PREDICT ADMISSION OF CONFIRMED COVID-19 CASES TO ICU

Project Report

for

KNOWLEDGE MANAGEMENT SYSTEM

in

B. Tech Computer Science and Engineering

By

20BCE2053 - Kartikey Matolia

Slot : C2

Under the Guidance of

Prof. HARI RAM VISHWAKARMA

Senior Professor, SCOPE

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

1.Abstract

Health systems all throughout the world have been impacted by the most recent COVID-19 pandemic. Critically sick patients have received crucial care in intensive care units (ICUs), in particular. The quick spread of the virus has increased admissions, but this has also created a number of issues for ICU wards, including a lack of ICU beds, a staff that is overworked caring for patients, and a lack of medical resources to treat everyone in hospitals. These problems may have had a direct impact on a patient's survival by lowering the quality of healthcare services offered. The project's goal is to predict the admission of covid-19 patients to ICU because we already have a shortage of beds in ICU to treat severely affected patients. This application assists hospitals in admitting critical patients only to ICU by analysing their reports, which include various attributes collected from the patients such as temperature difference, age, blood pressure, heart rate, respiratory rate, oxygen saturation, and a few other attributes. The accurate prediction is challenging task so we use possible machine learning techniques such as logistic regression, gaussian naïve bayes, SGD classifier (Stochastic gradient descent) and XGB regressor(Extreme Gradient Boosting) and compare the performances of individual model based upon the metrics such as accuracy, precision, ROC curve, F1 score and others to implement the application with better model.

2.Introduction

a. General overview or background of the project,

When the World Health Organization declared Covid 19 a pandemic, everyone was urged to seek personal protection**.** The number of patients infected by this virus is growing by the day. Many countries have faced numerous challenges in trying to keep their health systems responsive and capable of providing essential health services. There were numerous issues raised during the treatment of critically ill patients. To avoid overburdening staff and insufficient medical resources in the event of an upcoming virus, we must improve the ICU management plan by taking into account the true patient severity. To avoid the time-consuming process and incorrect decisions made by doctors on the spot, we developed a prediction system based on training the severe covid patient data and testing the new patient data to determine the severity. The prediction system that determines whether the patient needs to be admitted to the ICU needs to be more accurate, so we investigated all available machine learning techniques that provide accurate results in less time and built a prediction system for admission of confirmed COVID-19 patients to the ICU. In this project, we aim to improve the ICU management plan by using the best machine learning techniques to predict whether a person needs to be admitted to the ICU. This allows hospitals to treat the right person in less time. As a result, doctors in hospitals do not need to spend more time analysing patient records and do not need to waste ICU beds.

b. Literature survey

Title	**Description**	**Advantages & Disadvantages**
COVID-19 in CXR: from Detection and Severity Scoring to Patient Disease Monitoring	It suggests a CXR-based severity scoring system for disease prognosis and treatment decisions, as well as the use of AI algorithms to automate CXR analysis. CXR can track disease progression and response to treatment, detect complications, and aid in triaging patients in resource-limited settings. The paper also discusses ethical considerations related to CXR use. Overall, the paper emphasizes the valuable role of CXR in COVID-19 detection and monitoring, and recommends further research in this area.	**Advantages:** Non-invasive Wide availability Easy to interpret **Disadvantages:** Low sensitivity False negatives Limited information

Classification of Severe and Critical Covid-19 Using Deep Learning and Radiomics	This paper proposes a novel approach to predict severe and critical cases of COVID-19 using deep learning and radiomics. The authors developed a model that combines deep learning and radiomics features from CT scans to accurately classify patients as severe or critical. The proposed model achieved high accuracy, sensitivity, and specificity in differentiating between severe and critical cases, which can aid in clinical decision-making and resource allocation. The spaper also highlights the potential for the proposed model to be used as a screening tool for early detection of severe and critical COVID-19 cases, and recommends further research to validate its clinical effectiveness.	**Advantages:** High accuracy Automated diagnosis Large dataset utilization **Disadvantages:** High computational cost Lack of interpretability Lack of standardization
A machine learning approach to predicting Covid-19 cases Amongst suspected cases and their category of admission	Authors presents a machine learning model to predict COVID-19 cases among suspected cases and their category of admission. The authors developed a model that uses demographic, clinical, and laboratory data to predict the likelihood of a patient being COVID-19 positive and their category of admission. The proposed model achieved high accuracy and sensitivity in predicting COVID-19 cases and their category of admission, which can aid in clinical decision-making and resource allocation. The paper also highlights the potential for the proposed model to be used as a screening tool for early detection of COVID-19 cases, and recommends further research to validate its clinical effectiveness.	**Advantages:** Improved accuracy Automation Data-driven **Disadvantages:** Bias in data Lack of interpretability Limitation in clinical use Need for large amounts of data
COVID 19 severity of Pneumonia	The authors proposes a deep learning model to analyze chest X-rays and predict the severity of COVID-19 pneumonia. The authors developed a	**Advantages:** Wide availability Non-invasive

Analysis using chest X-rays	model that utilizes convolutional neural networks (CNN) to classify chest X-rays into four categories of pneumonia severity. The proposed model achieved high accuracy and sensitivity in predicting the severity of COVID-19 pneumonia, which can aid in clinical decision-making and treatment planning. The paper also discusses the importance of frequent chest X-ray monitoring in COVID-19 patients with pneumonia, and the potential for the proposed model to assist in the early detection and treatment of severe COVID-19 cases. Overall, the paper highlights the valuable role of chest X-rays and deep learning models in the assessment and management of COVID-19 pneumonia.	Quick results **Disadvantages:** Limited sensitivity Inter-observer variability Radiation exposure Limited quantification
A comparative study of machine learning models for COVID-19 prediction in India	This paper presents a comparative analysis of different machine learning models for predicting the spread of COVID-19 in India. The authors evaluated the performance of various models, including random forest, XGBoost, support vector machine (SVM), and artificial neural network (ANN), based on different metrics such as accuracy, sensitivity, and specificity. The results showed that the random forest model had the highest accuracy and sensitivity in predicting COVID-19 cases in India. The paper also discusses the importance of accurate and timely predictions for effective COVID-19 control and management. Overall, the paper highlights the potential of machine learning models in predicting COVID-19 spread in India and recommends further research to improve their performance.	**Advantages:** Helps in identifying the most effective model for COVID-19 prediction, taking into account the Indian context and relevant factors. Enables public health officials to make informed decisions based on accurate and reliable

		predictions of COVID-19 spread. **Disadvantages:** Dependence on accurate and up-to-date data Model limitations Complexity Overreliance on technology
A covid-19 patient severity stratification using a 3d convolutional strategy on ct-scans	This study proposes a deep learning model to predict the severity of COVID-19 patients based on CT scans. The authors developed a model that uses 3D convolutional neural networks (CNN) to analyze CT scans and classify patients into four categories of severity. The proposed model achieved high accuracy and sensitivity in predicting the severity of COVID-19 patients, which can aid in clinical decision-making and treatment planning. The paper also highlights the potential for the proposed model to be used as a screening tool for early detection of severe COVID-19 cases, and the importance of frequent CT scan monitoring in COVID-19 patients. Overall, the paper emphasizes the valuable role of deep learning models and CT scans in the assessment and management of COVID-19 patients.	**Advantages:** High resolution Quantitative information Automated analysis Improved accuracy **Disadvantages:** High cost Radiation exposure Computational cost Limitations in clinical use
Prediction of ICU admission for COVID-19	This paper proposes a machine learning model to predict the likelihood of ICU admission for COVID-19 patients based on complete blood	**Advantages:** With accurate prediction, the

patients: a Machine Learning approach based on Complete Blood Count data	count (CBC) data. The authors developed a model that uses decision trees and support vector machines (SVM) to analyze CBC data and predict the likelihood of ICU admission. The proposed model achieved high accuracy and sensitivity in predicting ICU admission for COVID-19 patients, which can aid in clinical decision-making and resource allocation. The paper also discusses the potential for the proposed model to be used as a screening tool for early detection of high-risk COVID-19 patients and the importance of CBC monitoring in COVID-19 patients. Overall, the paper highlights the valuable role of machine learning models and CBC data in the assessment and management of COVID-19 patients.	healthcare system can prepare for an expected increase in demand for ICU beds, and allocate resources accordingly. **Disadvantages:** Blood count data may not capture all relevant information about a patient's health, and so the model's predictions may not be 100% accurate. Predictive models can also give false results, leading to inappropriate patient management and waste of resources.
Machine Learning to Predict ICU	This paper proposes a machine learning model to predict ICU admission, ICU mortality, and survivors' length of stay among COVID-19	**Advantages:** Improved accuracy

Admission, ICU Mortality and Survivors' Length of Stay among COVID-19 Patients: Toward Optimal Allocation of ICU Resources	patients. The authors developed a model that uses demographic, clinical, and laboratory data to predict the likelihood of ICU admission, ICU mortality, and survivors' length of stay. The proposed model achieved high accuracy and sensitivity in predicting ICU admission, ICU mortality, and survivors' length of stay, which can aid in optimal allocation of ICU resources and clinical decision-making. The paper also highlights the potential for the proposed model to assist in the early detection and treatment of high-risk COVID-19 patients, and recommends further research to validate its clinical effectiveness. Overall, the paper emphasizes the valuable role of machine learning models in the assessment and management of COVID-19 patients in ICU settings.	Automation Personalization Early identification of high-risk patients **Disadvantages:** Data quality and bias Technical expertise Privacy and security Black box predictions
Predictions of COVID-19 Infection Severity based on co-associations between the SNPs of Co-morbid diseases and COVID-19 through Machine	This paper proposes a machine learning model to predict the severity of COVID-19 infection based on the co-associations between single nucleotide polymorphisms (SNPs) of co-morbid diseases and COVID-19. The authors developed a model that uses genetic data to identify potential genetic risk factors and predict the severity of COVID-19 infection. The proposed model achieved high accuracy and sensitivity in predicting the severity of COVID-19 infection, which can aid in clinical decision-making and treatment planning. The paper also highlights the potential for the proposed model to be used as a screening tool for early detection of high-risk COVID-19 patients and the importance of genetic data in COVID-19 research. Overall, the paper emphasizes the valuable role of	**Advantages:** Personalized predictions Improved understanding of disease mechanisms Increased efficiency Improved accuracy **Disadvantages:** Technical limitations Limited sample size

