who is most at risk. Although combining several models with random forest, AdaBoost and XGBoost increases accuracy, logistic regression, decision trees and K-nearest neighbors have shown potential in handling the risk categorization.

The combination of these techniques helps estimate ICU patients' death risk more accurately, so medical staff can act fast when needed. With predictive analytics, the plan is to increase success in ICU treatment by acting early and using appropriate resources.

Chapter 3

Research Methodology

You will learn the step-by-step approach for formulating ICU mortality predictions using both machine learning and ensemble techniques here. To ensure that predictions are correct, it states how data should be collected, prepared, models built and how well they work.

3.1 Methodology/Requirement Analysis & Design Specification

3.1.1 Overview

Using machine learning and ensemble techniques, the research uses a reliable process to examine and predict the risk of death for those in intensive care. Initially, clinicians and researchers collect a set of data that contains laboratory findings, physical exam measurements and key details about patients in intensive care.

Subsequently, the data process is carried out, where category variables are encoded, missing values are addressed and numerical numbers are normalized. Special techniques identify which features are most important for shaping the outcome of patients.

In the end, Decision Trees, KNN and Logistic Regression are employed to separate patients into each person's risk category. Moreover, using the benefits of several models, organizations depend on ensemble methods such as Random Forest, AdaBoost and XGBoost to improve the accuracy of their results. Models are tested using accuracy, precision, recall, F1-score and ROC-AUC scores. Besides, cross-validation checks whether the models will continue to work when used in different situations.

The ideal method for predicting ICU mortality is the one that achieves the highest accuracy and best results on every measure.

3.1.2 Proposed Methodology/ System Design

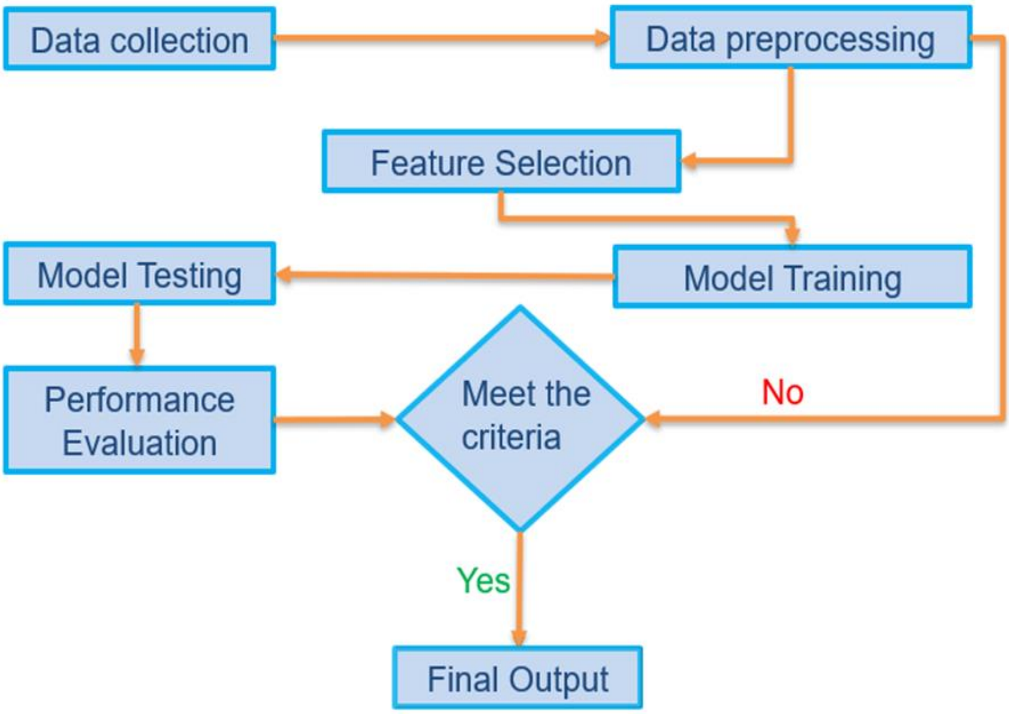
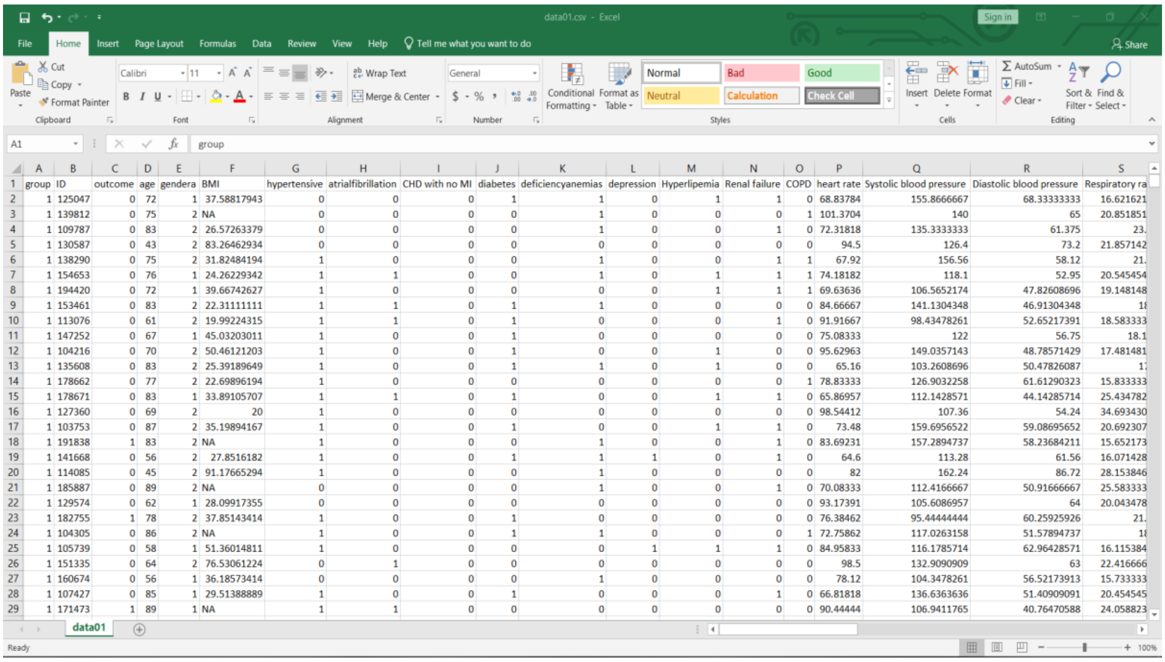
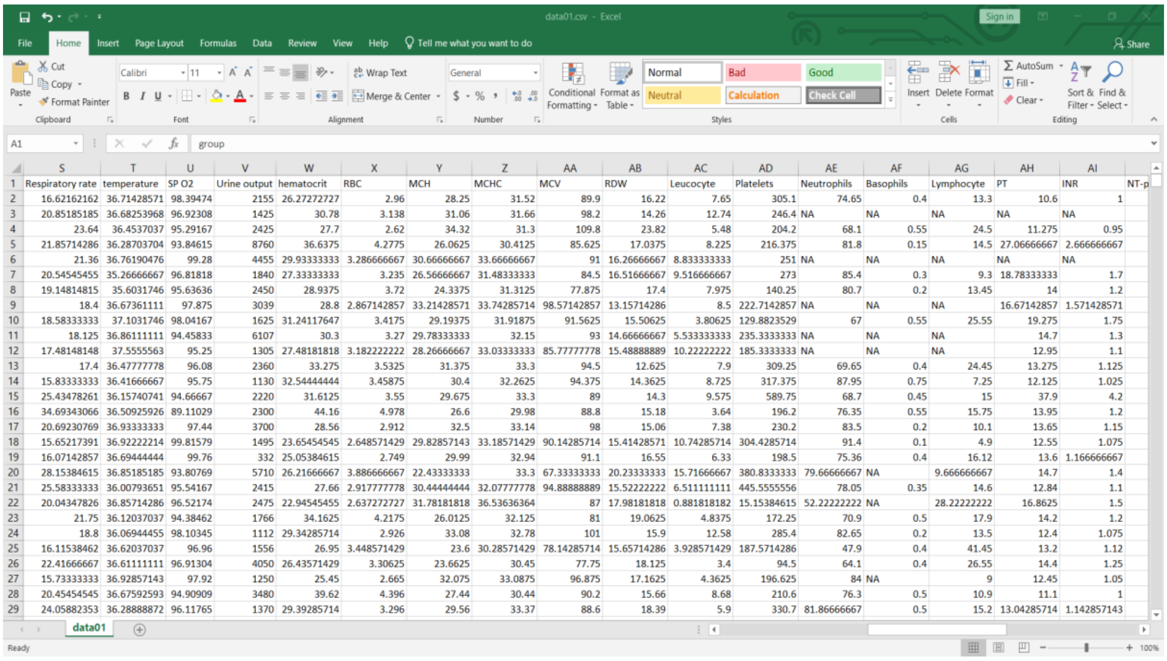


Figure 3.1: Methodology Diagram

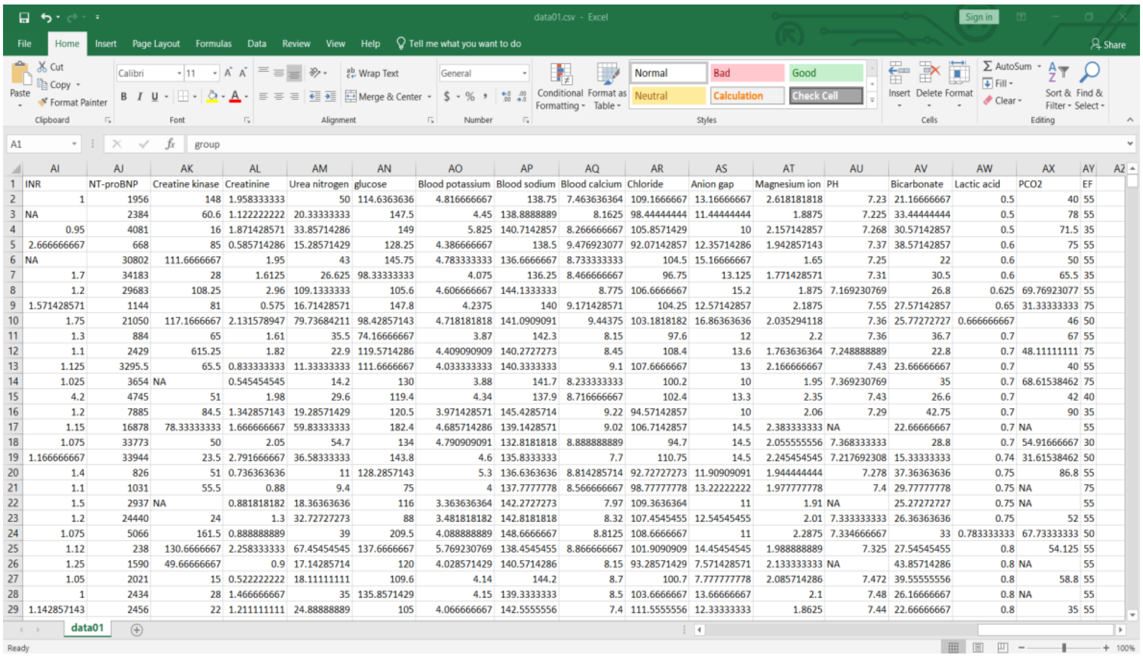
3.1.3 Data Collection



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	group	ID	outcome	age	gendera	BMI	hypertensive	atrialfibrillation	CHD with no MI	diabetes	deficiencyanemias	depression	Hyperlipemia	Renal failure	COPD	heart rate	Systolic blood pressure	Diastolic blood pressure	Respiratory ra
2	1	125047	0	72	1	37.58817943	0	0	0	1	1	0	1	1	0	68.83784	155.8666667	68.33333333	16.621621
3	1	139812	0	75	2	NA	0	0	0	0	1	0	0	0	1	101.3704	140	65	20.851851
4	1	109787	0	83	2	26.57263379	0	0	0	0	1	0	0	1	0	72.31818	135.3333333	61.375	23.
5	1	130587	0	43	2	83.26462934	0	0	0	0	0	0	0	0	0	94.5	126.4	73.2	21.857142
6	1	138290	0	75	2	31.82484194	1	0	0	0	1	0	0	1	1	67.92	156.56	58.12	21.
7	1	154653	0	76	1	24.26229342	1	1	0	0	1	0	1	1	1	74.18182	118.1	52.95	20.545454
8	1	194420	0	72	1	39.66742627	1	0	0	0	0	0	1	1	1	69.63636	106.5652174	47.82608696	19.148148
9	1	153461	0	83	2	22.31111111	1	1	0	1	1	0	0	0	0	84.66667	141.1304348	46.91304348	18
10	1	113076	0	61	2	19.99224315	1	1	0	1	0	0	0	1	0	91.91667	98.43478261	52.65217391	18.583333
11	1	147252	0	67	1	45.03203011	1	0	0	1	0	0	0	0	0	75.08333	122	56.75	18.1
12	1	104216	0	70	2	50.46121203	1	0	0	1	0	0	1	0	0	95.62963	149.0357143	48.78571429	17.481481
13	1	135608	0	83	2	25.39189649	1	0	0	1	1	0	1	0	0	65.16	103.2608696	50.47826087	17
14	1	178662	0	77	2	22.69896194	1	0	0	0	0	0	0	0	1	78.83333	126.9032258	61.61290323	15.833333
15	1	178671	0	83	1	33.89105707	1	1	0	1	0	0	1	1	0	65.86957	112.1428571	44.14285714	25.434782
16	1	127360	0	69	2	20	1	0	0	0	0	0	0	0	0	98.54412	107.36	54.24	34.693430
17	1	103753	0	87	2	35.19894167	1	0	0	1	0	0	1	1	0	73.48	159.6956522	59.08695652	20.692307
18	1	191838	1	83	2	NA	1	0	0	0	1	0	0	1	0	83.69231	157.2894737	58.23684211	15.652173
19	1	141668	0	56	2	27.8516182	1	0	0	1	1	1	0	1	0	64.6	113.28	61.56	16.071428
20	1	114085	0	45	2	91.17665294	1	0	0	0	1	0	0	0	0	82	162.24	86.72	28.153846
21	1	185887	0	89	2	NA	0	0	0	0	1	0	0	1	0	70.08333	112.4166667	50.91666667	25.583333
22	1	129574	0	62	1	28.09917355	0	0	0	0	0	0	0	0	0	93.17391	105.6086957	64	20.043478
23	1	182755	1	78	2	37.85143414	1	0	0	1	0	0	0	0	0	76.38462	95.44444444	60.25925926	21.
24	1	104305	0	86	2	NA	1	0	0	1	1	0	0	0	1	72.75862	117.0263158	51.57894737	18
25	1	105739	0	58	1	51.36014811	1	0	0	0	0	1	1	1	0	84.95833	116.1785714	62.96428571	16.115384
26	1	151335	0	64	2	76.53061224	0	1	0	0	0	0	0	0	0	98.5	132.9090909	63	22.416666
27	1	160674	0	56	1	36.18573414	0	0	0	0	1	0	0	0	0	78.12	104.3478261	56.52173913	15.733333
28	1	107427	0	85	1	29.51388889	1	0	0	1	0	0	0	1	0	66.81818	136.6363636	51.40909091	20.454545
29	1	171473	1	89	1	NA	1	1	0	0	0	0	0	0	0	90.44444	106.9411765	40.76470588	24.058823



	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	
1	Respiratory rate	temperature	SP O2	Urine output	hematocrit	RBC	MCH	MCHC	MCV	RDW	Leucocyte	Platelets	Neutrophils	Basophils	Lymphocyte	PT	INR	NT-p
2	16.62162162	36.71428571	98.39474	2155	26.27272727	2.96	28.25	31.52	89.9	16.22	7.65	305.1	74.65	0.4	13.3	10.6	1	
3	20.85185185	36.68253968	96.92308	1425	30.78	3.138	31.06	31.66	98.2	14.26	12.74	246.4	NA	NA	NA	NA	NA	
4	23.64	36.4537037	95.29167	2425	27.7	2.62	34.32	31.3	109.8	23.82	5.48	204.2	68.1	0.55	24.5	11.275	0.95	
5	21.85714286	36.28703704	93.84615	8760	36.6375	4.2775	26.0625	30.4125	85.625	17.0375	8.225	216.375	81.8	0.15	14.5	27.06666667	2.666666667	
6	21.36	36.76190476	99.28	4455	29.93333333	3.286666667	30.66666667	33.66666667	91	16.26666667	8.833333333	251	NA	NA	NA	NA	NA	
7	20.54545455	35.26666667	96.81818	1840	27.33333333	3.235	26.56666667	31.48333333	84.5	16.51666667	9.516666667	273	85.4	0.3	9.3	18.78333333	1.7	
8	19.14814815	35.6031746	95.63636	2450	28.9375	3.72	24.3375	31.3125	77.875	17.4	7.975	140.25	80.7	0.2	13.45	14	1.2	
9	18.4	36.67361111	97.875	3039	28.8	2.867142857	33.21428571	33.74285714	98.57142857	13.15714286	8.5	222.7142857	NA	NA	NA	16.67142857	1.571428571	
10	18.58333333	37.1031746	98.04167	1625	31.24117647	3.4175	29.19375	31.91875	91.5625	15.50625	3.80625	129.8823529	67	0.55	25.55	19.275	1.75	
11	18.125	36.86111111	94.45833	6107	30.3	3.27	29.78333333	32.15	93	14.66666667	5.533333333	235.3333333	NA	NA	NA	14.7	1.3	
12	17.48148148	37.5555563	95.25	1305	27.48181818	3.182222222	28.26666667	33.03333333	85.77777778	15.48888889	10.22222222	185.3333333	NA	NA	NA	12.95	1.1	
13	17.4	36.47777778	96.08	2360	33.275	3.5325	31.375	33.3	94.5	12.625	7.9	309.25	69.65	0.4	24.45	13.275	1.125	
14	15.83333333	36.41666667	95.75	1130	32.54444444	3.45875	30.4	32.2625	94.375	14.3625	8.725	317.375	87.95	0.75	7.25	12.125	1.025	
15	25.43478261	36.15740741	94.66667	2220	31.6125	3.55	29.675	33.3	89	14.3	9.575	589.75	68.7	0.45	15	37.9	4.2	
16	34.69343066	36.50925926	89.11029	2300	44.16	4.978	26.6	29.98	88.8	15.18	3.64	196.2	76.35	0.55	15.75	13.95	1.2	
17	20.69230769	36.93333333	97.44	3700	28.56	2.912	32.5	33.14	98	15.06	7.38	230.2	83.5	0.2	10.1	13.65	1.15	
18	15.65217391	36.92222214	99.81579	1495	23.65454545	2.648571429	29.82857143	33.18571429	90.14285714	15.41428571	10.74285714	304.4285714	91.4	0.1	4.9	12.55	1.075	
19	16.07142857	36.69444444	99.76	332	25.05384615	2.749	29.99	32.94	91.1	16.55	6.33	198.5	75.36	0.4	16.12	13.6	1.166666667	
20	28.15384615	36.85185185	93.80769	5710	26.21666667	3.886666667	22.43333333	33.3	67.33333333	20.23333333	15.71666667	380.8333333	79.66666667	NA	9.666666667	14.7	1.4	
21	25.58333333	36.00793651	95.54167	2415	27.66	2.917777778	30.44444444	32.07777778	94.88888889	15.52222222	6.511111111	445.5555556	78.05	0.35	14.6	12.84	1.1	
22	20.04347826	36.85714286	96.52174	2475	22.94545455	2.637272727	31.78181818	36.53636364	87	17.98181818	0.881818182	15.15384615	52.22222222	NA	28.22222222	16.8625	1.5	
23	21.75	36.12037037	94.38462	1766	34.1625	4.2175	26.0125	32.125	81	19.0625	4.8375	172.25	70.9	0.5	17.9	14.2	1.2	
24	18.8	36.06944455	98.10345	1112	29.34285714	2.926	33.08	32.78	101	15.9	12.58	285.4	82.65	0.2	13.5	12.4	1.075	
25	16.11538462	36.62037037	96.96	1556	26.95	3.448571429	23.6	30.28571429	78.14285714	15.65714286	3.928571429	187.5714286	47.9	0.4	41.45	13.2	1.12	
26	22.41666667	36.61111111	96.91304	4050	26.43571429	3.30625	23.6625	30.45	77.75	18.125	3.4	94.5	64.1	0.4	26.55	14.4	1.25	
27	15.73333333	36.92857143	97.92	1250	25.45	2.665	32.075	33.0875	96.875	17.1625	4.3625	196.625	84	NA	9	12.45	1.05	
28	20.45454545	36.67592593	94.90909	3480	39.62	4.396	27.44	30.44	90.2	15.66	8.68	210.6	76.3	0.5	10.9	11.1	1	
29	24.05882353	36.28888872	96.11765	1370	29.39285714	3.296	29.56	33.37	88.6	18.39	5.9	330.7	81.86666667	0.5	15.2	13.04285714	1.142857143	



	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY
1	INR	NT-proBNP	Creatine kinase	Creatinine	Urea nitrogen	glucose	Blood potassium	Blood sodium	Blood calcium	Chloride	Anion gap	Magnesium ion	PH	Bicarbonate	Lactic acid	PCO2	EF
2	1	1956	148	1.958333333	50	114.6363636	4.816666667	138.75	7.463636364	109.1666667	13.16666667	2.618181818	7.23	21.16666667	0.5	40	55
3	NA	2384	60.6	1.122222222	20.33333333	147.5	4.45	138.8888889	8.1625	98.44444444	11.44444444	1.8875	7.225	33.44444444	0.5	78	55
4	0.95	4081	16	1.871428571	33.85714286	149	5.825	140.7142857	8.266666667	105.8571429	10	2.157142857	7.268	30.57142857	0.5	71.5	35
5	2.666666667	668	85	0.585714286	15.28571429	128.25	4.386666667	138.5	9.476923077	92.07142857	12.35714286	1.942857143	7.37	38.57142857	0.6	75	55
6	NA	30802	111.6666667	1.95	43	145.75	4.783333333	136.6666667	8.733333333	104.5	15.16666667	1.65	7.25	22	0.6	50	55
7	1.7	34183	28	1.6125	26.625	98.33333333	4.075	136.25	8.466666667	96.75	13.125	1.771428571	7.31	30.5	0.6	65.5	35
8	1.2	29683	108.25	2.96	109.1333333	105.6	4.606666667	144.1333333	8.775	106.6666667	15.2	1.875	7.169230769	26.8	0.625	69.76923077	55
9	1.571428571	1144	81	0.575	16.71428571	147.8	4.2375	140	9.171428571	104.25	12.57142857	2.1875	7.55	27.57142857	0.65	31.33333333	75
10	1.75	21050	117.1666667	2.131578947	79.73684211	98.42857143	4.718181818	141.0909091	9.44375	103.1818182	16.86363636	2.035294118	7.36	25.77272727	0.666666667	46	50
11	1.3	884	65	1.61	35.5	74.16666667	3.87	142.3	8.15	97.6	12	2.2	7.36	36.7	0.7	67	55
12	1.1	2429	615.25	1.82	22.9	119.5714286	4.409090909	140.2727273	8.45	108.4	13.6	1.763636364	7.248888889	22.8	0.7	48.11111111	75
13	1.125	3295.5	65.5	0.833333333	11.33333333	111.6666667	4.033333333	140.3333333	9.1	107.6666667	13	2.166666667	7.43	23.66666667	0.7	40	55
14	1.025	3654	NA	0.545454545	14.2	130	3.88	141.7	8.233333333	100.2	10	1.95	7.369230769	35	0.7	68.61538462	75
15	4.2	4745	51	1.98	29.6	119.4	4.34	137.9	8.716666667	102.4	13.3	2.35	7.43	26.6	0.7	42	40
16	1.2	7885	84.5	1.342857143	19.28571429	120.5	3.971428571	145.4285714	9.22	94.57142857	10	2.06	7.29	42.75	0.7	90	35
17	1.15	16878	78.33333333	1.666666667	59.83333333	182.4	4.685714286	139.1428571	9.02	106.7142857	14.5	2.383333333	NA	22.66666667	0.7	NA	55
18	1.075	33773	50	2.05	54.7	134	4.790909091	132.8181818	8.888888889	94.7	14.5	2.055555556	7.368333333	28.8	0.7	54.91666667	30
19	1.166666667	33944	23.5	2.791666667	36.58333333	143.8	4.6	135.8333333	7.7	110.75	14.5	2.245454545	7.217692308	15.33333333	0.74	31.61538462	50
20	1.4	826	51	0.736363636	11	128.2857143	5.3	136.6363636	8.814285714	92.72727273	11.90909091	1.944444444	7.278	37.36363636	0.75	86.8	55
21	1.1	1031	55.5	0.88	9.4	75	4	137.7777778	8.566666667	98.77777778	13.22222222	1.977777778	7.4	29.77777778	0.75	NA	75
22	1.5	2937	NA	0.881818182	18.36363636	116	3.363636364	142.2727273	7.97	109.3636364	11	1.91	NA	25.27272727	0.75	NA	55
23	1.2	24440	24	1.3	32.72727273	88	3.481818182	142.8181818	8.32	107.4545455	12.54545455	2.01	7.333333333	26.36363636	0.75	52	55
24	1.075	5066	161.5	0.888888889	39	209.5	4.088888889	148.6666667	8.8125	108.6666667	11	2.2875	7.334666667	33	0.783333333	67.73333333	50
25	1.12	238	130.6666667	2.258333333	67.45454545	137.6666667	5.769230769	138.4545455	8.866666667	101.9090909	14.45454545	1.988888889	7.325	27.54545455	0.8	54.125	55
26	1.25	1590	49.66666667	0.9	17.14285714	120	4.028571429	140.5714286	8.15	93.28571429	7.571428571	2.133333333	NA	43.85714286	0.8	NA	55
27	1.05	2021	15	0.522222222	18.11111111	109.6	4.14	144.2	8.7	100.7	7.777777778	2.085714286	7.472	39.55555556	0.8	58.8	55
28	1	2434	28	1.466666667	35	135.8571429	4.15	139.3333333	8.5	103.6666667	13.66666667	2.1	7.48	26.16666667	0.8	NA	55
29	1.142857143	2456	22	1.211111111	24.88888889	105	4.066666667	142.5555556	7.4	111.5555556	12.33333333	1.8625	7.44	22.66666667	0.8	35	55