Learning of Genetic Data	machine learning models and genetic data in the assessment and management of COVID-19 patients.	Ethical concerns Lack of standardization
Severity Monitoring Device for COVID-19 Positive Patients	This paper proposes a device for monitoring the severity of COVID-19 in positive patients. The authors developed a device that measures vital signs such as heart rate, respiration rate, and oxygen saturation, and uses machine learning algorithms to predict the severity of COVID-19 infection. The proposed device is non-invasive, easy to use, and can be remotely monitored, which can aid in early detection of deterioration in patients and timely medical intervention. The paper also highlights the potential for the proposed device to assist in the efficient use of hospital resources and reduce the risk of infection transmission. Overall, the paper emphasizes the valuable role of technology and machine learning algorithms in the assessment and management of COVID-19 patients.	**Advantages:** Early detection of deterioration Reduced exposure for healthcare workers Improved resource allocation **Disadvantages:** Technical challenges High Cost Data privacy Dependence on technology
Predicting the severity of COVID-19 in patients: A machine learning approach" by Chen, et al. (2020)	A machine learning approach" by Chen et al. (2020) aims to use machine learning techniques to predict the severity of COVID-19 in patients based on clinical data. The authors collected data from 53 patients with COVID-19 and used logistic regression, decision tree, random forest, and support vector machine algorithms to predict disease severity.	**Advantages:** The study uses a diverse set of machine learning algorithms, which could improve the accuracy of the predictions. **Disadvantages:**

		The study does not consider the impact of demographic and genetic factors on COVID-19 severity, which could affect the accuracy of the prediction models. The study only uses clinical data and does not incorporate other sources of information such as imaging or genetic data.
COVID-19 Severity Prediction using Electronic Health Record Data and Machine Learning" by Ghosh et al. (2021)	The study "COVID-19 Severity Prediction using Electronic Health Record Data and Machine Learning" by Ghosh et al. (2021) aims to predict the severity of COVID-19 in patients using electronic health record (EHR) data and machine learning techniques. The authors collected data from over 11,000 patients with COVID-19 and used various machine learning algorithms, including logistic regression, decision tree, random forest, and gradient boosting, to predict disease severity.	**Advantages:** The study uses a large sample size, which can improve the accuracy and generalizability of the prediction models. The study uses EHR data, which includes a variety of clinical

		information such as vital signs, laboratory results, and medications. **Disadvantages:** The study is limited to a single healthcare system, which may not be representative of the entire population. The study only considers the severity of COVID-19 and does not predict other outcomes such as mortality or hospital readmission**.**
"Machine Learning Approaches for COVID-19 Severity Prediction: A Systematic Review" by	The study "Machine Learning Approaches for COVID-19 Severity Prediction: A Systematic Review" by Alqahtani et al. (2021) is a systematic review of the literature on machine learning approaches for predicting the severity of COVID-19 in patients. The authors reviewed 44 studies that used machine learning algorithms to predict COVID-19 severity based on various clinical and non-clinical parameters.	**Advantages:** The study provides a comprehensive review of the literature on machine learning approaches for COVID-19

Alqahtani et al. (2021)		severity prediction. The study identifies common machine learning algorithms and features used in COVID-19 severity prediction, which could guide future research. **Disadvantages:** The study is limited by the quality and quantity of the included studies, which may not be representative of the entire literature. The study does not compare the accuracy of different machine learning algorithms or feature sets. The study does not provide a

		meta-analysis of the results of the included studies.
"Predicting the Severity of COVID-19 with Deep Learning Algorithms" by Xu et al. (2021)	The study "Predicting the Severity of COVID-19 with Deep Learning Algorithms" by Xu et al. (2021) aims to predict the severity of COVID-19 using deep learning algorithms. The authors collected data from 474 COVID-19 patients and used convolutional neural network (CNN) and long short-term memory (LSTM) models to predict disease severity.	**Advantages:** The study uses deep learning algorithms, which can capture complex patterns in the data and improve the accuracy of the predictions. The study uses a relatively large sample size, which can improve the generalizability of the results. **Disadvantages:** The study is retrospective and may not capture all relevant information about the patients. The study does not compare the accuracy of different deep learning

		algorithms or feature sets. The study is limited to a single healthcare system, which may not be representative of the entire population.
"Predicting the severity of COVID-19 in patients: A machine learning approach" by Chen, et al. (2020)	The study "Predicting the severity of COVID-19 in patients: A machine learning approach" by Chen et al. (2020) aims to use machine learning techniques to predict the severity of COVID-19 in patients based on clinical data. The authors collected data from 53 patients with COVID-19 and used logistic regression, decision tree, random forest, and support vector machine algorithms to predict disease severity.	**Advantages:** The study demonstrates the potential of machine learning algorithms in predicting the outcomes of COVID-19 treatment, which could improve patient care. The study uses a diverse set of machine learning algorithms, which could improve the accuracy of the predictions. **Disadvantages:** The study does not consider the

		impact of demographic and genetic factors on COVID-19 severity, which could affect the accuracy of the prediction models. The study only uses clinical data and does not incorporate other sources of information such as imaging or genetic data.
"Development and Validation of a Clinical Score to Predict Severe Outcomes in Hospitalized Patients with COVID-19" by Fang, et al. (2020)	The study "Development and Validation of a Clinical Score to Predict Severe Outcomes in Hospitalized Patients with COVID-19" by Fang et al. (2020) aims to develop and validate a clinical score to predict severe outcomes in hospitalized patients with COVID-19. The authors collected data from over 1,000 COVID-19 patients and used logistic regression to develop a risk score based on 10 clinical factors.	**Advantages:** The study develops a practical and easy-to-use clinical score that can predict severe outcomes in COVID-19 patients. The study uses a relatively large sample size, which can improve the

		generalizability of the results. The study incorporates a diverse set of clinical factors, including demographic, clinical, and laboratory data. **Disadvantages:** The study is retrospective and may not capture all relevant information about the patients. The study is limited to a single healthcare system, which may not be representative of the entire population. The study only predicts severe outcomes and does not consider other outcomes such as mortality or hospital readmission.

"Predictive Modeling of COVID-19 Disease Severity Based on Electronic Health Records" by Li, et al. (2020)	The study "Predictive Modeling of COVID-19 Disease Severity Based on Electronic Health Records" by Li et al. (2020) aims to develop predictive models to determine the severity of COVID-19 based on electronic health record (EHR) data. The authors collected data from over 5,000 COVID-19 patients and used machine learning algorithms to predict disease severity.	**Advantages:** The study uses a variety of machine learning algorithms, including logistic regression, random forest, and gradient boosting machine, to predict disease severity. The study demonstrates the potential of using EHR data and machine learning algorithms to predict COVID-19 severity, which could improve patient care. **Disadvantages:** The study does not compare the accuracy of the developed models with other machine learning models or clinical scores.

		The study does not consider the impact of demographic and genetic factors on COVID-19 severity, which could affect the accuracy of the prediction models.
"Prognostic models for predicting severe outcomes in patients with COVID-19: a systematic review" by Wang, et al. (2021)	The study "Prognostic models for predicting severe outcomes in patients with COVID-19: a systematic review" by Wang et al. (2021) aims to identify and evaluate existing prognostic models for predicting severe outcomes in patients with COVID-19. The authors conducted a systematic review of 26 studies that developed and validated prognostic models for COVID-19.	**Advantages:** The study provides a comprehensive overview of existing prognostic models for predicting severe outcomes in COVID-19 patients. The study evaluates the performance of the prognostic models using metrics such as sensitivity, specificity, and area under the curve.

		Disadvantages: The study is limited to studies that developed and validated prognostic models and may not include all relevant studies that used other approaches. The study does not evaluate the impact of demographic and genetic factors on COVID-19 severity, which could affect the performance of the prognostic models.
"A Deep Learning Approach for Predicting the Severity of COVID-19 in Patients" by Zhang, et al. (2021)	The study "A Deep Learning Approach for Predicting the Severity of COVID-19 in Patients" by Zhang et al. (2021) aims to develop a deep learning model to predict the severity of COVID-19 in patients based on chest CT scans. The authors collected data from over 2,000 patients and used a convolutional neural network (CNN) to predict the severity of COVID-19 based on chest CT scans.	**Advantages:** The study uses a large dataset with over 2,000 patients, which can improve the generalizability of the results. The study uses a deep learning approach, which

		can automatically extract relevant features from chest CT scans without the need for manual annotation. The study demonstrates the potential of using chest CT scans and deep learning algorithms to predict COVID-19 severity, which could improve patient care. **Disadvantages:** The study is limited to a single healthcare system, which may not be representative of the entire population. The study does not compare the accuracy of the developed deep learning model with other

		machine learning models or clinical scores.
Evidential Reasoning Rule Based Decision Support System for Predicting ICU Admission and In Hospital Death of Trauma	presents a decision support system that uses evidential reasoning and rule-based methods to predict ICU admission and in-hospital death of trauma patients. The study uses a dataset of over 10,000 trauma patients and evaluates the system's performance using sensitivity, specificity, and accuracy. The study focuses on trauma patients, which is a crucial area in healthcare, but the system's limited scope and retrospective nature may not capture all relevant patient information. Finally, the study does not compare the accuracy of the decision support system with other machine learning models or clinical scores.	**Advantages:** 1. Easy to Analyse 2. Easy ICU seat Management 3. Fast Allotment 4. Severity level Analysis **Disadvantages:** Not good Accuracy
Severity Assessment of COVID-19 Based on Feature Extraction and Descriptors	aims to develop a severity assessment system for COVID-19 patients based on feature extraction and descriptors from chest CT images. The study uses a dataset of 112 COVID-19 patients and evaluates the system's performance using metrics such as accuracy, sensitivity, specificity, and area under the curve. The study's non-invasive approach and focus on chest CT images could be beneficial for patients who cannot undergo more invasive diagnostic procedures. However, the limited dataset and lack of comparison to other models or clinical scores may limit the generalizability of the results.	**Advantages:** 1. Severity Assessment 2. Easy detection of Severity level **Disadvantages:** Less Accurate system
Severity and Consolidation	aims to develop a deep learning model to quantify the severity and consolidation of COVID-19 from CT images using hybrid weak labels. The study	**Advantages:** Quality control of collected samples