Figure 3.2: Sample Data

The dataset, which is called In-hospital Mortality Rate Prediction, was gathered via Kaggle. Zhou, Jingmin, and colleagues (2021) are the dataset's original authors. There are 51 characteristics and 1177 thousand unique records in this data set. All the information needed to forecast the patient death rate is included in this dataset. The dataset is licensed under the CC0 1.0 Universal (CC0 1.0) Public Domain Dedication license.

3.1.4 Data Preprocessing

One method for transforming raw data into clean data is data preparation. Data preprocessing is utilized here to transform the messy data into clean data so that greater insights can be gleaned from the data. We can say that the data is unstructured when it is collected and contains a variety of noise types. To get better results, we train the machine learning model on this clean, or perhaps more accurately, appropriate data. We must sanitize the data before we can feed it. The largest obstacle was cleaning the data because it contains a large number of null values. We employ the mode for categorical features and the mean value for continuous features to deal with those null values. An outlier is a result in a random sample drawn from the population that deviates abnormally from the norm. This formulation, in a sense, leaves it up to the analyst to decide what is abnormal. The first step in identifying aberrant observations is to characterize regular ones. There are extremely few data under 30 and above 90 that we deemed to be outliers, even though the age range in the age columns normal distribution is 60 to 90. "ZScore" helped us deal with this oddity. One technique for effectively managing outliers is ZScore. The amount of standard deviations separating a particular data point from the mean is measured by the Z-Score. Data points whose Z-Score exceeds a predetermined threshold are sometimes referred to as outliers. We must first compute the standard deviation in order to determine the ZScore. The standard deviation calculation formula is as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}\left(X_i - \mu\right)^2}{N}}$$

Here,

N is the number of data points,
μ is the dataset mean,
σ is the standard deviation,
and
● is an individual data point.
Finally, after determining the standard deviation, we can use the following formula to determine the ZScore:

$$Z = \frac{X - \mu}{\sigma}$$

Z = ZScore,

X = Raw score that I want to convert,

μ = Population mean,

σ = Standard deviation.

After handling outliers there is another problem arises the data is spreading very high. We used a typical scaler to scale the data because some of its values are modest, while others are very large. And then we saw that data are highly correlated and we used PCA here to handle this problem. Before using PCA the data looked like this,

Correlation matrix. A graph that displays the correlation coefficients between different variables in a dataset is called a correlation matrix. Every row and column in the matrix represents a variable, and the rows and columns of cells in the matrix display the correlation between the parameters that cross each other.

How strongly and in which direction the two variables are related is indicated by the correlation coefficient. It falls within the -1 to 1 range. When one variable rises, the other rises proportionately, indicating a perfect positive linear relationship with a correlation of 1. A correlation of -1 indicates a perfect negative linear relationship, meaning that when one measure grows, the other declines accordingly. If the correlation is 0, there isn't an exponential relationship between the variables.

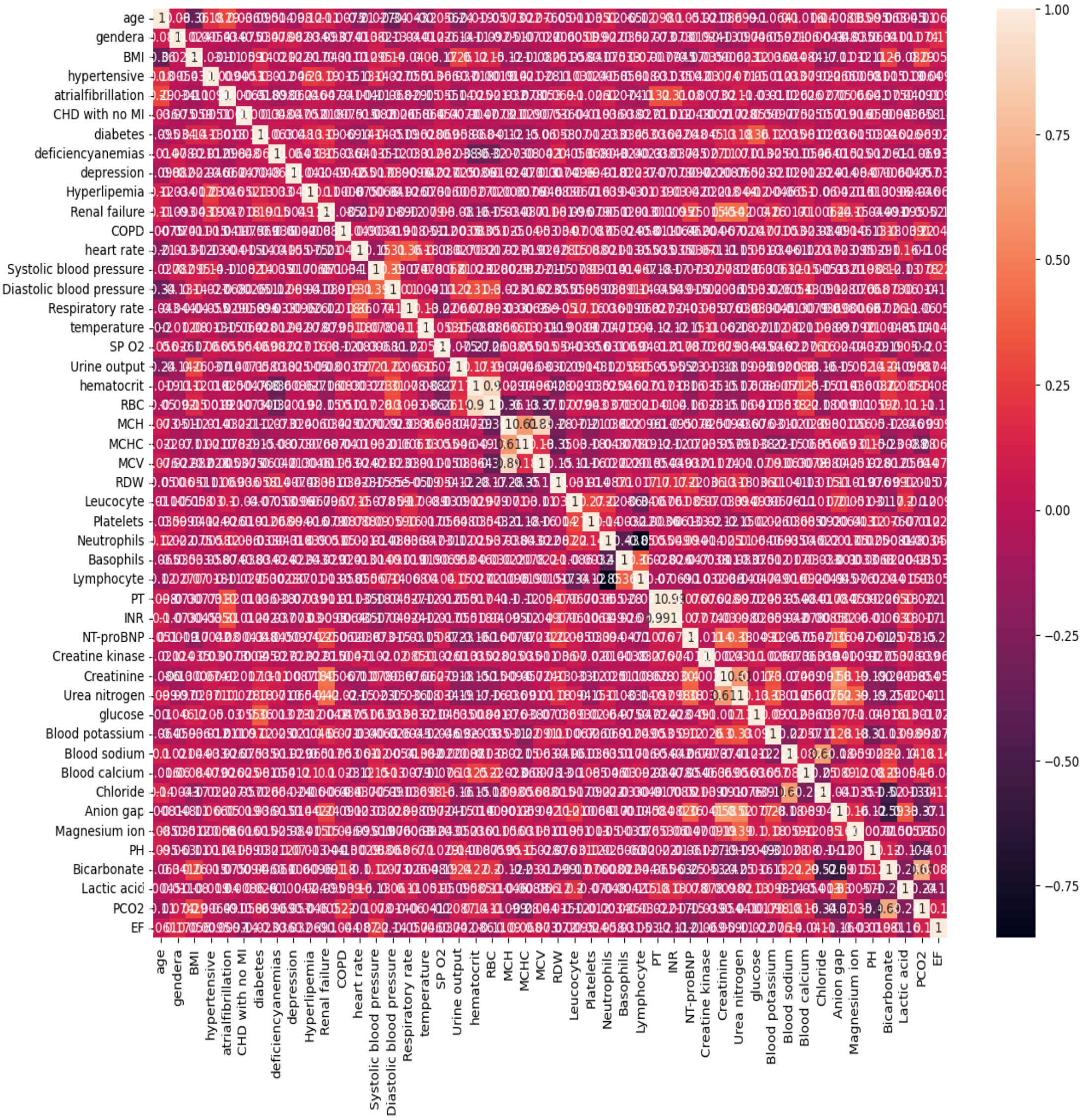


Figure 3.3: Correlation

This is how PCA correlation appears after use: A machine learning dimensionality reduction method called Principal Component Analysis (PCA) makes datasets simpler while preserving a significant amount of variance. A new set of uncorrelated variables called principal components is created from the original variables, and they are sorted based on how much of the original variation they can explain. The majority of the dataset's information is retained by the first few principal components, which lowers the number of features required for analysis. Machine learning models perform better when PCA is used to visualize high-dimensional data, lower computational costs, and mitigate multicollinearity.

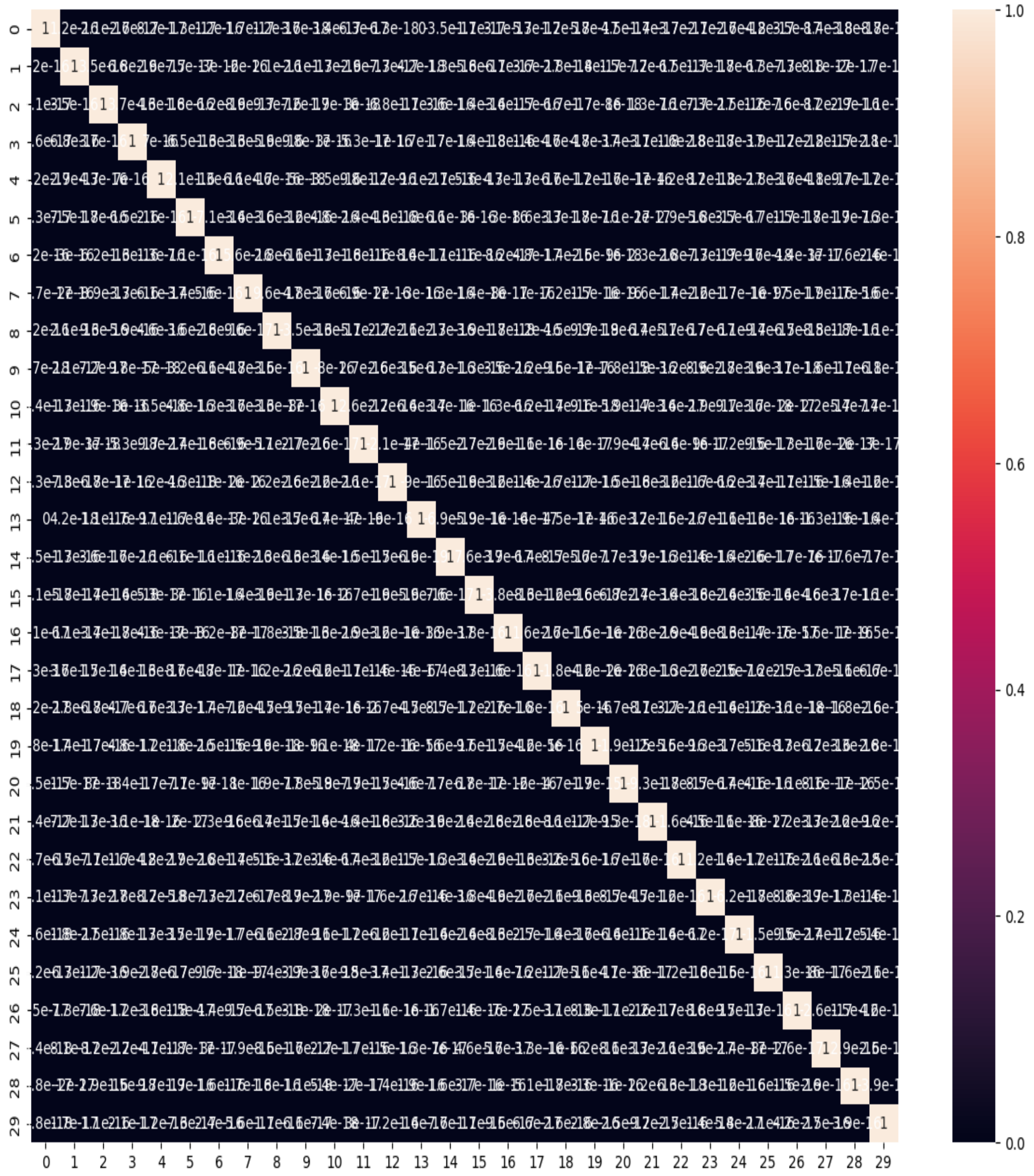


Figure 3.4: Correlation After PCA

Feature selection: The process of choosing a particular set of variables, attributes, or noteworthy and pertinent features from a larger pool of available features is known as feature selection in the context of machine learning. Avoiding overfitting, increasing computing effectiveness, and occasionally improving interpretability are the goals in improving the model's performance. Having irrelevant features could be a cause for the model's bad performance we called it the curse of dimensionality. Too many features can lead the model towards overfitting which is very big problem. So, feature selection is more important before we feed the data into the model. But we cannot choose the feature randomly we need to go through some procedure and then we have to calculate feature importance, after calculating feature importance we can easily understand which feature is important and which one is not. In this study, I calculate the feature importance based on P-value and I use here XGBoost, F-ClassIf, Logistic Regression, and Gradient boosting.

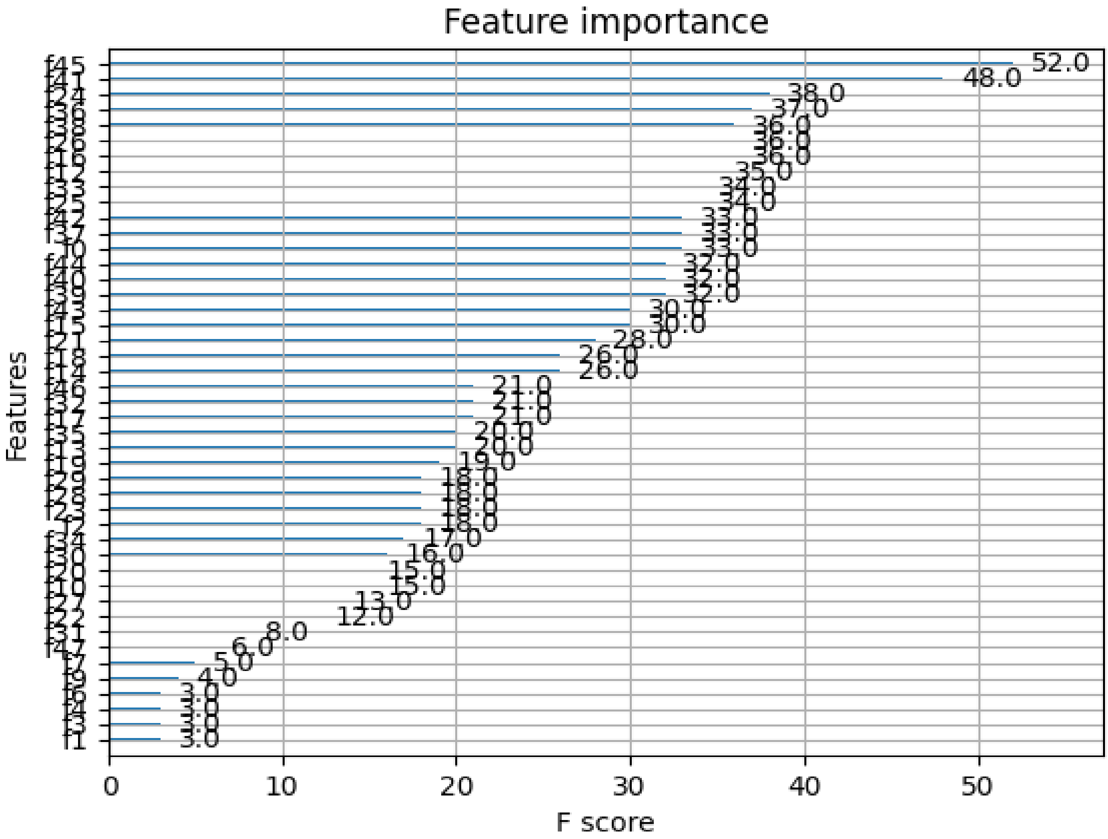


Figure 3.5: Feature Importance

Here, f45 carries the highest important for our model and f41 is 2nd and f24 is third highest important feature. The lowest important feature is f1. We calculate this importance using xgboost.

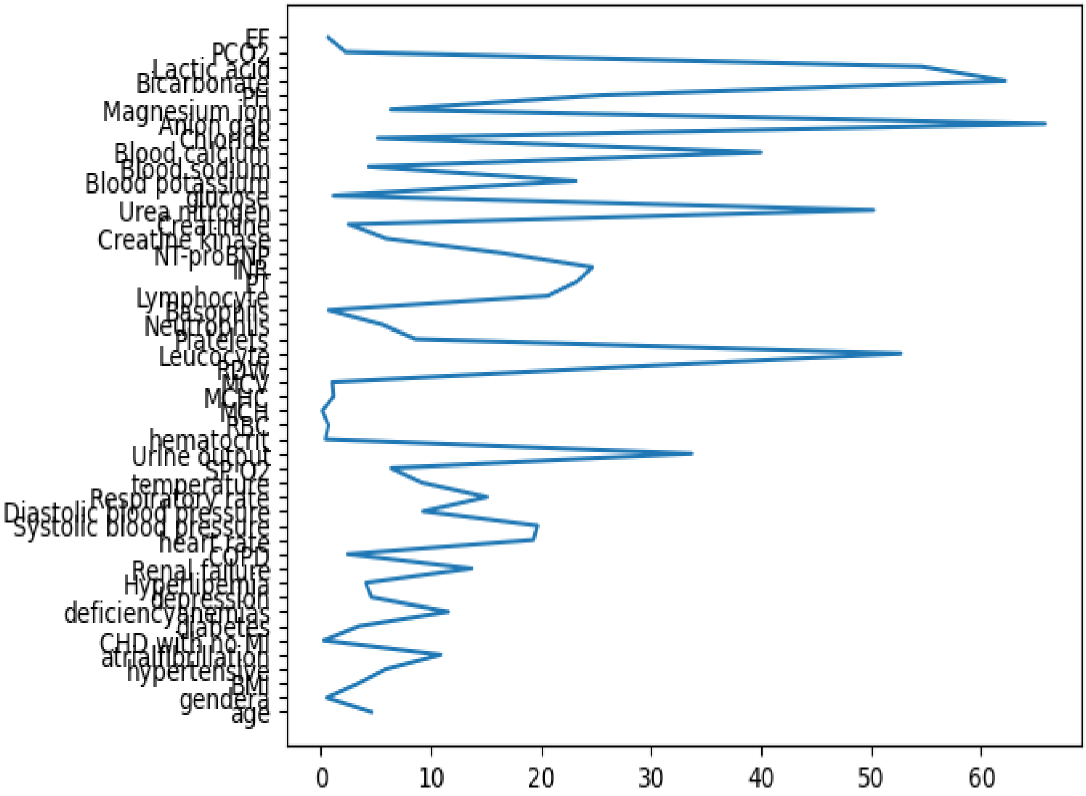


Figure 3.6: Feature Importance

We calculate this importance before training the model using F-classif. After doing all of this we feed the data into the model.

Decision Tree

A decision tree is a flexible supervised learning technique that creates a tree-like model to assist in making decisions based on input data. Each node represents a feature or an option, and each edge represents the outcome of a decision. This division of the feature space is done recursively. By selecting the best feature to divide the data at each node—typically based on metrics like entropy or Gini impurity—it aims to optimize the homogeneity of the target variable within the resulting partitions. Decision trees can be used for classification, where the leaf nodes store class labels, or regression, where the leaf nodes hold continuous data. Although they are easily interpreted and visualized, they are prone to overfitting, which can be lessened by using pruning strategies. Even while decision trees provide transparent and easy-to-understand decision-making processes, they can be biased when faced with unbalanced data, therefore ensemble techniques like Random Forests can help them perform better and be more robust.

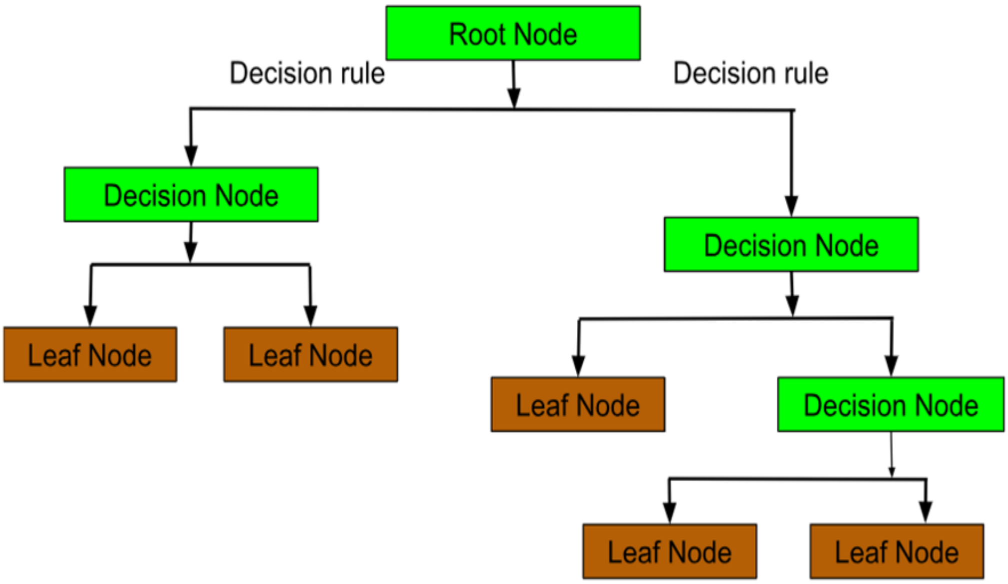


Figure 3.7: Decision Tree

Random Forest

Thanks to randomization, each tree in the ensemble is related differently which brings down the risk of overfitting and achieves better generalizing. Bootstrapping is used to build each tree in the forest, so random parts of the data are sampled and reused to form the tree branch. Each node looks at only a random set of attributes to split, in order to keep making the trees more diverse. Combining the forecasts of individual trees is how Random Forest arrives at a prediction. For regression problems, it selects the average prediction; for classification, it looks at which tree most people agreed with. The fact that Random Forests can scale, remain resilient and process high dimensional data makes them popular. Machine learning experts choose these methods often as they are less likely to overfit than an individual Decision Tree. They help users compare and decide which features are most significant by giving them a “feature importance” evaluation. On the whole, Random Forests tradeoff between being complex, informative and effective at.

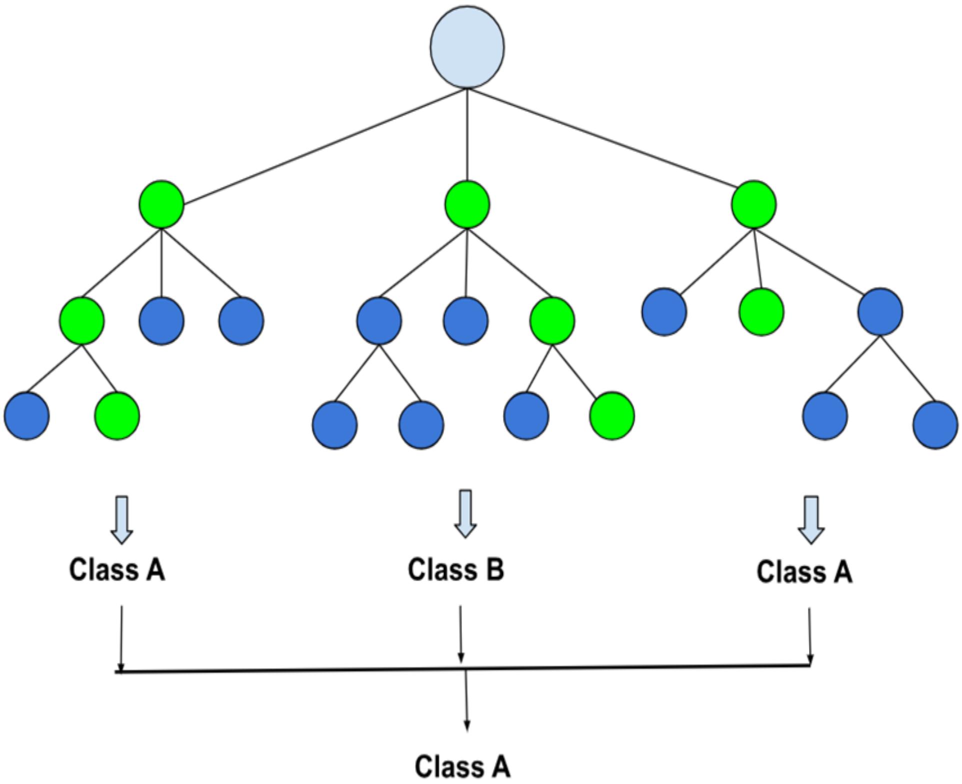


Figure 3.8: Random Forrest

Logistic Regression

Machine learning issues related to binary categorization often depend on logistic regression in statistics. The name may be confusing, but it’s actually more like a classification algorithm than a regression method. It estimates the chance an instance belongs to a class based on one or more variables used to predict. Logistic regression assigns a probability to an example using a logistic function (the sigmoid function). Through this particular Generalized Linear Model (GLM), the level of confidence in the probability of being assigned to the positive category is reached by transforming the output from a linear combination of the input features into a value between 0 and 1. The method Logistic Regression uses to define its feature weights is optimization, using gradient descent to reduce a typical cost function which can be the negative log-likelihood or cross-entropy loss. The idea is to make the observed information as probable as possible using the estimated parameters. Logistic regression assumes that there is an approximately linear connection between the log of the chances of the result and the input measures. Even though it is simple, logistic regression clearly shows the effect that each feature has on the chance of an event when the data is split in straight segments.