Quantification of COVID-19 from CT Images Using Deep Learning Based on Hybrid Weak Labels	uses a dataset of 348 COVID-19 patients and evaluates the model's performance using metrics such as accuracy, sensitivity, and specificity. The hybrid weak labeling approach allows for faster and more efficient labeling of data, making it a promising approach for future studies. However, the study's limited dataset and lack of external validation may limit the generalizability of the results.	**Disadvantages:** accuracy of 92.85%, Not 100%
A Generic Deep Learning Based Cough Analysis System from Clinically Validated Samples for Point of-Need Covid-19 Test and Severity Levels	aims to develop a deep learning model to analyze cough sounds for the detection of COVID-19 and prediction of disease severity. The study uses a dataset of clinically validated cough samples from COVID-19 patients and evaluates the model's performance using metrics such as accuracy, sensitivity, and specificity. The non-invasive and low-cost approach of cough analysis could be useful in point-of-care settings and resource-limited areas. However, the limited size of the dataset and lack of external validation may limit the generalizability of the results.	**Advantages:** Automated analysis **Disadvantages:** High cost
Forecasting COVID-19 via Registration Slips of Patients using	aims to develop a deep learning model to forecast COVID-19 using registration slips of patients. The study uses a dataset of 414,557 registration slips from a hospital in Pakistan and evaluates the model's performance using metrics such as accuracy, sensitivity, and specificity. The study's	**Advantages:** .Early identification of high-risk patients **Disadvantages:** Bias in data

ResNet-101 and Performance Analysis and Comparison	non-invasive approach and use of readily available data could be useful for early detection and prevention of COVID-19. However, the lack of external validation and generalizability to other healthcare settings may limit the study's broader impact.	
Prediction of COVID-19 Spreading Using Support Vector Regression and Susceptible Infectious Recovered Model	proposes a predictive model to forecast the spread of COVID-19 using a combination of support vector regression and a susceptible infectious recovered model. The study uses a dataset of daily COVID-19 confirmed cases from March to August 2020 in Iran and evaluates the model's performance using metrics such as mean absolute error, mean squared error, and R-squared. The study's prediction model could be useful for resource allocation and planning for COVID-19 control and prevention. However, the limited dataset and lack of external validation may limit the generalizability of the results.	**Advantages:** 1. Improved accuracy 2. Automation **Disadvantages:** More dependence of data, lack of human expertise and lack of transparency
Severity and Consolidation Quantification of COVID-19 from CT Images Using Deep Learning Based on Hybrid Weak Labels	proposes a deep learning-based system for quantifying the severity and consolidation of COVID-19 from CT images. The study uses a hybrid weak label approach to train the deep learning model using a limited amount of labeled data and a larger set of weakly labeled data. The system's performance is evaluated using various metrics such as the Dice score, specificity, and sensitivity. The study's proposed system could aid radiologists and clinicians in diagnosing and monitoring COVID-19 patients using CT scans. However, the limited sample size and the lack of external validation may limit the generalizability of the results.	**Advantages:** Improved understanding of disease mechanisms **Disadvantages:** Lack of human Expertise.

Coronavirus Disease (COVID19) Global Prediction Using Hybrid Artificial Intelligence Method of ANN Trained with Grey Wolf Optimizer	The study combines machine learning algorithms such as decision tree, support vector regression, and linear regression to forecast the number of COVID-19 cases globally. The study uses a dataset of confirmed COVID-19 cases and evaluates the model's performance using metrics such as mean absolute error, root mean squared error, and correlation coefficient. The proposed hybrid AI model could be useful for policymakers and healthcare professionals in making informed decisions and allocating resources for COVID-19 prevention and control. However, the lack of external validation and the limited sample size may limit the generalizability of the results.	**Advantages:** 1.Early detection 2. Fast Detection 3. Low Cost Implementation **Disadvantages:** Large data to be handled.
A Support Vector Machine Classification of Computational Capabilities of 3D Map on Mobile Device for Navigation Aid	presents a support vector machine (SVM) based approach to classify the computational capabilities of 3D maps on mobile devices for navigation aid. The authors propose a set of features such as the number of vertices and triangles, surface area, and volume of the 3D map to train the SVM classifier. The study evaluates the proposed approach on a dataset of 3D maps of different areas and the results show that the SVM classifier achieved high accuracy in classifying the computational capabilities of 3D maps on mobile devices. The proposed approach could be useful in developing mobile navigation aid applications that can optimize the computational capabilities of 3D maps to improve navigation performance. However, the study's limitations include the small sample size and the lack of comparison with other classification methods.	**Advantages:** 1. Fast Prediction 2. Easy Analysis **Disadvantages:** Low Data Quality

3.Overview of proposed project

a. Motivation

Unavailability or saturation of the ICU may be associated with the fatality of COVID-19. Prioritizing the patients for hospitalization and intensive care may be critical for reducing the fatality of COVID-19. This leads the major point for our motivation for implementing this project.

b. Aim

The aim of this project is to provide ease in process of admitting COVID-19 patients for ICU beds based upon their severity levels and make this process a “Doctor-free” in order to reduce the stress levels of Doctors.

c. Objectives

The main objective is to develop a predictive model that can accurately identify which confirmed COVID-19 cases are likely to require ICU admission. The research aims to analyse various factors, such as temperature difference, age, blood pressure, heart rate, respiratory rate, oxygen saturation, and a few other attributes, to determine the most significant predictors of ICU admission. The goal is to create a model that can be used by healthcare professionals to identify high-risk patients early and allocate resources more efficiently, ultimately improving patient outcomes and reducing the burden on healthcare systems.

d. Development tools and methodologies to be used

Development tools : Python, Google collab(IDE)

Methodology:

1) Download Kaggle Sirio Libanes ICU Prediction dataset.

2) Understand the attributes of the datasets and load.

3) Preprocess the data.

4) Train using binary classification models

- **Logistic regression**
- **Gaussian Naïve Bayes**
- **SGD classifier (Stochastic gradient descent)**
- **XGB regressor (Extreme Gradient Boosting)**

5) Predict the results using the above machine learning techniques.

6) Compare the performances of individual model and find the accurate results.

4.Requirements specification of the proposed system

a. Functional requirements specification

Data Analysis

Analyzing the data based upon temperature difference, age, blood pressure, heart rate, respiratory rate, oxygen saturation, and a few other attributes.

Prediction Algorithms

The prediction algorithm should be optimized to produce accurate and reliable predictions for the number of ICU seats needed.

Prediction scoring

The prediction system must be able to predict new cases based on the model chosen and generate a accurate result of severity to confirm admission in ICU.

b. Non-functional requirements specification

- Reliability: The system must be reliable, with a high level of availability and minimal downtime. This is especially important in a healthcare setting, where delays or system failures can have serious consequences.

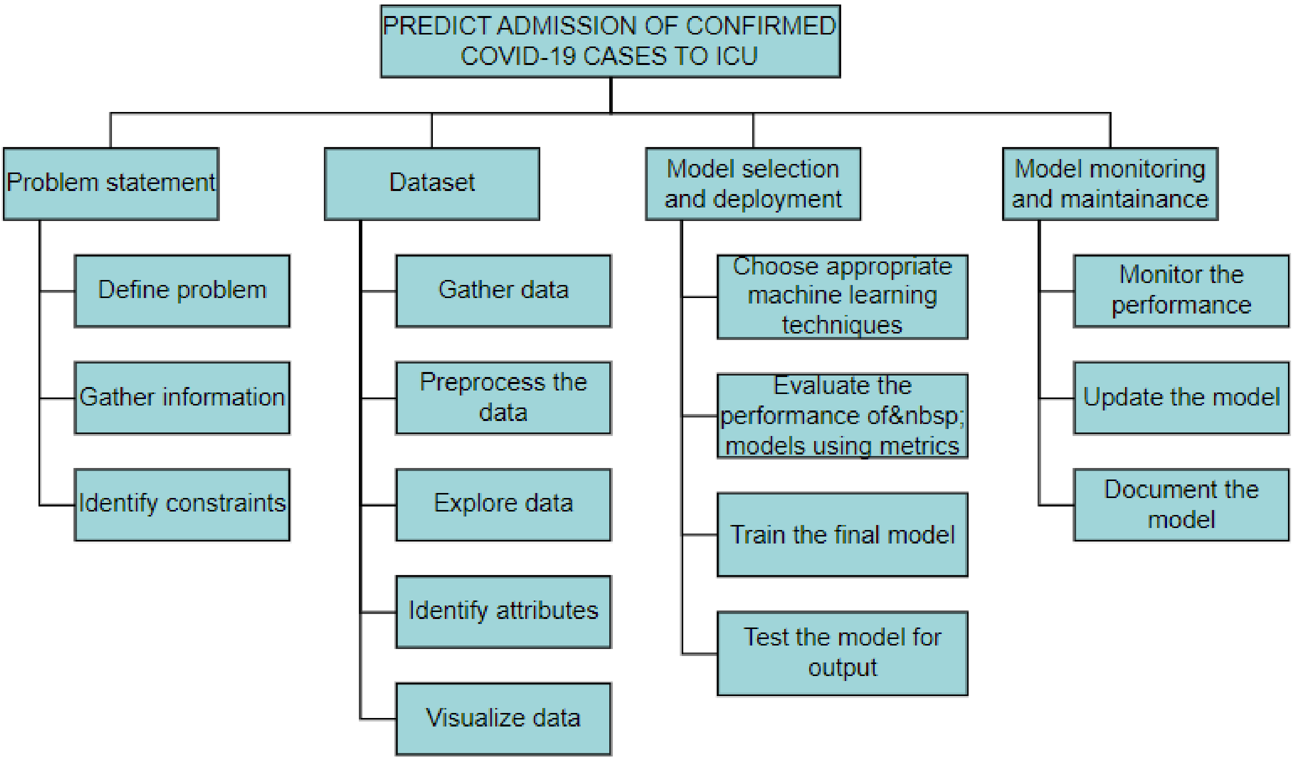
- Performance: The system must be able to handle large datasets and perform prediction tasks in a timely and efficient manner, with response times that meet user expectations.
- Maintainability: The system must be designed to be easy to maintain and update, with clear documentation and well-organized code.
- Scalability: The system must be able to scale up or down as needed to accommodate changes in data volume or user demand.
- Usability: The system must be designed to be user-friendly and easy to navigate, with clear and intuitive interfaces for healthcare professionals and patients.
- Interoperability: The system must be able to exchange data with other healthcare systems and technologies, using common standards and protocols.