While logistic regression was first created for binary classification, you can adapt it for multi-class by either using one-vs-rest methods or multinomial logistic regression. In general, logistic regression is preferred because it is straightforward, easy to understand and useful for a wide variety of real problems. In cases where the data is highly uneven,questaion, these more sophisticated models may function better.

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

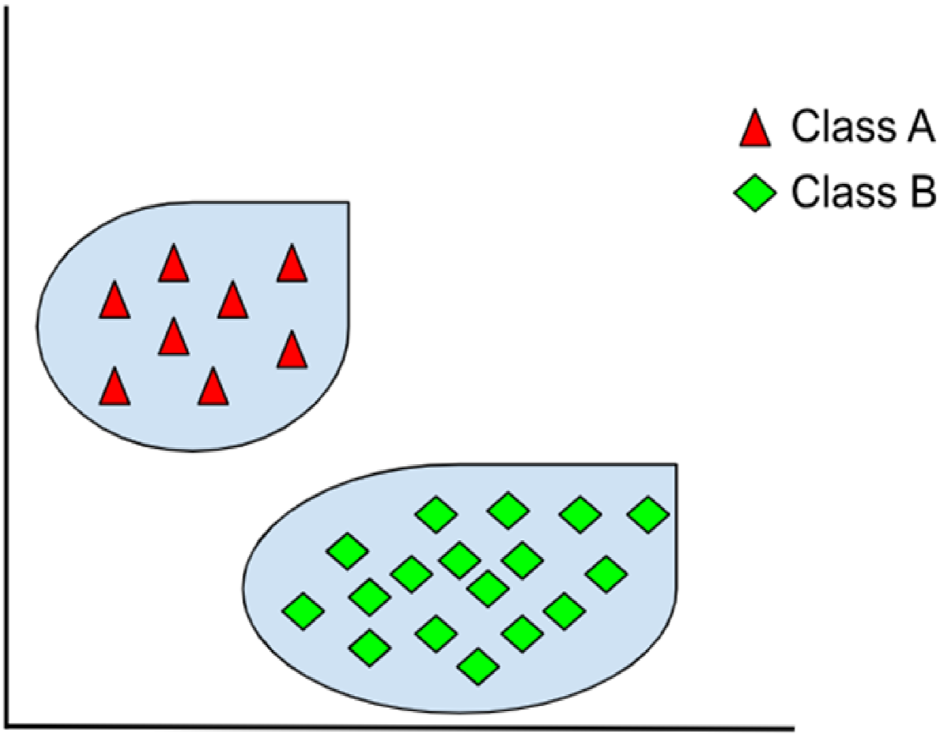
Where,

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

KNN

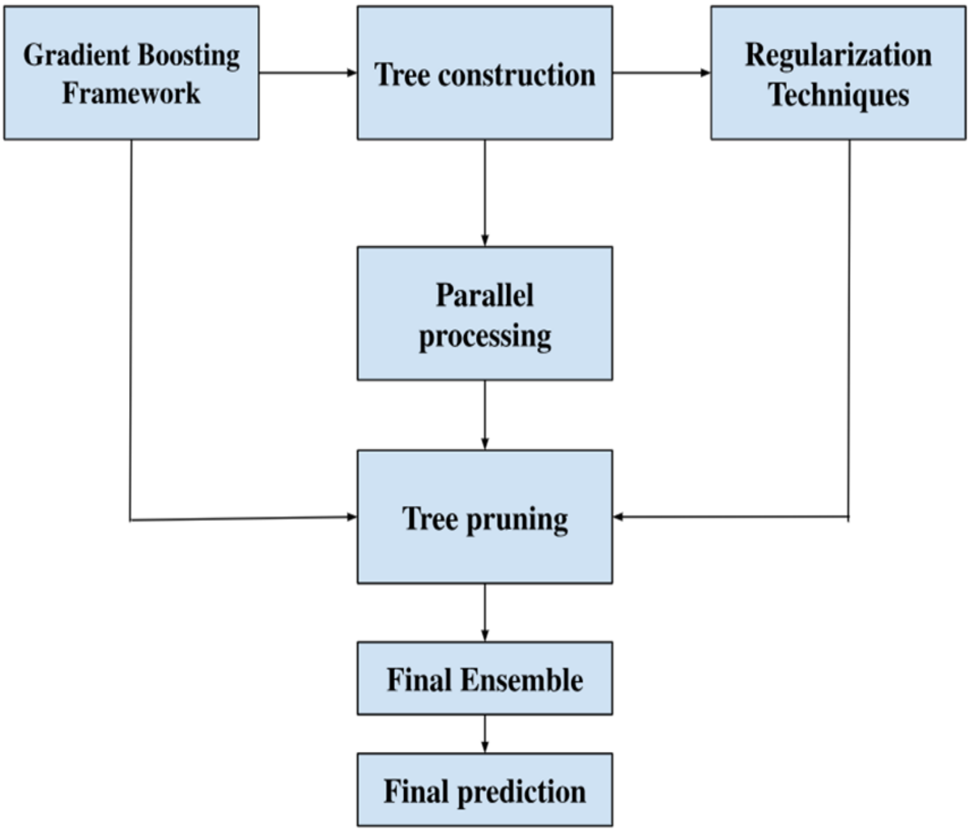
KNN makes its prediction by looking for the k points that are most similar to the data point it is analyzing. Usually, the nearest neighbors are determined using the Euclidean distance for continuous features, even though Manhattan distance and cosine similarity are also sometimes used with different types of data. Representation of feature vectors and labels of the training instances is conducted by KNN just while training. Once given a new instance, it computes how far each new instance is from each training instance. To make its prediction, it takes the k nearest instances and adds their labels (classification) or values (regression) together. In KNN, you should pick the hyperparameter k which tells you how many neighbors to look at. A high k often gives us biased and underfitting results, whereas a lower k can cause models to fit only the training data and overfit. Many times, teams use cross-validation or adjust their hyperparameters to determine the optimal k value. Being non-parametric means KNN does not need to know anything about the chances of data in the dataset. Since it is easy to construct and explain, it is regularly employed as a basic model and as a reference for advanced algorithms. Yet, it may run slowly with large datasets, due to needing to calculate and keep track of the distances from every training point during prediction. The formula of Euclidian distance is:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} \left(y_i - x_i\right)^2}$$



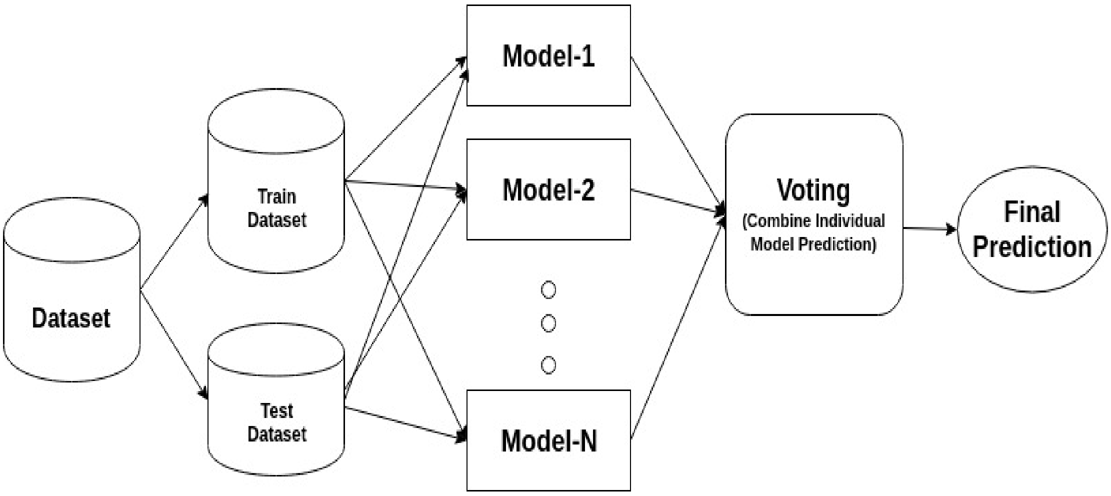
XGBOOST

XGBoost produces an additive ensemble of weak learners, usually decision trees, with each tree explanatory learning from the mistakes of its predecessors. It minimizes a differentiable loss function by optimization, using gradient descent or a variation like Newton's method. By "gradient boosting" we refer to the optimization process. The trees are built sequentially, and each tree is built to compensate for the errors of the previous tree. This is accomplished by fitting each successive tree to the residual errors of the previous ensemble, which is the discrepancy between the current ensemble's forecast and the target value. XGBoost contains regularization terms to help with model generalization and the potential issues with overfitting. Regularization terms penalize complexity by adding in deviations and restrictions of tree depth, weights of leaves per tree, and leaf count. XGBoost is also a great resource to help guide users to determine feature importance, and determining which features significantly affect the prediction performance of the model, which is useful for feature selections as well as useful in understanding patterns in the data. XGBoost also provides exceptional optimization techniques for both speed and memory efficiency, meaning even large datasets can be leveraged effectively when used grapefully in terms of distributed and parallel computing. XGBoost also offers a myriad of hyperparameters (like learning rate, tree depth, and regularization parameters) that can be tune and optimized for how to best utilize for maximum model performance while trying to govern its behavior.

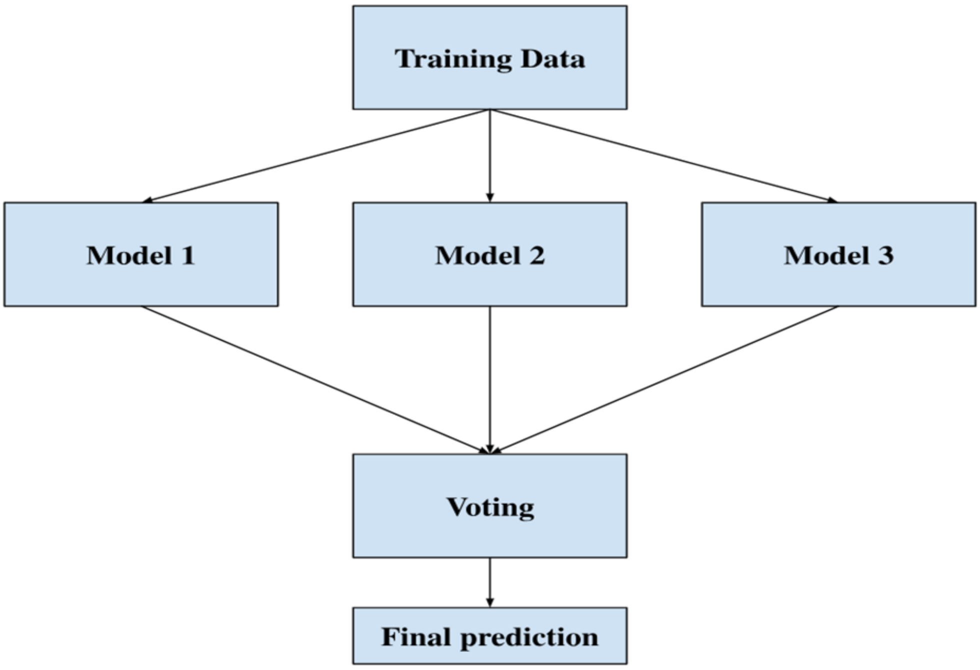


ADABOOST

After each round, the errors are made more important while the correct answers are ignored. As a result, newer learners can focus more on fixing those earlier cases that were not guessed correctly. AdaBoost builds learners one after another, each being weak. With every iteration, a fresh weak learner is added to the group and their success is weighted based on their accuracy. The stronger influence weak learners have is reflected in an improved accuracy. All weak learners' predictions are merged by AdaBoost with weighted voting and the weight of each learner depends on how correctly their training predictions were matched. When all the weak learners have made their forecasts, the last step is to add up those predictions. AdaBoost uses "boosting" to mean that it helps improve the performance of poorer cases by giving additional weight to difficult cases during training. The AdaBoost technique combines a group of weak learners and changes the weights of learning cases, so it creates a strong ensemble classifier for correct results. People often say that AdaBoost is quick to implement, very efficient and less likely to overfit the data.

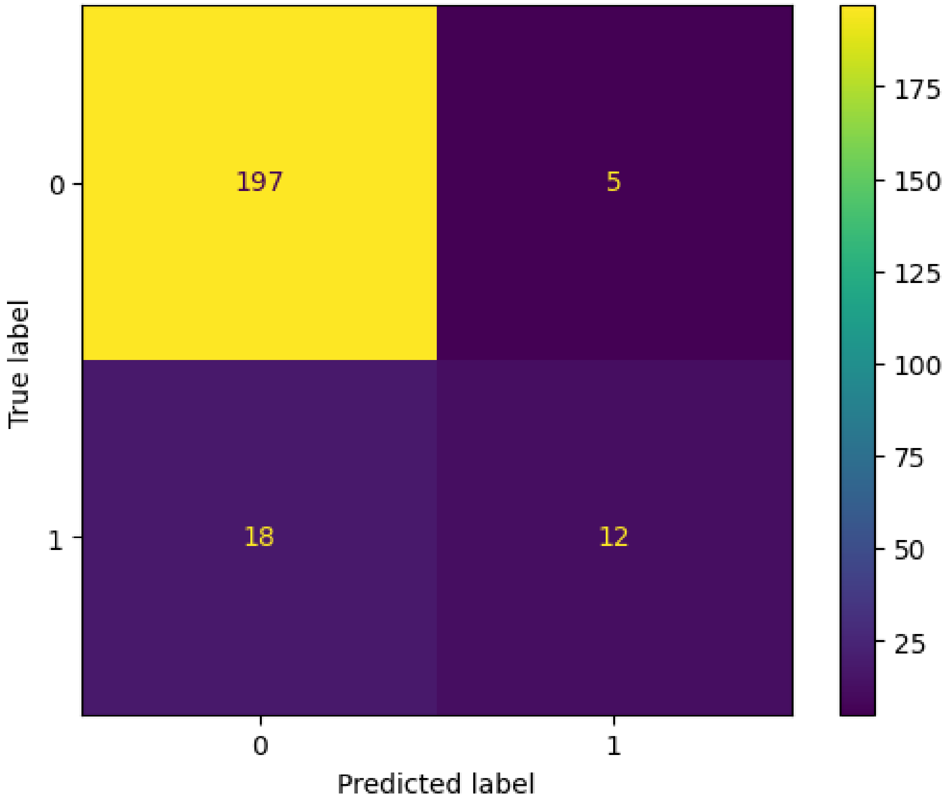


Voting Classifier: To create the ultimate prediction, a Voting Classifier uses a voting method (either hard or soft voting) to aggregate results from several different classifiers. This type of ensemble learning makes use of the combined expertise of several classifiers to generate results that are more precise and reliable than those made by a single classifier. As we know it's a combination of many more classifier in my study I used here Gradient Boosting twice, Random Forest classifier and Logistic Regression classifier. I set the voting mechanism as soft that is means it's taken all the classifier predication and average them to make final prediction.

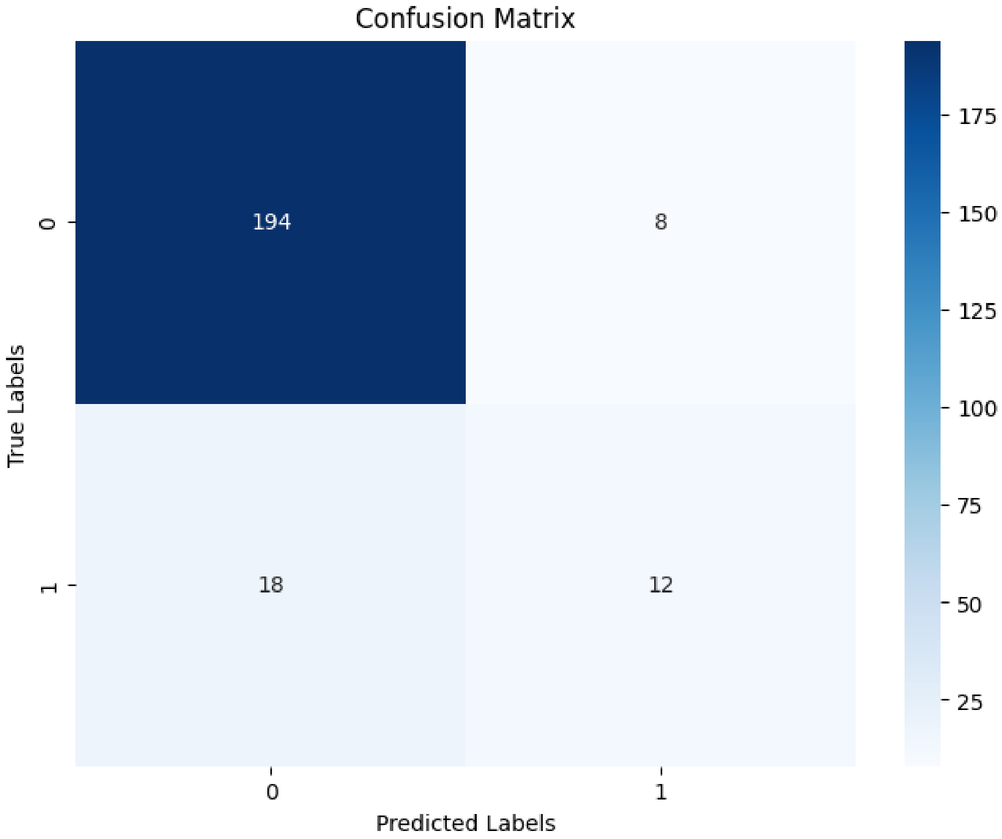


Confusion Matrix:

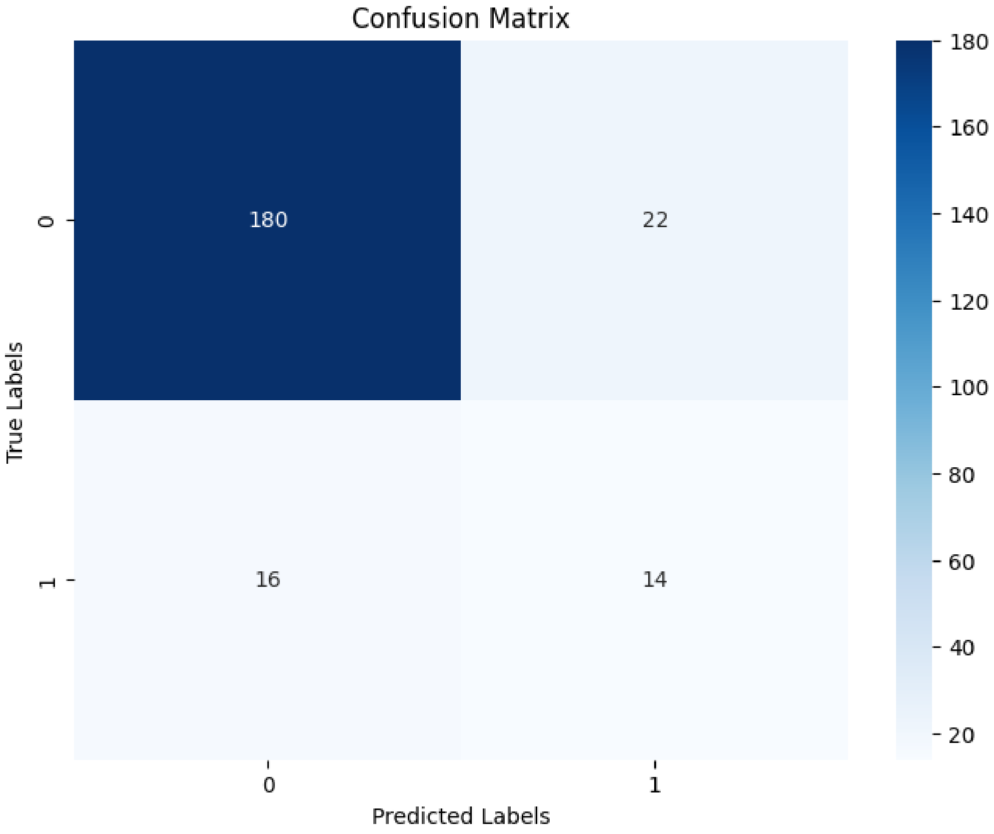
Logistic regression



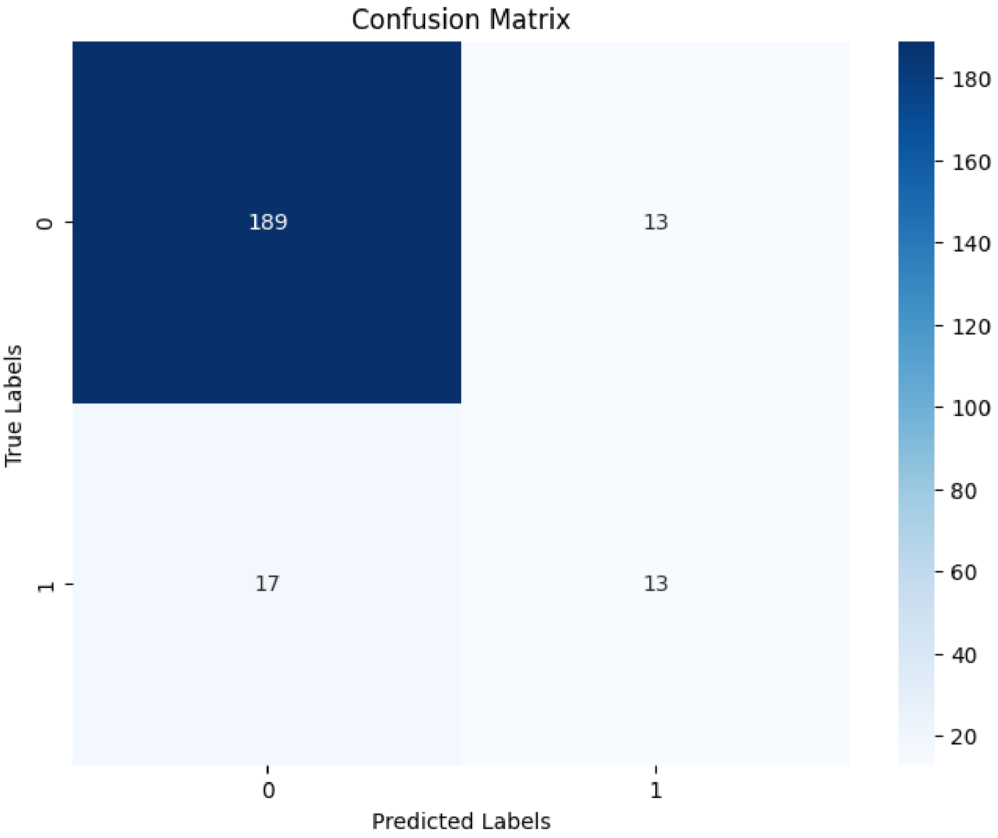
Random forest



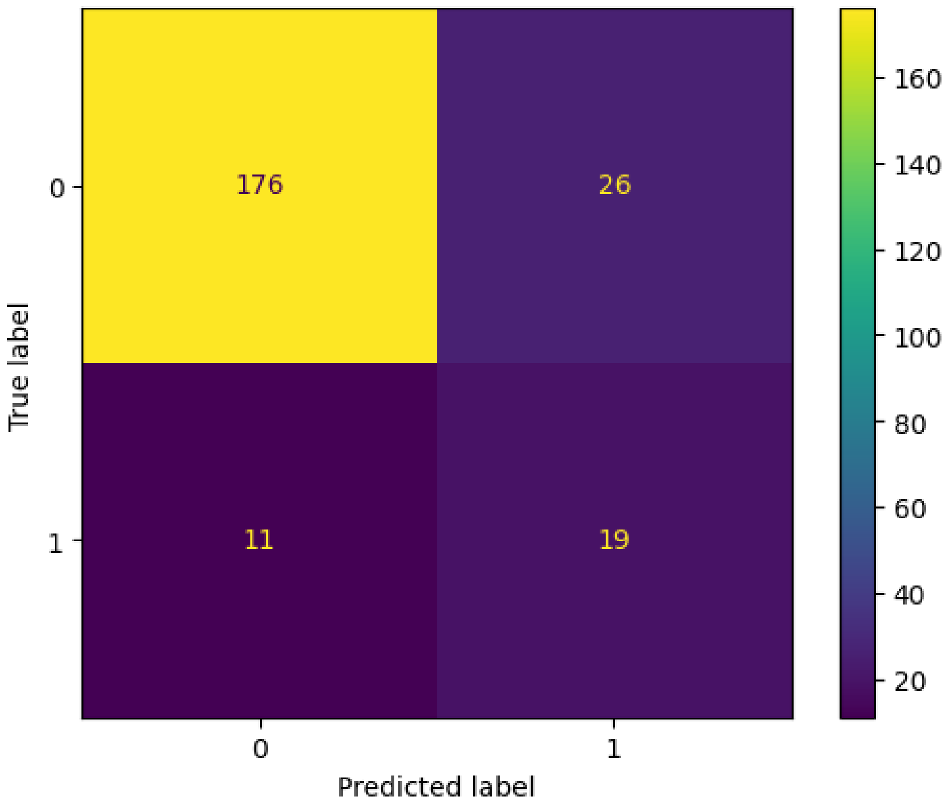
Decision Tree



XGB Classifier



Voting Classifier



3.2 Detailed Methodology and Design

Several ways and techniques were examined to design a dependable system for making forecasts about ICU patient death. The first set of models to be tested were K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM) and Logistic Regression. It is well known that these models perform well and are easy to interpret in situations where classification happens to be binary. Even so, they often fail to give accurate results when used on complicated datasets without correcting their unbalanced or noisy features.