c. Design constraints, if any

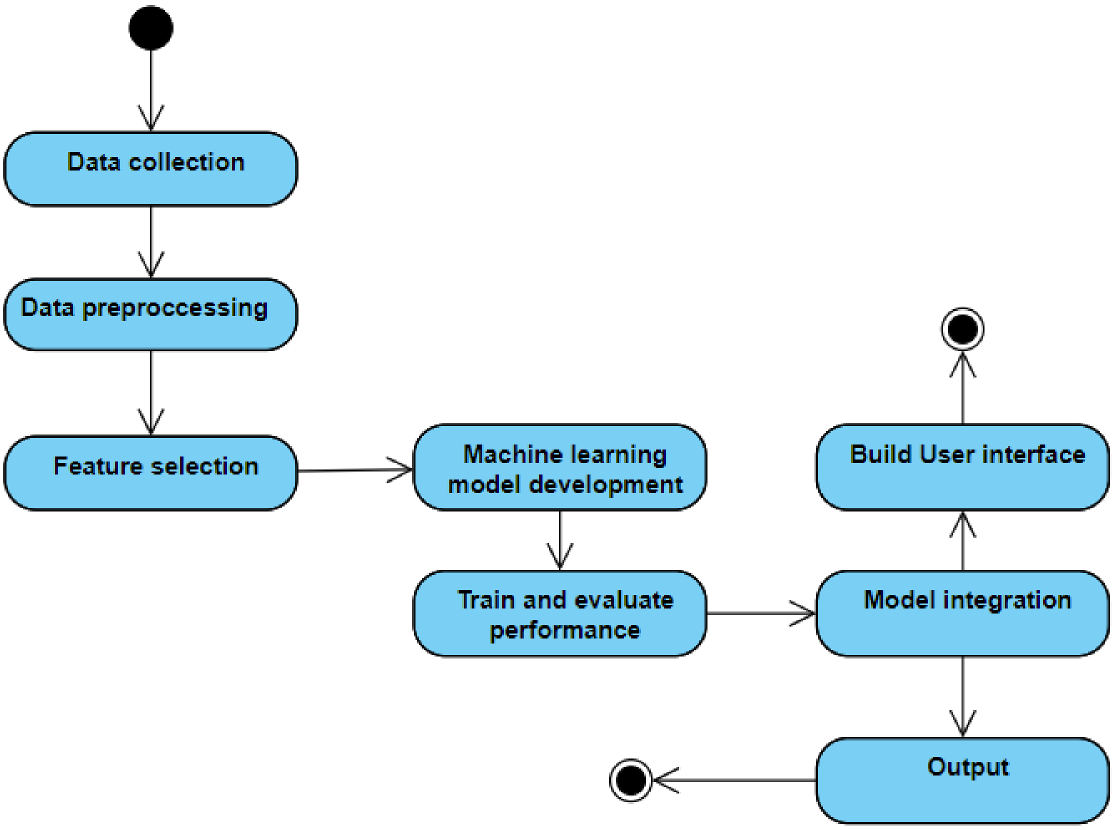
- Performance Requirements: The system must be designed to handle large datasets and perform prediction tasks in a timely and efficient manner. This may require optimizing algorithms, using parallel computing, or selecting appropriate hardware and software resources.
- Usability and User Experience: The system must be designed to be user-friendly and easy to navigate, with clear and intuitive interfaces for healthcare professionals and patients.
- Accessibility: The system must be designed to be accessible to all users, including those with disabilities or who use assistive technologies.

5.Project plan

a. Work breakdown structure (WBS) and/or



b. Activity diagram



c.Team Members (max. 3) and their roles/responsibilities

A VISVASS REDDI

- Abstract, Introduction, Literature survey, Implementation

PANASA YAMINI RAMA PAVANI

- Project plan, High-level &detailed design, Implementation.

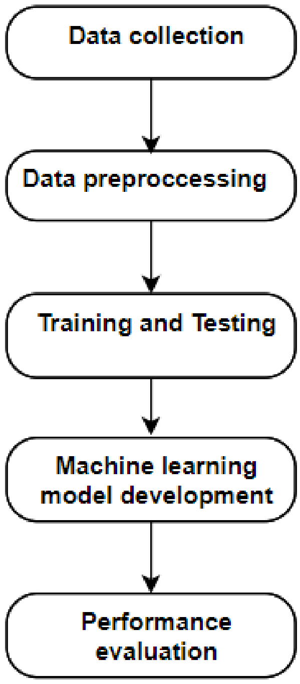
KATKAM SHANMUKH AKUL

- Overview, Requirements specification, Implementation.

6. High-level design and modules description

- Data collection:

Load patient's dataset downloaded from Kaggle. Understand the data with 231 columns.



- Data pre-processing:

Clear the null values in the data set. Cleaning of the dataset is very important for any dataset which helps the model to train a better way.

First change the column values in string type to numerical type. Fill the null values with the mean values of the column. This can be done by using imputer method in sklearn module.

- Training and Testing:

Divide the dataset into train and testing parts in 7:3 ratio for train and test data using train test split method in sklearn module.

- Machine learning model development:

Train the binary classification models.

We use 4 algorithms to train the dataset.

- Performance evaluation

  Evaluate the model using metrics like accuracy and confusion matrix.

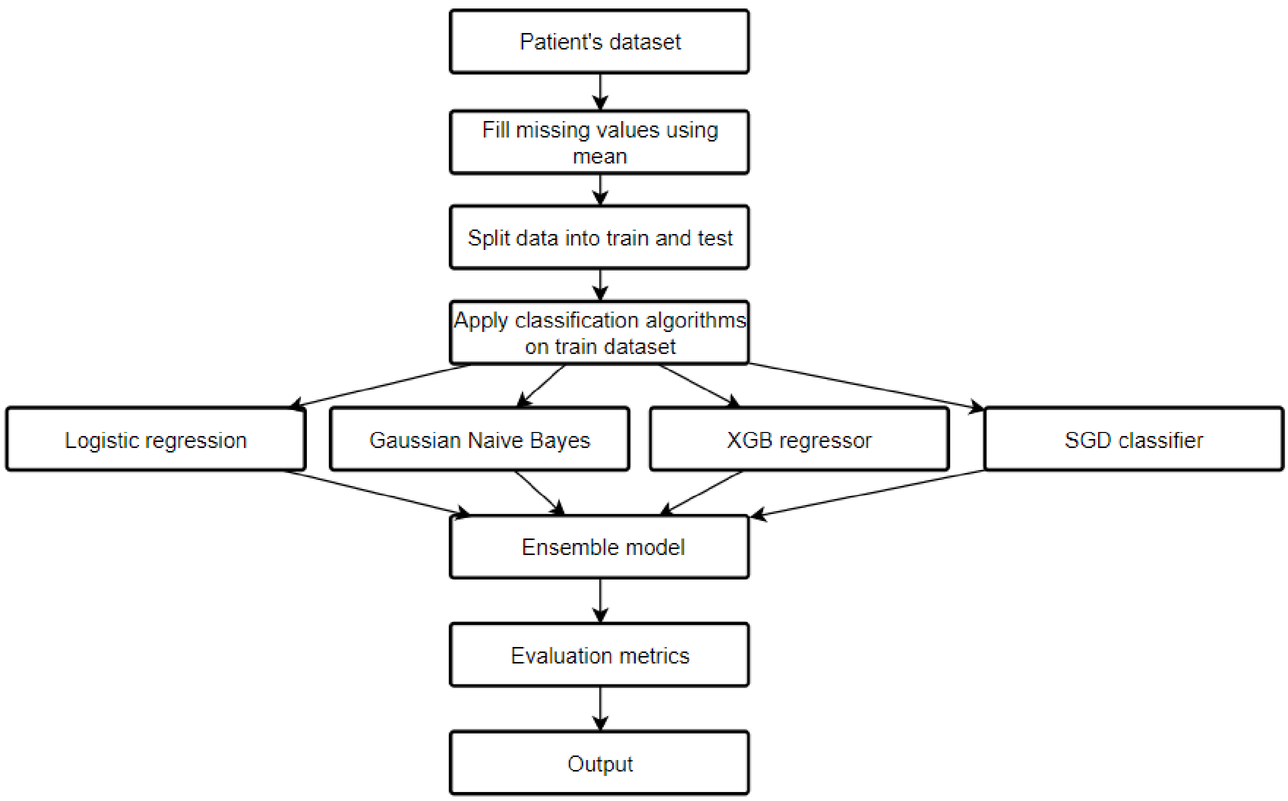
  Visualize ROC curve of the model which require two parameters (model predictions of test data part and original test values).

  Compare the performances of individual model.

7. Detailed design including database design, user interface design

The below diagram depicts the flow of project and detailed design of the process including the algorithms used for training.

There's no database requirement for the project as of now.



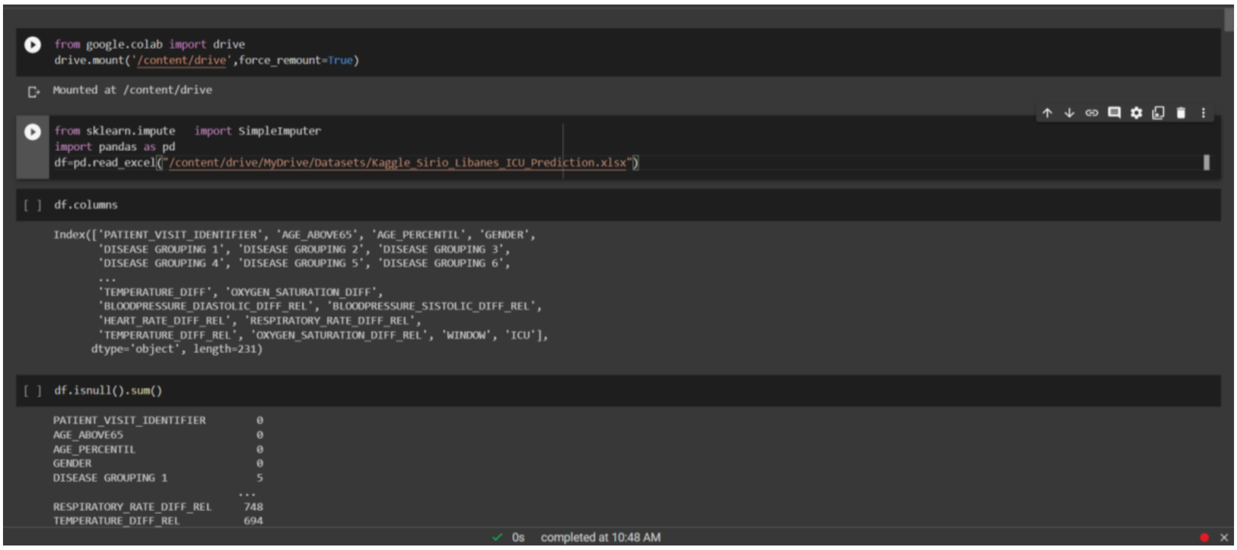
User Interface design:

8. Implementation

We load the dataset available in Kaggle from google drive

We understand the data and the attributes.

Then we count null value of each column in the dataset.



We now clear the null values from the dataset. Since cleaning of the dataset is very important for any dataset that helps model to train in a better way.

First, change the column values in string type to numerical type. Then, fill null values with mean values of the column. This can be done using imputer method in sklearn module. Then, we check the null values in the dataset.

We can see that there are no null vakues and each column is of datatype integer in below fig.

```
# Data Preparation
df['AGE_PERCENTIL'] = df['AGE_PERCENTIL'].str.replace('Above ','').str.extract(r'(.+?)th')
df['WINDOW'] = df['WINDOW'].str.replace('ABOVE_12','12-more').str.extract(r'(.+?)-')
```

```
# Missingness as features
df['row_missingness'] = df.isnull().sum(axis=1)
```

```
# Mean imputation
mean_impute  = SimpleImputer(strategy='mean')
imputed_data = mean_impute.fit_transform(df)
imputed_data = pd.DataFrame(imputed_data, columns = df.columns)
```

```
imputed_data.isnull().sum()
```

```
PATIENT_VISIT_IDENTIFIER      0
AGE_ABOVE65                   0
AGE_PERCENTIL                 0
GENDER                        0
DISEASE GROUPING 1            0
                             ..
TEMPERATURE_DIFF_REL          0
OXYGEN_SATURATION_DIFF_REL    0
WINDOW                        0
```

✓ 0s completed at 10:48 AM

Dividing dataset into train and testing parts. We divide data in 7:3 ratio for train and test data using train-test method in sklearn module.

```
target=["ICU"]
un=["row_missingness"]
numericals=list(set(imputed_data.columns.values)-set(target)-set(un))
len(numericals)
```

```
230
```

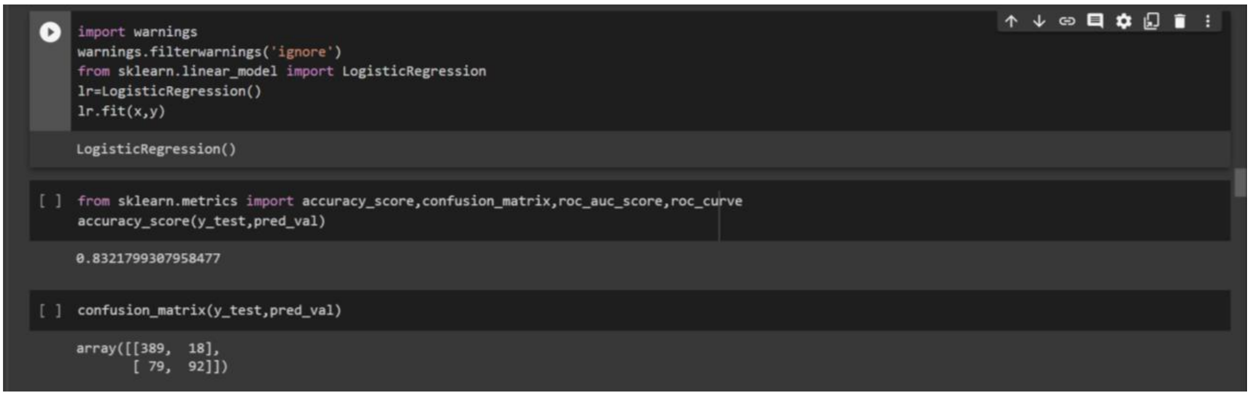
```
new_df=imputed_data[numericals]
new_df.shape
tar=imputed_data[target]
tar.shape
from sklearn.model_selection import train_test_split
x,x_test,y,y_test=train_test_split(new_df,tar,test_size=0.3)
x.shape
```

```
(1347, 230)
```

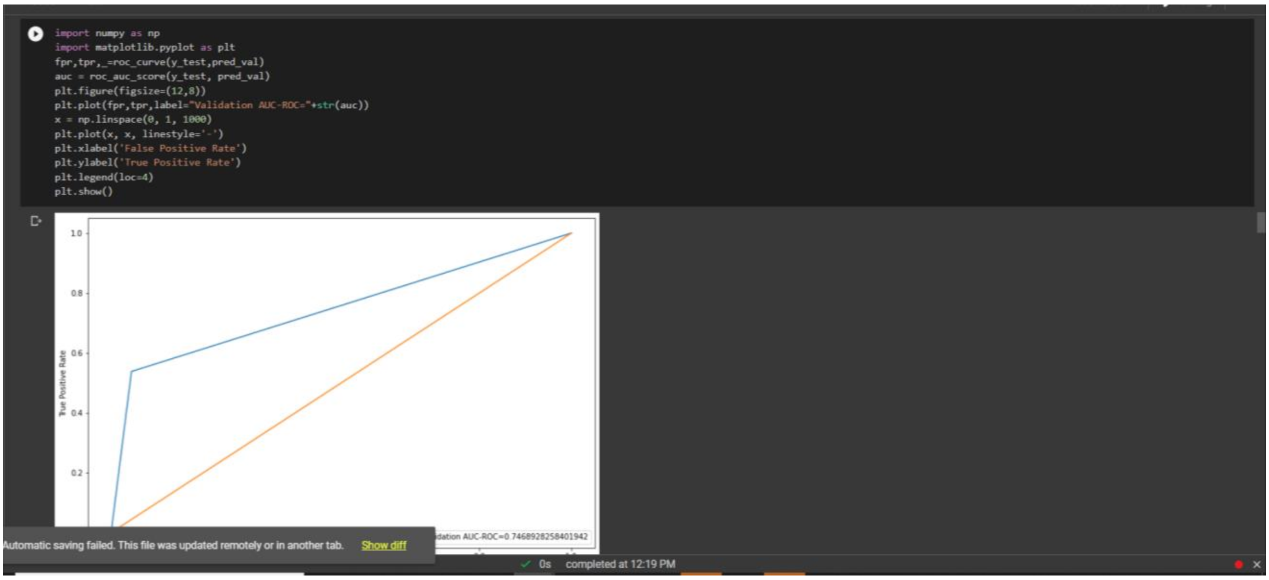
Train the binary classification models now. We are using 4 algorithms to train the data.

Logistic regression:

We create model based on logistic regression and we train the model with training data. Then we evaluate the model using metrics. Accuracy of the model is reported as 83%

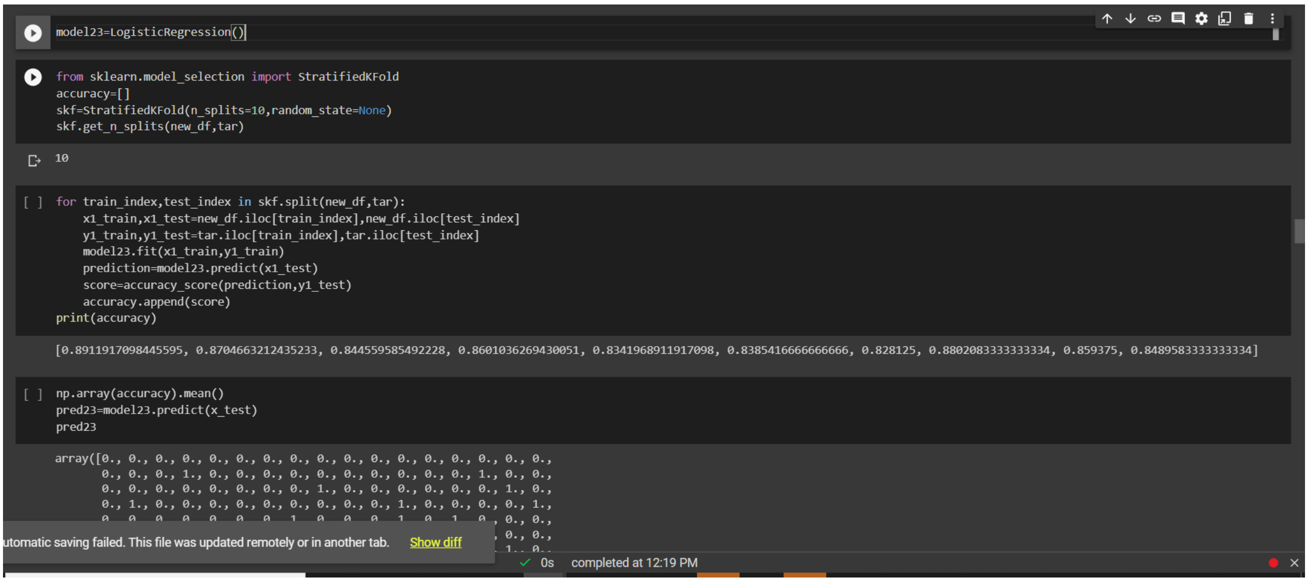


Visualising the ROC curve of the model using matplotlib library. For visualising ROC curve two parameters are required. Model predictions of test data part and original test values.

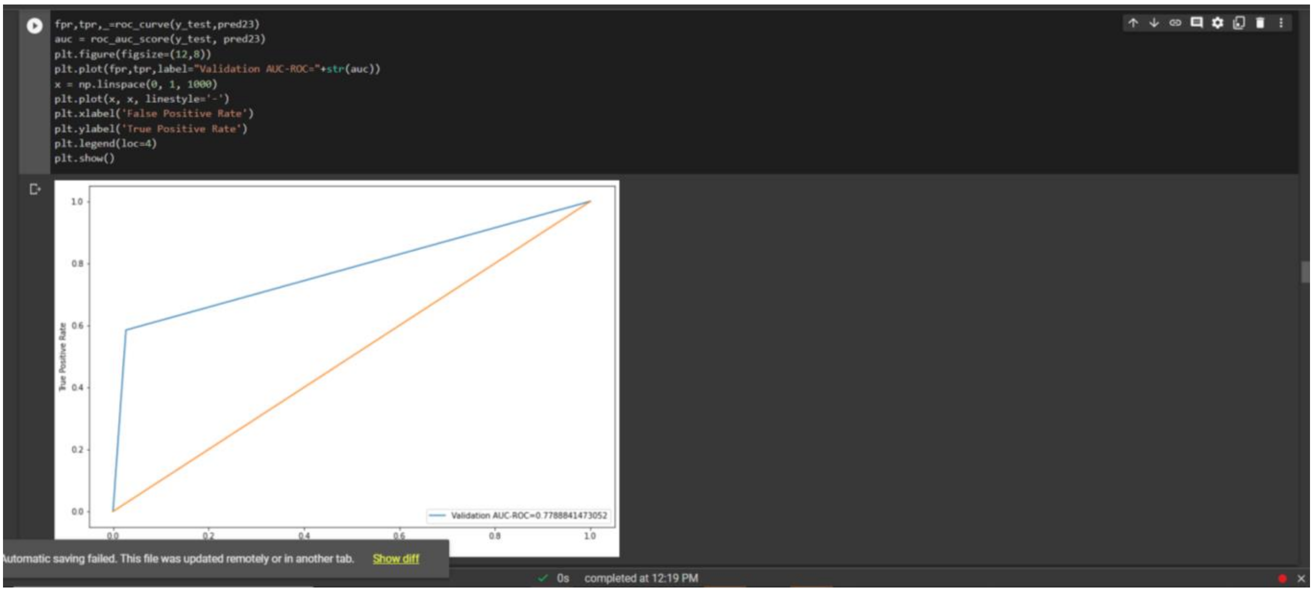


Logistic regression with stratified k fold

The accuracy of each fold is shown and the final accuracy of the model can be taken as the mean of array.