Besides simple classifiers, Ridge Regression and Artificial Neural Networks were also examined in this experiment. Because it is crucial to explain models in healthcare, neural networks are not as popular there, as they are "black boxes," use a great deal of data and are computationally expensive.

Improving the prediction results, a Voting Classifier was used that takes advantage of the different skills of several models. The approach uses suggestions from various classifiers and selects the majority response, raising the method's reliability. Since interpretability and computation were limited with conventional ensemble methods, a mix of base classifiers and ensemble techniques was chosen. Strong performance was ensured by exhaustively testing the models with ROC-AUC, F1-score, accuracy, precision and recall.

3.3 Summary

The project forecasts who among ICU patients may be at risk of death by using both ensemble techniques and individual machine learning tools. To start, the models K-Nearest Neighbors, Decision Trees and Logistic Regression were analyzed. Such methods were chosen after evaluations for their stronger accuracy and ability to resist mistakes. All of the strengths of the individual models were put together in a single predictive system that uses a Voting Classifier.

Chapter 4

Implementation and Results

This chapter shows how to use machine learning and ensemble methods, evaluate their performance and analyze their results. Logistic regression is shown to be the most accurate model and outcomes are compared using well-known evaluation methods.

4.1 Environment Setup

The development environment had to be established to help carry out this project the best way possible. Experimentation in the system is achieved with Python data science libraries. Easy visualization and simple operation made Jupyter Notebook the preferred way to code and run tests.

Various frameworks and libraries were needed to implement the application.

For working with numerical data and simple manipulation, you should use NumPy and Pandas.

To show results from evaluation, use Matplotlib and Seaborn.

Collection of ensemble methods and machine learning models is accomplished with scikit-learn.

All experiments were done using devices that met these specifications:

Intel Core i5/i7 or equivalent CPUs are required.

You should get a memory of at most 8 GB.

the user has a choice between Windows and Linux.

There are two choices for an integrated development environment: Jupyter Notebook (via Anaconda) or Google Colab for cloud computing.

This kind of setup made it possible to make the same results again and trained, assessed and compared models efficiently.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

Successful prediction of mortality in ICU patients was tested and analyzed to better understand the models' results. Using the same data, we tested all machine learning and ensemble models to guarantee that none of them were unfair. Models were judged based on how accurate, precise, recall-able and fair they were, as well as their AUC on the ROC curve.

4.3 Results and Discussion

Existing work	Accuracy (%)	Our accuracy (%)
Luo, C., Zhu, Y., Zhu, Z., Li, R., Chen, G., & Wang, Z. (2022).	83%	90%
Veith, N., & Steele, R. (2018, April).	75%	90%
Lorenzoni, G., Sella, N., Boscolo, A., Azzolina, D., Bartolotta, P., Pasin, L., ... & Gregori, D. (2021).	85%	90%
Huang, B., Liang, D., Zou, R., Yu, X., Dan, G., Huang, H., ... & Liu, Y. (2021).	89%	90%
Nie, X., Cai, Y., Liu, J., Liu, X., Zhao, J., Yang, Z., ... & Liu, L. (2021).	81%	90%

Table 4.1: Comparative result

Our model's performance:

Algorithm	Precision (%)	Recall (%)	F1-score (%)	ROC (%)	Accuracy (%)
Logistic Regression	81.5%	69%	77.5%	69%	90%
Decision Tree	65.5%	68%	66%	70%	84%
Random Forest	75%	72%	73%	68%	89%
KNN	81.5%	63%	66.5%	69%	89%
Ada Boost	64.5%	66.5%	65.5%	68%	84%
XG Boost	86%	87%	87%	78%	87%

Table 4.2: Our model's performance

Many models are shown with their results in this table. I selected various machine learning techniques such as KNN, Decision Tree and Logistic Regression, plus other ensemble models such as Random Forest, Ada Boost, XG Boost and voting classifier. When we do logistic regression, we find that its accuracy is 90%, precision is 81.5%, recall is 69%, F1-score is 77.5% and ROC is 69%. XG boost, Random Forest and KNN all give the same second-best result of 89%, while the decision tree gives 84%.

4.4 Summary

This chapter shows how the proposed method was applied and analyzes all the results thoroughly. The environment used Python and a number of helpful libraries to prepare the data, build models, review their results and visualize them. All the patient records in the data, including their symptoms, test results and details, were processed and cleaned to achieve accuracy. When called for, the data was normalized, any missing data was resolved and categorical features were encoded. After doing preprocessing, several machine learning models were tried, including support vector machines (SVM), logistic regression, decision trees and K-nearest neighbors (KNN). In addition to single models, ensembles of Random Forest, AdaBoost and XGBoost were tested to better predict patient outcomes. We chose the most relevant predictors of mortality risk using feature selection methods which made the model less complex without making it less accurate. Once the datasets were processed, each model was tested using accuracy, precision, recall, F1-score and ROC-AUC. All three methods scored about the same except logistic regression which obtained the highest accuracy of 90%. Following that, AdaBoost with 87.07% and XGBoost with 84% were tried as ensemble methods. The performance of thresholds and the accuracy of classification were checked using ROC curves and confusion matrices.

Chapter 5

Engineering Standards and Design Challenges

In this chapter, we review the hardware, software and communication standards necessary to maintain safety and dependability in a system. As well as a sustainability plan for the future, it also looks at how the system influences society, morals and the environment

.

5.1 Compliance with the Standards

5.1.1 Software Standards

In this project, software standards help ensure quality, maintainability, and compatibility with healthcare systems. We followed general software engineering practices, including coding standards, data privacy guidelines, and evaluation methods. One key standard followed is ISO/IEC 25010, which focuses on software product qualityIt guarantees the system's dependability, usability, and performance—all of which are essential for medical applications. HIPAA (Health Insurance Portability and Accountability Act) is another crucial regulation that would be particularly pertinent if the system were to be implemented in actual hospitals. It ensures patient data privacy and security. Although this project uses publicly available data, the system was designed with these standards in mind for future real-world use. Alternatives like using only internal coding practices without reference to international standards were considered. While simpler, such alternatives lack global recognition and may not be suitable for deployment in medical environments. Therefore, choosing recognized standards ensures better acceptance, safety, and compliance.

5.1.2 Hardware Standards

Although the focus is on software, hardware standards are essential for using this system in actual ICUs. The safety and functioning of medical electrical equipment are most often regulated by IEC 60601. Because of this standard, any equipment connected to patients or the hospital is designed to be reliable and safe. When the model is eventually used in medical devices or systems, it must watch EMC standards to avoid interfering with other hospital equipment. Instead of certified hardware, the system could be put on computers built for general use as long as they follow most of the rules. The method is effective in simulated settings but cannot be safely used in real ICUs. When you design according to these standards, medical hardware can be more easily added, protecting the system's safety, accuracy and trust in clinical settings.

5.1.3 Communication Standards

In ICUs and other hospital departments, consistent communication allows different systems such as monitors, ventilators and central data systems, to share information accurately and safely. HL7 which is short for Health Level Seven, plays a key role in sharing clinical and administrative data between healthcare computer systems. Information such as patient records, testing results and admissions data is easier to share using one format. DICOM (Digital Imaging and Communications in Medicine) is a standard for when medical imaging information (such as X-ray or MRI records) must be incorporated into the system. Building web applications that exchange patient data through hospital servers usually depends on TCP/IP and HTTP/HTTPS. In some ways, alternative approaches like rolling your own communication protocols give flexibility for software development but because they are not standard, it can be hard to integrate them into practical systems used by hospitals. If you use accepted communication methods, it is easier for others to join your system, ensures security and fits better with your hospital's IT system.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

Intensive care unit patients' outcomes are strongly affected by using machine learning and ensemble approaches. An early and proper analysis of data collected in these systems can allow doctors to detect dangerous health problems beforehand. Quickly identifying the disease allows doctors to start treatment sooner, possibly saving lives and keeping risks of complications down. In addition, accurate evaluation of a patient's death risk allows healthcare providers to give immediate attention to those cases that are most critical. It helps patients live longer and takes less work for the medical team. In the future, these systems can help ICUs offer improved care, lessen family worries and guide doctors with data when taking action. As a result, healthcare can be delivered faster, more safely and with better responsiveness.

5.2.2 Impact on Society & Environment

The use of machine learning for ICU mortality forecasting can help society. Easily finding and addressing problems reduces the need for unneeded medical actions and enhances the performance of healthcare systems. As a result, families can access better healthcare for lower costs and the public's trust in healthcare technology grows. Such systems make patient care more effective and boost survival hopes in understaffed and poorly resourced hospitals. Having data tools in health information technology makes it easier for medical workers to decide fast, leading to better outcomes, mostly in case of a pandemic crisis. Because digital analysis uses records and tests electronically, it saves on paper and medical waste and also leads to using less energy. Even with a small environmental effect, data-focused strategies encourage environmentally-friendly healthcare.

5.2.3 Ethical Aspects

There are many ethical problems involved in using machine learning to predict ICU mortality. Data privacy should be the top issue. All confidentiality measures must be applied when dealing with data about patients which could include sensitive information. Any healthcare data should be protected with privacy and it can only be used for the betterment of healthcare, by ethical rule. It is also a serious problem that algorithms may be biased. As a result, some groups of patients might not receive the care they deserve. For this reason, we should strive to make model development fair and transparent. Another issue is to valuing the freedom of patients and those working in health care. Medical personnel should still depend on their experience, despite how helpfully machine learning approaches can guide decisions. Technology can make healthcare professionals' decisions more informed, instead of fully replacing what they know. Let's not forget the topic of responsibility. It is very important to know exactly who is responsible when there are errors in predictions, regardless of whether the team, healthcare professionals or the system caused them. Paying attention to these ethical problems ensures that machine learning benefits healthcare, is treated fairly and is performed transparently.

5.2.4 Sustainability Plan

Making sure the ICU mortality–prediction system works well in the future requires an organized plan for upkeeping, developing and complying with rules. The models will be updated regularly using recent patient records, ensuring treatment policy modifications and demographic trends are considered in their results. Because the application is hosted on the cloud, it can manage both more data and users without having to change the infrastructure much. Meaningful data exchange happens without hassle, since the new tool is integrated with the existing information systems. Everyone involved in the project will be taught how to expertly interpret results and offer analysis on the model's quality, allowing for regular and better updates. The workings of the platform will include closely watching and adjusting resource use so optimization is possible, thus saving on spending and environmental resources.

5.3 Complex Engineering Problem

5.3.1 Complex Problem Solving

EP1 (Dept of Knowledge)	**EP2 (Range of Conflicting Requirements)**	**EP3 (Depth of Analysis)**	**EP4 (Familiarity of Issues)**	**EP5 (Extent of Applicable Codes)**	**EP6 (Extent of Stakeholder Involvement)**	**EP7 (Interdependence)**
Data Science, Machine Learning	High complexity in predicting ICU mortality	In-depth analysis of data, model training, and testing	High familiarity with ML algorithms like Logistic Regression, Decision Trees, etc.	Use of ML frameworks, Python, and cloud services	Involves healthcare providers, researchers, and data engineers	Interdependent relationships between data, models, and outcomes in a healthcare context

Table 5.1: Mapping with complex problem solving.