Evaluation metrics



Gaussian NB

We create a model based on Gaussian NB and we train the model with training data. We evaluate the model using metrics.

Accuracy of model is reported as 77%

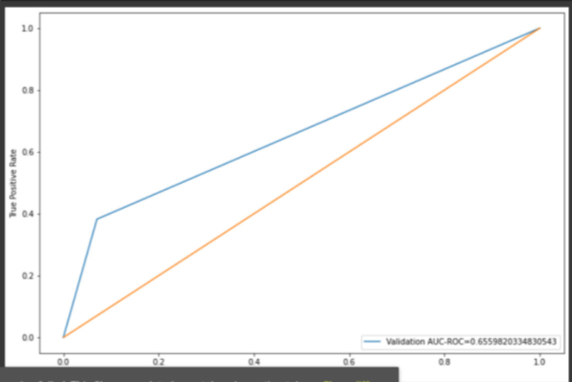
```
[ ] from sklearn.model_selection import train_test_split
    x,x_test,y,y_test=train_test_split(new_df,tar,test_size=0.4)

    from sklearn.naive_bayes import GaussianNB
    nb=GaussianNB()
    nb.fit(x,y)

    GaussianNB()

[ ] from sklearn.metrics import accuracy_score

[ ] accuracy_score(y_test,y_pred1)

    0.7753246753246753

[ ] confusion_matrix(y_test,y_pred1)

    array([[514,  39],
           [134,  83]])
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

0s completed at 12:19 PM

ROC curve:

```
fpr,tpr,_=roc_curve(y_test,y_pred1)
auc = roc_auc_score(y_test, y_pred1)
plt.figure(figsize=(12,8))
plt.plot(fpr,tpr,label="Validation AUC-ROC="+str(auc))
x = np.linspace(0, 1, 1000)
plt.plot(x, x, linestyle='-')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend(loc=4)
plt.show()
```

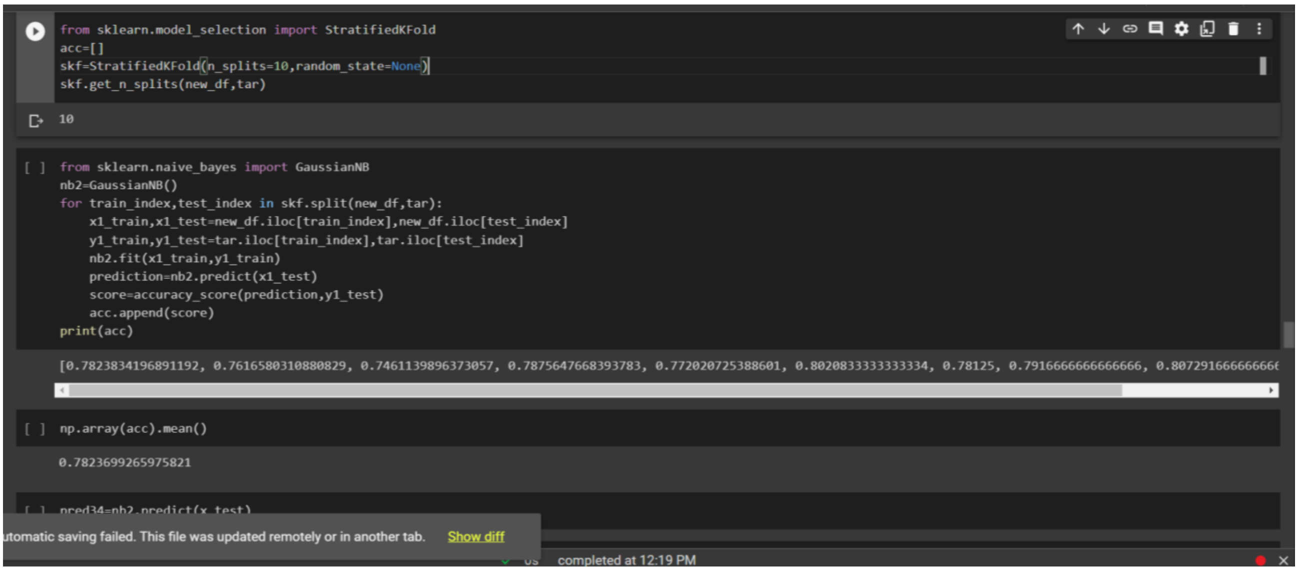


Automatic saving failed. This file was updated remotely or in another tab. Show diff

0s completed at 12:19 PM

Gaussian NB with stratified k-fold

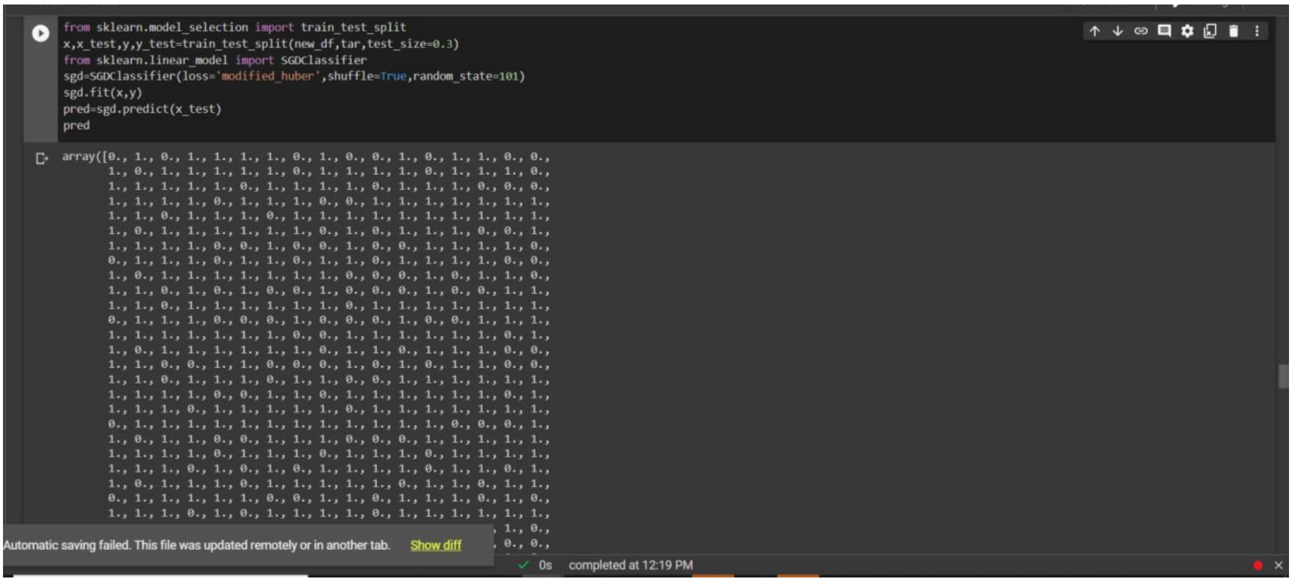
Accuracy and confusion matrix as are follows:

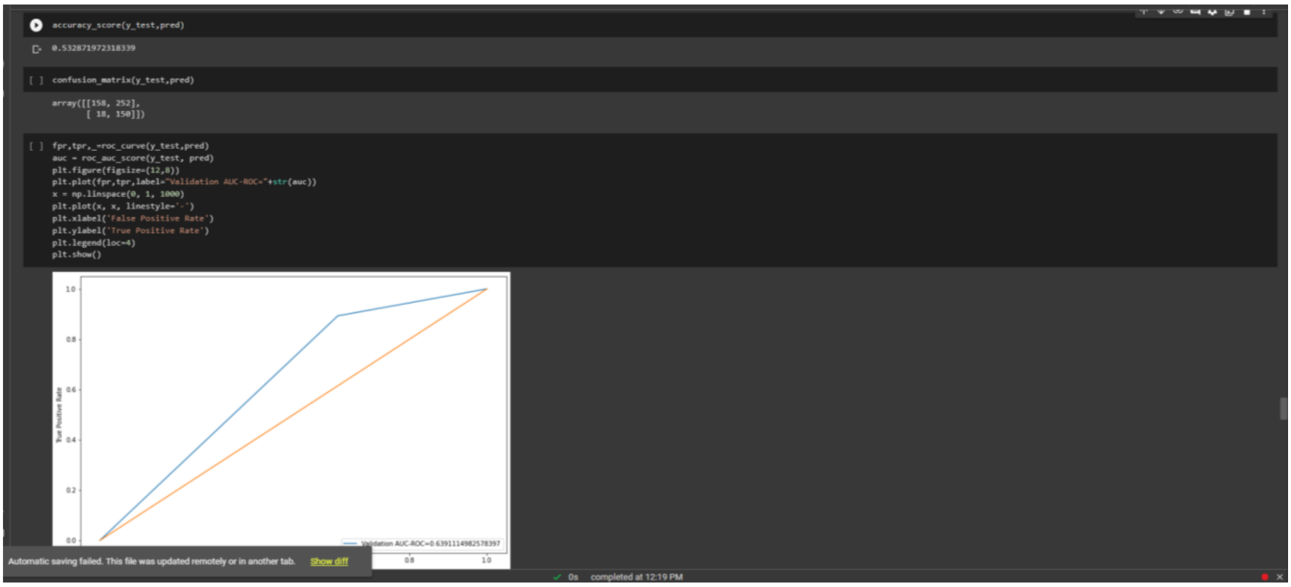


SGD classifier

We create a model based on SGD classifier and we train the model with training data. We evaluate the model using metrics.

Accuracy if model is reported as





SGD classifier with stratified k-fold

```
from sklearn.model_selection import StratifiedKFold
acc2=[]
skf=StratifiedKFold(n_splits=10,random_state=None)
skf.get_n_splits(new_df,tar)
```

```
10
```

```
from sklearn.linear_model import SGDClassifier
sgd2=SGDClassifier(loss='modified_huber',shuffle=True,random_state=101)
for train_index,test_index in skf.split(new_df,tar):
    x1_train,x1_test=new_df.iloc[train_index],new_df.iloc[test_index]
    y1_train,y1_test=tar.iloc[train_index],tar.iloc[test_index]
    sgd2.fit(x1_train,y1_train)
    prediction=nb2.predict(x1_test)
    score=accuracy_score(prediction,y1_test)
    acc2.append(score)
print(acc2)
saved_model6=pickle.dumps(sgd2)
```

```
[0.7823834196891192, 0.7616580310880829, 0.7461139896373057, 0.8031088082901554, 0.7823834196891192, 0.8072916666666666, 0.796875, 0.796875, 0.8125, 0.7916666666666666]
```

```
np.array(acc2).mean()
```

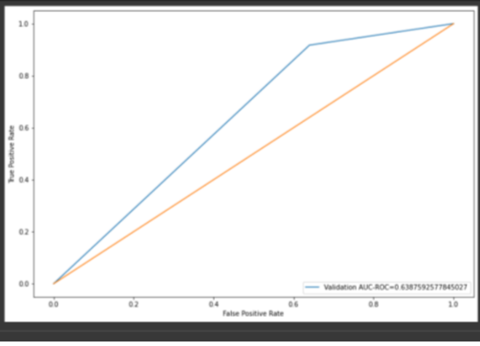
```
0.7880856001727117
```

```
pred45=sgd2.predict(x_test)
```

```
confusion_matrix(y_test,pred45)
```

```
array([[156, 277],
       [ 12, 133]])
```

```
fpr,tpr,_=roc_curve(y_test,pred45)
auc = roc_auc_score(y_test, pred45)
plt.figure(figsize=(12,8))
plt.plot(fpr,tpr,label="Validation AUC-ROC="+str(auc))
x = np.linspace(0, 1, 1000)
plt.plot(x, x, linestyle='-')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend(loc=4)
plt.show()
```



XGB regressor

We create a model based on XGB regressor and we train the model with training data. Accuracy of the model is reported as 89%

```
from sklearn.model_selection import train_test_split
x,x_test,y,y_test=train_test_split(new_df,tar,test_size=0.4)
from xgboost import XGBClassifier
model7=XGBClassifier()
model7.fit(x,y)
```

```
XGBClassifier()
```

```
pred7=model7.predict(x_test)
```

```
accuracy_score(pred7,y_test)
```

```
0.8909090909090909
```

```
confusion_matrix(y_test,pred7)
```

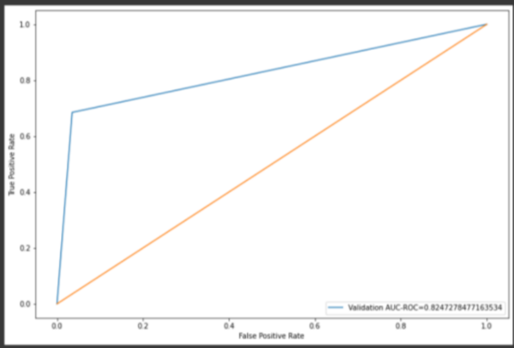
```
array([[547,  20],
       [ 64, 139]])
```

```
fpr,tpr,_=roc_curve(y_test,pred7)
```

utomatic document saving has been pending for 2 minutes. Reloading may fix the problem. Save and reload the page.

ROC curve:

```
plt.figure(figsize=(12,8))
plt.plot(fpr,tpr,label="Validation AUC-ROC="+str(auc))
x = np.linspace(0, 1, 1000)
plt.plot(x, x, linestyle='-')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend(loc=4)
plt.show()
```



Automatic document saving has been pending for 2 minutes. Reloading may fix the problem. Save and reload the page.

<u>XGB regressor with stratified k-fold</u>

```
from sklearn.model_selection import StratifiedKFold
acc3=[]
skf=StratifiedKFold(n_splits=10,random_state=None)
skf.get_n_splits(new_df,tar)
```

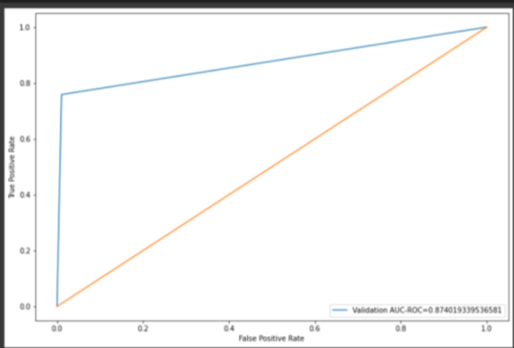
```
10
```

```
from xgboost import XGBClassifier
model8=XGBClassifier()
for train_index,test_index in skf.split(new_df,tar):
    x1_train,x1_test=new_df.iloc[train_index],new_df.iloc[test_index]
    y1_train,y1_test=tar.iloc[train_index],tar.iloc[test_index]
    model8.fit(x1_train,y1_train)
    prediction=model8.predict(x1_test)
    score=accuracy_score(prediction,y1_test)
    acc3.append(score)
print(np.array(acc3).mean())
```

```
0.8763493091537132
```

```
pred56=model8.predict(x_test)
```

```
auc = roc_auc_score(y_test, pred56)
plt.figure(figsize=(12,8))
plt.plot(fpr,tpr,label="Validation AUC-ROC="+str(auc))
x = np.linspace(0, 1, 1000)
plt.plot(x, x, linestyle='-')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend(loc=4)
plt.show()
```



Automatic document saving has been pending for 4 minutes. Reloading may fix the problem. Save and reload the page.

9. RESULTS AND ANALYSIS

We can ensemble all models into one stack model and we can train model by using the dataset. Create and run ensemble model. We can see the comparison of individual models performance and stacked model by box plot and bar plot. From the comparison we conclude stacked model have high performance.

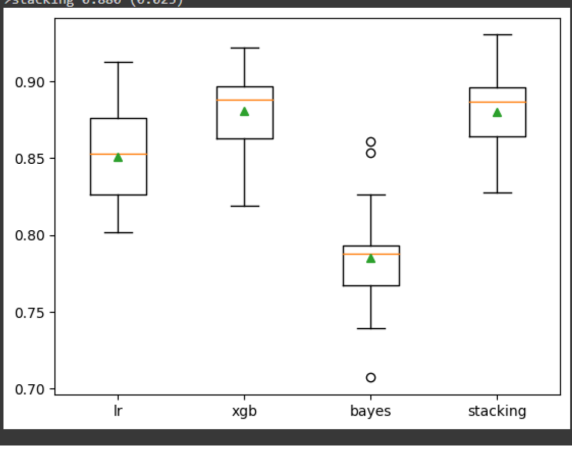
```
[66] from sklearn.ensemble import StackingClassifier
     from sklearn.model_selection import RepeatedStratifiedKFold
     from sklearn.model_selection import cross_val_score
     def get_stacking():
       # define the base models
       level0 = list()
       level0.append(('lr', LogisticRegression()))
       level0.append(('xgb', XGBClassifier()))
       level0.append(('bayes', GaussianNB()))
       # define meta learner model
       level1 = LogisticRegression()
       # define the stacking ensemble
       model = StackingClassifier(estimators=level0, final_estimator=level1, cv=5)
       return model
```

```
[67] def get_models():
       models = dict()
       models['lr'] = LogisticRegression()
       models['xgb'] = XGBClassifier()
       models['bayes'] = GaussianNB()
       models['stacking'] = get_stacking()
       return models
```

```
[68] def evaluate_model(model, X, y):
       cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
       scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1, error_score='raise')
       return scores
```

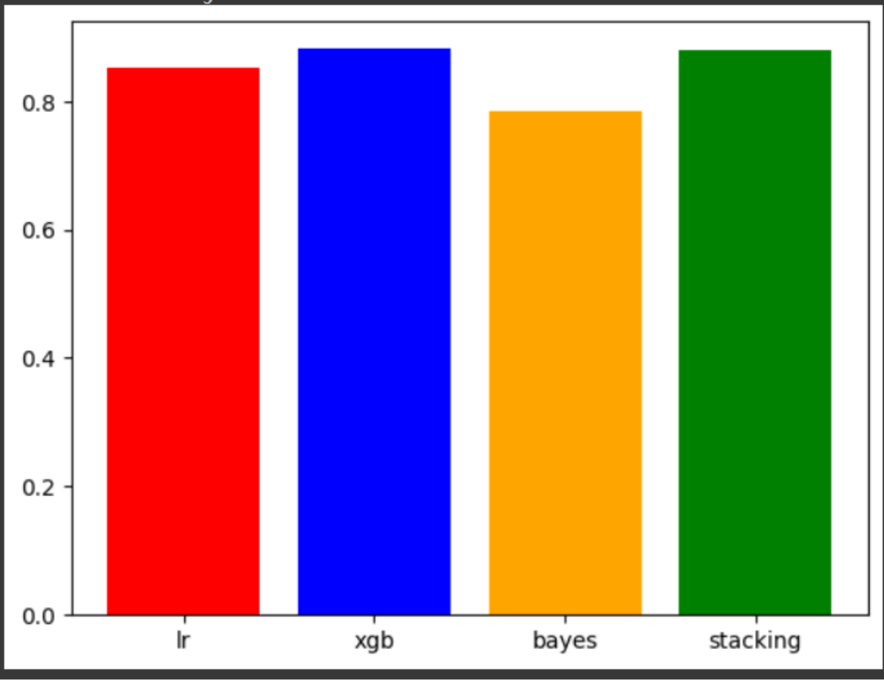
```
from sklearn.model_selection import train_test_split
x,x_test,y,y_test=train_test_split(new_df,tar,test_size=0.4)
models = get_models()
# evaluate the models and store results
results, names = list(), list()
for name, model in models.items():
  scores = evaluate_model(model, x, y)
  results.append(scores)
  names.append(name)
  print('>%s %.3f (%.3f)' % (name, np.mean(scores), np.std(scores)))
# plot model performance for comparison
plt.boxplot(results, labels=names, showmeans=True)
plt.show()
```

```
>lr 0.851 (0.032)
>xgb 0.881 (0.027)
>bayes 0.785 (0.031)
>stacking 0.880 (0.025)
```