Mapping with Knowledge Profile for EP1

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
Statistics, probability theory, linear algebra	Machine learning algorithms (logistic regression, decision trees, SVM)	Architecture of data pipelines and model ensembles	Coding in Python; use of Scikit-learn, XGBoost, hyperparameter tuning	Key publications on ICU mortality prediction and ensemble methods

Table 5.2: Mapping with knowledge Profile.

5.3.2 Engineering Activities

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
Use of computing resources (servers, cloud), datasets (ICU data), and software tools (Python, Scikit-learn, XGBoost).	Interaction with ICU healthcare experts, data scientists, and developers for model improvement.	Development of ensemble models (e.g., random forest, XGBoost) for higher accuracy.	The model can improve early detection of ICU patient mortality, positively impacting patient outcomes.	Familiarity with machine learning algorithms, healthcare data, and the importance of prediction in critical care.

Table 5.3: Mapping with complex engineering activities.

5.4 Summary

Just like industry practices in data analysis and predictive modeling, the method counts on standard machine learning algorithms such as logistic regression, decision trees and more advanced methods such as random forests and gradient boosting.

Creating a design is not always easy. Larger and noisier sets of data need to be managed well, understanding what the model results mean is crucial and keeping data private is essential in healthcare. Because healthcare is complex, with many types of people involved and lots of different medical conditions, the design process is not simple. To face these issues, you need both knowledge of technology and a dedication to safety, security and dependability standards.

The system will perform well and scale as needed only if you select the proper hardware and software standards. It is important also that communication norms in healthcare systems let professionals use the system quickly and easily.

Chapter 6

Conclusion

This study shows how machine learning, particularly ensemble approaches and logistic regression, can improve patient care and ICU mortality prediction. For wider clinical use, future advancements will incorporate explainable AI, deep learning, and real-time data.

6.1 Summary

The experiment revealed that using machine learning and ensemble methods may greatly improve care in intensive care units. It divides patients based on risk and calculates expected deaths by studying key signs, laboratory results and information about a person's background. Random Forest, XGBoost and AdaBoost were examined, as well as logistic regression, decision trees and K closest neighbors. Logistic regression was most accurate with 90%, then random forest and KNN with 89%. These findings show that blending various algorithms in ensemble learning leads to more accurate results and greater robustness to problems with inadequate or noisy data. If prediction technology is used with clinicians' workflows, it can provide advance notice about high-risk patients for their attention. As a result, doctors can act fast, save money and see better outcomes for patients. Moving forward, more improvements can be made by feeding real-time information, relying on deep learning to discover more complicated patterns and trying out the models in many medical settings. Developing this tool further would allow it to provide helpful guidance in emergencies and save lives in critical care units, as long as it keeps up-to-date and works with specialist teams.

6.2 Limitation

Notwithstanding good outcomes, this system still has some restrictions. A further problem is that models developed from one hospital's data might not do well in other hospitals with different groups of patients or medical practices. In addition, even though ensemble approaches improve the accuracy of risk determination, they can make it tough for doctors to understand exactly why a patient was identified as high risk. Additionally, all patient data is viewed at once and rapid changes happening to patients may not be detected by it. Because advanced models need a lot of computing, using them for real-time deployment could be tough in places where resources are scarce.

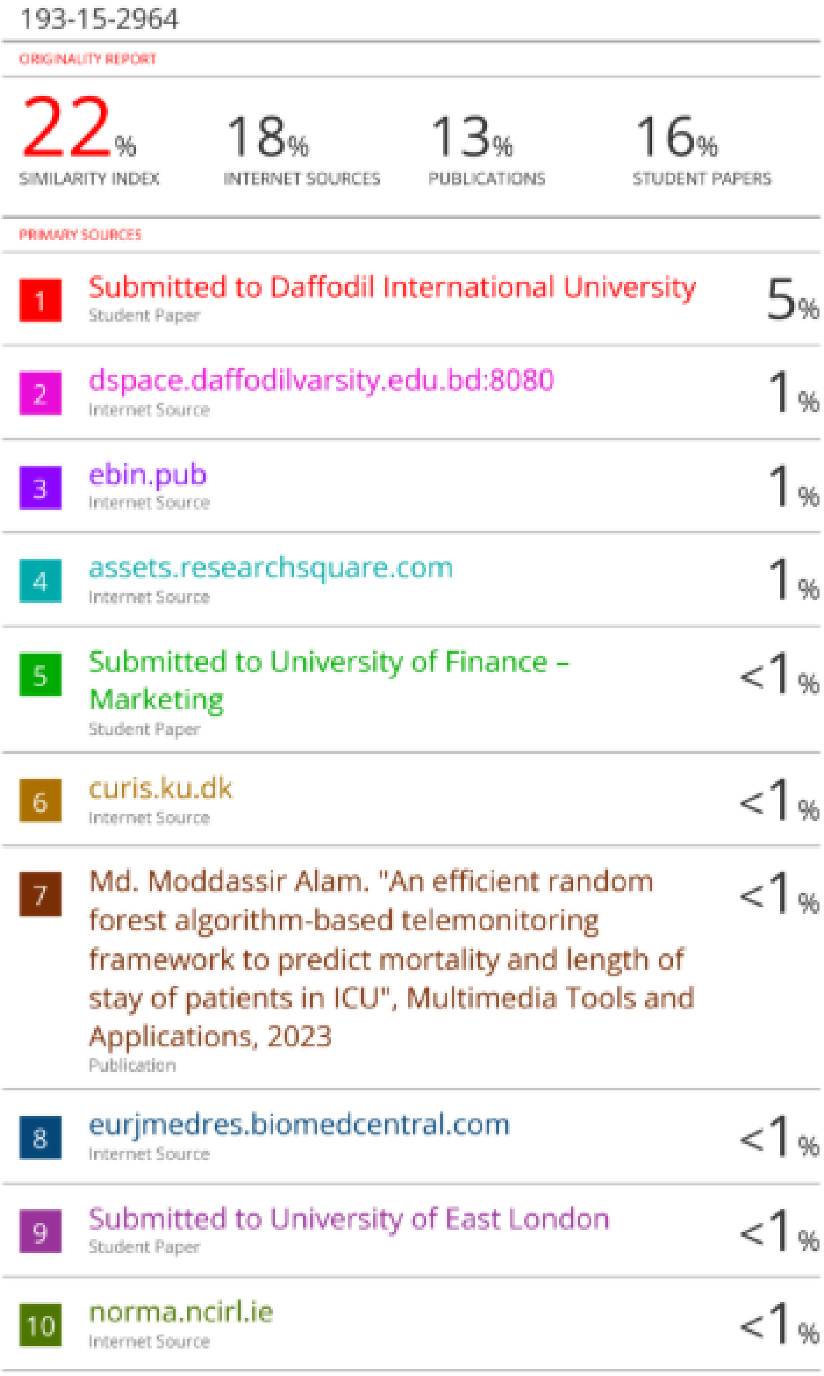
6.3 Future Work

Following the current results, there are a number of ways this ICU mortality prediction system can be improved and extended. First, using a constant feed of vital signs and lab results in real time would allow the model to respond faster to changes in a patient. Second, looking into deep learning structures such as recurrent neural networks or transformers, helps detect hard-to-find sequence patterns that simple algorithms do not notice. Besides, bringing more kinds of data into the system, for example clinical literature or imaging, might enhance the model and make it more precise. Seeing how models work is now a vital concern. Using explainable AI ways such as SHAP values or LIME, would assist clinicians in grasping the reasons for a predicted risk. If the system is used in various hospitals with different kinds of patients, we can learn if this approach works for all types of patients.

References

1. Iwase, S., Nakada, T. A., Shimada, T., Oami, T., Shimazui, T., Takahashi, N., ... & Kawakami, E. (2022). Prediction algorithm for ICU mortality and length of stay using machine learning. *Scientific reports*, *12*(1), 12912.
2. Yun, K., Oh, J., Hong, T. H., & Kim, E. Y. (2021). Prediction of mortality in surgical intensive care unit patients using machine learning algorithms. *Frontiers in Medicine*, *8*, 621861.
3. Kong, G., Lin, K., & Hu, Y. (2020). Using machine learning methods to predict in-hospital mortality of sepsis patients in the ICU. *BMC medical informatics and decision making*, *20*, 1-10.
4. Luo, C., Zhu, Y., Zhu, Z., Li, R., Chen, G., & Wang, Z. (2022). A machine learning-based risk stratification tool for in-hospital mortality of intensive care unit patients with heart failure. *Journal of translational medicine*, *20*(1), 136.
5. Thorsen-Meyer, H. C., Nielsen, A. B., Nielsen, A. P., Kaas-Hansen, B. S., Toft, P., Schierbeck, J., ... & Perner, A. (2020). Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, *2*(4), e179-e191.
6. Kong, G., Lin, K., & Hu, Y. (2020). Using machine learning methods to predict in-hospital mortality of sepsis patients in the ICU. *BMC medical informatics and decision making*, *20*, 1-10.
7. Veith, N., & Steele, R. (2018, April). Machine learning-based prediction of ICU patient mortality at time of admission. In *Proceedings of the 2nd international conference on information system and data mining* (pp. 34-38).
8. Nistal-Nuño, B. (2022). Developing machine learning models for prediction of mortality in the medical intensive care unit. *Computer Methods and Programs in Biomedicine*, *216*, 106663.
9. Lorenzoni, G., Sella, N., Boscolo, A., Azzolina, D., Bartolotta, P., Pasin, L., ... & Gregori, D. (2021). COVID-19 ICU mortality prediction: a machine learning approach using SuperLearner algorithm. *Journal of Anesthesia, Analgesia and Critical Care*, *1*, 1-10.
10. Huang, B., Liang, D., Zou, R., Yu, X., Dan, G., Huang, H., ... & Liu, Y. (2021). Mortality prediction for patients with acute respiratory distress syndrome based on machine learning: a population-based study. *Annals of translational medicine*, *9*(9).
11. Subudhi, S., Verma, A., Patel, A. B., Hardin, C. C., Khandekar, M. J., Lee, H., ... & Jain, R. K. (2021). Comparing machine learning algorithms for predicting ICU admission and mortality in COVID-19. *NPJ digital medicine*, *4*(1), 87.
12. Nie, X., Cai, Y., Liu, J., Liu, X., Zhao, J., Yang, Z., ... & Liu, L. (2021). Mortality prediction in cerebral hemorrhage patients using machine learning algorithms in intensive care units. *Frontiers in neurology*, *11*, 610531.
13. Awad, A., Bader-El-Den, M., McNicholas, J., & Briggs, J. (2017). Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach. *International journal of medical informatics*, *108*, 185-195.
14. Thorsen-Meyer, H. C., Nielsen, A. B., Nielsen, A. P., Kaas-Hansen, B. S., Toft, P., Schierbeck, J., ... & Perner, A. (2020). Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, *2*(4), e179-e191.

15. Alghatani, K., Ammar, N., Rezgui, A., & Shaban-Nejad, A. (2021). Predicting intensive care unit length of stay and mortality using patient vital signs: machine learning model development and validation. *JMIR medical informatics*, *9*(5), e21347.
16. A. M. B. Fentie, M. A. D. Tadesse, A. K. Demessie, and D. A. Jemere, “Determinants of mortality of patients admitted to the intensive care unit in a tertiary hospital,” *Patient Related Outcome Measures*.
17. J. D. Nguyen *et al.*, “Mortality prediction in the intensive care unit using machine learning techniques,” *Scientific Reports*, vol. 13, no. 1
18. M. Sharma, “Ensemble Learning Methods: Bagging, Boosting and Stacking,” *Analytics Vidhya*, Jan.
19. E. C. Chang, J. R. Wang, and M. T. Lin, “Predicting ICU mortality risk using machine learning algorithms with electronic medical records,” *IEEE Access*, vol. 8, pp. 76737–76748.
20. A. Rajkomar, J. Dean, and I. Kohane, “Machine learning in medicine,” *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358.



193-15-2964

ORIGINALITY REPORT

22%	18%	13%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	ebin.pub Internet Source	1%
4	assets.researchsquare.com Internet Source	1%
5	Submitted to University of Finance – Marketing Student Paper	<1%
6	curis.ku.dk Internet Source	<1%
7	Md. Moddassir Alam. "An efficient random forest algorithm-based telemonitoring framework to predict mortality and length of stay of patients in ICU", Multimedia Tools and Applications, 2023 Publication	<1%
8	eurjmedres.biomedcentral.com Internet Source	<1%
9	Submitted to University of East London Student Paper	<1%
10	norma.ncirl.ie Internet Source	<1%