```
b=[]
for i in range(4):
    b.append(results[i].mean())
b
plt.bar(names,b,color=["red","blue","orange","green"])
```

<BarContainer object of 4 artists>



Conclusion

In conclusion, predicting the admission of confirmed COVID-19 cases to the ICU is a critical task that can aid in the effective management of resources and ultimately improve patient outcomes. With the use of advanced machine learning algorithms and predictive models, healthcare providers can leverage data from various sources, including patient demographics, clinical characteristics, and laboratory findings, to identify patients at higher risk of ICU admission. This information can help healthcare providers prioritize resources and provide more targeted interventions to patients who are most in need. However, it is important to note that these models are not infallible, and healthcare providers should use their clinical judgment and expertise in conjunction with these predictive tools. Additionally, ongoing research and refinement of these models are necessary to ensure their accuracy and applicability across different populations and healthcare settings. Ultimately, the use of predictive models can enhance the quality of care for COVID-19 patients and help mitigate the impact of this pandemic on healthcare systems worldwide.

Future scope

We can integrate the backend running part with API and implement a android application which makes easy for hospital management which can be finished with one click from mobile application and desktop and increase the accuracy of the model by creating a best stack model.

Annexure-1: References

[1] M. Frid-Adar, R. Amer, O. Gozes, J. Nassar and H. Greenspan, "COVID-19 in CXR: From Detection and Severity Scoring to Patient Disease Monitoring," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 6, pp. 1892-1903, June 2021, doi: 10.1109/JBHI.2021.3069169.

[2] C. Li et al., "Classification of Severe and Critical Covid-19 Using Deep Learning and Radiomics," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 12, pp. 3585-3594, Dec. 2020, doi: 10.1109/JBHI.2020.3036722.

[3] N. Darapaneni et al., "A Machine Learning Approach to Predicting Covid-19 Cases Amongst Suspected Cases and Their Category of Admission," 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), RUPNAGAR, India, 2020, pp. 375-380, doi: 10.1109/ICIIS51140.2020.9342658.

[4] L. Famiglini, G. Bini, A. Carobene, A. Campagner and F. Cabitza, "Prediction of ICU admission for COVID-19 patients: a Machine Learning approach based on Complete Blood Count data," 2021 IEEE 34th International Symposium on

Computer-Based Medical Systems (CBMS), Aveiro, Portugal, 2021, pp. 160-165, doi: 10.1109/CBMS52027.2021.00065.
[5] V. Bhadana, A. S. Jalal and P. Pathak, "A Comparative Study of Machine Learning Models for COVID-19 prediction in India," 2020 IEEE 4th Conference on Information & Communication Technology (CICT), Chennai, India, 2020, pp. 1-7, doi: 10.1109/CICT51604.2020.9312112.
[6] J. Rodríguez et al., "A Covid-19 Patient Severity Stratification using a 3D Convolutional Strategy on CT-Scans," 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), Nice, France, 2021, pp. 1665-1668, doi: 10.1109/ISBI48211.2021.9434154.
[7] N. Darapaneni et al., "COVID 19 Severity of Pneumonia Analysis Using Chest X Rays," 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), RUPNAGAR, India, 2020, pp. 381-386, doi: 10.1109/ICIIS51140.2020.9342702.
[8] R. Y. Wang, T. Q. Guo, L. G. Li, J. Y. Jiao and L. Y. Wang, "Predictions of COVID-19 Infection Severity Based on Co-associations between the SNPs of Co-morbid Diseases and COVID-19 through Machine Learning of Genetic Data," 2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2020, pp. 92-96, doi: 10.1109/ICCSNT50940.2020.9304990.
[9] T. Dan et al., "Machine Learning to Predict ICU Admission, ICU Mortality and Survivors’ Length of Stay among COVID-19 Patients: Toward Optimal Allocation of ICU Resources," 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Seoul, Korea (South), 2020, pp. 555-561, doi: 10.1109/BIBM49941.2020.9313292.
[10] A. Dhadge and G. Tilekar, "Severity Monitoring Device for COVID-19 Positive Patients," 2020 3rd International Conference on Control and Robots (ICCR), Tokyo, Japan, 2020, pp. 25-29, doi: 10.1109/ICCR51572.2020.9344386.

Annexure-2: Sample screenshots

Annexure-3: Sample source code listing

https://colab.research.google.com/drive/1U-IJk9p1SYHuZKf4-AS-wL0RFzO4Agl8?usp=sharing

```
#pip3 install -U imbalanced-learn
#pip install tf-nightly

# Import required libraries for GUI
from tkinter import *
from PIL import ImageTk, Image
from tkinter import filedialog
import pandas as pd
from tkinter import messagebox as msg

# Import required libraries for ML code
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Activation, Dropout, Flatten, Dense
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA, IncrementalPCA

class predict_ecg:
  def __init__(self,win):
```

```
        self.win=win
        self.file_name=""
        self.frame = Frame(self.win, width=400, height=400)
        self.frame.pack()
        self.frame.place(anchor='w', relx=0.5, rely=0.5)

# Create an object of tkinter ImageTk
        self.img = ImageTk.PhotoImage(Image.open("img.jpg"))

# Create a Label Widget to display the text or Image
        self.label = Label(self.frame, image = self.img)
        self.label.pack()

        self.label1 = Label(self.win,text="ICU ADMISSION REPORT", fg = "black",bg="white", font =
("Castellar",40,"bold")).place(x=600,y=60)
        self.label2 = Label(self.win,text="PREDICT ADMISSION OF CONFIRMED COVID-19 CASES TO ICU !",
fg = "blue",bg="white", font = ("Arial",15)).place(x=650,y=130)
        self.label3 = Label(self.win,text="Upload file", fg = "black",bg="white", font =
("Arial",18)).place(x=200,y=280)
        self.photo = PhotoImage(file = r"upload.png")
        self.photoimage = self.photo.subsample(4,4)
        self.button1 = Button(self.win, text="Upload", font = ("Arial",15),
fg="black",bg="white",image=self.photoimage, compound = LEFT, activebackground = "blue",
activeforeground="white", command = self.open_csv)
        self.button1.place(x=400,y=260)
#label4 = Label(win,text="Processing..Please wait for few minutes...", fg = "black",bg="white", font =
("Arial",14)).place(x=200,y=360)
        self.result = Label(self.win,text="", fg = "blue",bg="white", font = ("Arial",16))
        self.result.place(x=200,y=460)



    def open_csv(self):
        try:
            self.file_name = filedialog.askopenfilename(initialdir = '/Desktop', title = 'Select a CSV file', filetypes
= (('csv file','*.csv'), ('csv file','*.csv')))
            df1 = pd.read_csv(self.file_name)
            label4 = Label(win,text="Processing..Please wait for few minutes...", fg = "black",bg="white", font
= ("Arial",14)).place(x=200,y=360)
            df1 = pd.read_csv(self.file_name)
            df =
pd.read_csv('C:\\Users\\Pyr.Pavani\\Downloads\\Kaggle_Sirio_Libanes_ICU_Prediction.csv')
            if (len(df) == 0):
                msg.showinfo('No Rows Selected', 'CSV has no rows')
            else:
                # Data Preparation
                df['AGE_PERCENTIL'] = df['AGE_PERCENTIL'].str.replace('Above ', '').str.extract(r'(.+?)th')
                df['WINDOW'] = df['WINDOW'].str.replace('ABOVE_12', '12-more').str.extract(r'(.+?)-')

                # Data Preparation for check
                df1['AGE_PERCENTIL'] = df1['AGE_PERCENTIL'].str.replace('Above ', '').str.extract(r'(.+?)th')
                df1['WINDOW'] = df1['WINDOW'].str.replace('ABOVE_12', '12-more').str.extract(r'(.+?)-')

                # Missingness as features
                df['row_missingness'] = df.isnull().sum(axis=1)

                # Missingness as features for check
                df1['row_missingness'] = df1.isnull().sum(axis=1)
```

```
        from sklearn.impute import SimpleImputer
        # Mean imputation
        mean_impute = SimpleImputer(strategy='mean')
        imputed_data = mean_impute.fit_transform(df)
        imputed_data = pd.DataFrame(imputed_data, columns=df.columns)
        imputed_data.isnull().sum()

        # Mean imputation for check
        mean_impute = SimpleImputer(strategy='mean')
        imputed_data1 = mean_impute.fit_transform(df1)
        imputed_data1 = pd.DataFrame(imputed_data1, columns=df1.columns)

        target = ["ICU"]
        un = ["row_missingness"]
        numericals = list(set(imputed_data.columns.values) - set(target) - set(un))
        numericals1 = list(set(imputed_data1.columns.values) - set(target) - set(un))

        new_df = imputed_data[numericals]
        new_df.shape

        new_df1 = imputed_data1[numericals1]
        new_df1.shape

        tar = imputed_data[target]
        tar.shape
        from sklearn.model_selection import train_test_split
        x, x_test, y, y_test = train_test_split(new_df, tar, test_size=0.3)
        import warnings
        warnings.filterwarnings('ignore')
        from sklearn.linear_model import LogisticRegression
        lr = LogisticRegression()
        lr.fit(x, y)
        pred_val = lr.predict(x_test)
        pred_val1 = lr.predict(new_df1)
        print(pred_val)

        x = pred_val1[0]
        if (x == 1):
          self.result.configure(text="RESULT: Patient have to be admitted to ICU")
          print("result is 1")
        elif (x == 0):
          self.result.configure(text="RESULT: Output is 0")
        else:
          self.result.configure(text="RESULT is unpredictable")

    except FileNotFoundError as e:
        msg.showerror('Error in opening file', e)


#Driver code
# Create an instance of tkinter window
win = Tk()
win.title("ICU report")
obj = predict_ecg(win)
# Define the geometry of the window
win.geometry("1000x5700")
win.configure(bg='white')

win.mainloop()
```

Annexure-4: Minutes of Meeting held

Meeting Name: Project of Predict admission of confirmed COVID-19 cases to ICU for Knowledge Management System course

Attendees: A Visvass Reddi, Panasa Yamini Rama Pavani, Katkam Shanmukh Akul

Agenda:

1. Review of project status
2. Updates on individual tasks

Minutes:

1. The meeting was called to order at once in every week.
2. Each attendee provided updates on their respective tasks, including any issues or concerns.
3. The group discussed the need for additional resources to complete certain tasks like implementing the GUI for the project.

Annexure-5: Weekly activity report of each team member

Week No	**19MIS0110**	**19MIS0120**	**19MIS0130**
1	Information gathering	Abstract, Introduction	Overview of existing system
2	Literature survey	Literature survey	Literature survey
3	Overview of project	Project plan	Requirements specification
4	UI design	Detailed design	High-level design
5	Implementation	Implementation	Implementation
6	Conclusion	Annexures	Result analysis