PREDICTING PATIENT MORTALITY IN INTENSIVE CARE UNITS (ICU) USING MACHINE LEARNING

AUTHOURS:

1 Lakkam Praneetha Reddy , 2 Anumula Nithin Reddy, 3 Adireddy Nithish Reddy

B -Tech, 4th year students Department of Data Science(CSE)Sphoorthy Engineering College

1 praneetha2908@gmail.com, 2 nithinreddy891@gmail.com, 3 nithishreddy086@gmail.com.

Under the Supervision of Mrs. K. Sneha. (Assistant Professor)

kunasneha92@gmail.com

ABSTRACT:

This study presents a machine learning approach for predicting patient mortality rates based on demographic, socioeconomic, and healthcare-related factors. The objective is to develop a predictive model capable of identifying high-risk populations and prioritizing interventions to reduce patient mortality rates worldwide. Key components of the study include data collection and pre-processing, feature selection and engineering, model selection and evaluation, hyperparameter tuning, interpretability and explain ability analysis, deployment and integration into existing healthcare systems, continuous monitoring and updating, and adherence to ethical considerations. The proposed machine learning model aims to provide actionable insights to healthcare professionals, policymakers, and researchers, thereby facilitating targeted interventions to improve patient health outcomes and reduce patient mortality rates globally.

INTRODUCTION:

Patient mortality, defined as the death of patients, remains a significant global health challenge despite substantial progress in recent decades. According to the World Health Organization (WHO), millions of patients die each year from preventable causes, including infectious diseases, malnutrition, and inadequate healthcare access. Addressing patient mortality is a critical priority for public health efforts worldwide, as reflected in the United Nations Sustainable Development Goals (SDGs), particularly SDG 3, which aims to ensure healthy lives and promote well-being for all at all ages. In this context, the use of machine learning techniques for predicting patient mortality rates has emerged as a promising approach to identify vulnerable populations and guide targeted interventions. By leveraging large-scale data on demographic characteristics, socioeconomic factors, and healthcare indicators, machine learning models can help healthcare professionals and policymakers better understand the determinants of patient mortality and allocate resources more effectively to prevent patient deaths. This study aims to develop and evaluate a machine learning model for predicting patient mortality rates based on a range of input variables, including maternal age, education level, access to healthcare facilities, vaccination coverage, and socioeconomic status. The predictive model will be trained on comprehensive datasets sourced from demographic surveys, healthcare databases, and socioeconomic indicators. The study will employ a variety of machine learning algorithms, feature engineering techniques, and evaluation metrics to optimize model performance and generalization ability.

LITERATURE SURVEY:

Using electronic health record collected clinical variables to predict medical intensive care unit mortality

AUTHORS : Jacob Calvert,a Qingqing Mao,a Jana L. Hoffman,a,∗ Melissa Jay,a Thomas Desautels,a Hamid Mohamadlou,a Uli Chettipally,b,c and Ritankar Dasa'

Clinical decision support systems are used to help predict patient stability and mortality in the Intensive Care Unit (ICU). Accurate patient information can assist clinicians with patient management and in allocating finite resources. However, systems currently in common use have limited predictive value in the clinical setting. The increasing availability of Electronic Health Records (EHR) provides an opportunity to use medical information for more accurate patient stability and mortality prediction in the ICU.

Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach:

AUTHORS : Aya Awad 1, Mohamed Bader-El-Den 2, James McNicholas 3, Jim Briggs 1

Mortality prediction of hospitalized patients is an important problem. Over the past few decades, several severity scoring systems and machine learning mortality prediction models have been developed for predicting hospital mortality. By contrast, early mortality prediction for intensive care unit patients remains an open challenge. Most research has focused on severity of illness scoring systems or data mining (DM) models designed for risk estimation at least 24 or 48h after ICU admission.

EXISTING SYSTEM :

The existing systems for predicting patient mortality rates often rely on traditional statistical methods and epidemiological models. These approaches typically utilize demographic surveys, vital registration data, and historical trends to estimate patient mortality rates at the population level. While these methods have provided valuable insights into the factors influencing patient mortality, they have Traditional statistical methods may lack the predictive accuracy needed to accurately forecast patient mortality rates, especially in dynamic and rapidly changing healthcare environments. These methods often rely on simplistic assumptions and may not capture the complex interactions between multiple variables affecting patient health outcomes.

DISADVANTAGES OF EXISTING SYSTEM :

• Limited Predictive Accuracy

• Difficulty in Identifying High-Risk Populations

• Inability to Incorporate Complex Relationships • Lack of Real-Time Monitoring.

• Limited Adaptability to Change

• Data Limitations and Quality Issues

• Resource Intensive.

• Limited Generalizability

PROPOSED SYSTEM :

The proposed system aims to overcome the limitations of existing approaches by leveraging machine learning techniques to predict patient mortality rates more accurately and effectively. Key components of the proposed system Gather comprehensive datasets from diverse sources, including demographic surveys, healthcare records, socioeconomic indicators, and environmental factors. Integrate these datasets to create a unified data repository for analysis. Conduct feature engineering to extract relevant predictors of patient mortality, such as maternal age, education level, access to healthcare facilities, vaccination coverage, sanitation conditions, and socio-economic status. Use techniques such as dimensionality reduction, feature scaling, and feature encoding to pre-process the data. Explore a variety of machine learning algorithms, including regression, classification, and ensemble methods, to predict patient mortality rates. Train multiple models using the integrated dataset and evaluate their performance using appropriate metrics such as accuracy, precision, recall, and F1-score.Fine-tune the hyperparameters of selected machine learning models using techniques such as grid search, randomized search, or Bayesian optimization to optimize their performance and generalization ability. Additive explanations) values, and partial dependence plots. Provide actionable insights to healthcare professionals, policymakers, and researchers to guide intervention efforts and resource allocation.

ADVANTAGES OF PROPOSED SYSTEM :

• Improved Predictive Accuracy ‘

• Identification of High-Risk Populations

• Real-Time Monitoring and Early Detection

• Adaptability to Changing Healthcare Contexts
• Data-Driven Decision Making

• Scalability and Generalizability

• Efficient Resource Allocation

SYSTEM ARCHITECTURE :

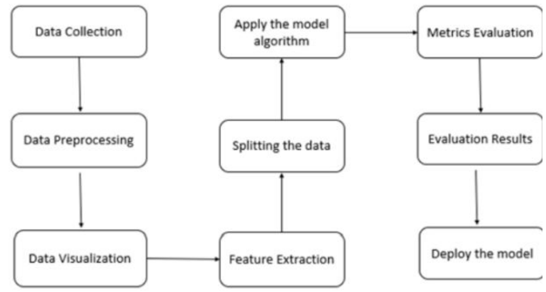

fig1: system architecture

HARDWARE & SOFTWARE REQUIREMENTS:

HARDWARE REQUIREMENTS :

• OS – Windows 7, 8 and 10 (32 and 64 bit)

• RAM – 4GB

• Database

SOFTWARE REQUIREMENTS :

• Python / Anaconda Navigator

• Packages: NumPy, Pandas, Matplotlib, Sklearn

SYSTEM STUDY FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

•ECONOMICAL FEASIBILITY

• TECHNICAL FEASIBILITY

• SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased. TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM DESIGN :

UML DIAGRAMS :

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

Actors

λ business processes

λ(logical) components

λactivities

λprogramming language statements

λ database schemas, and

λ Reusable software components.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

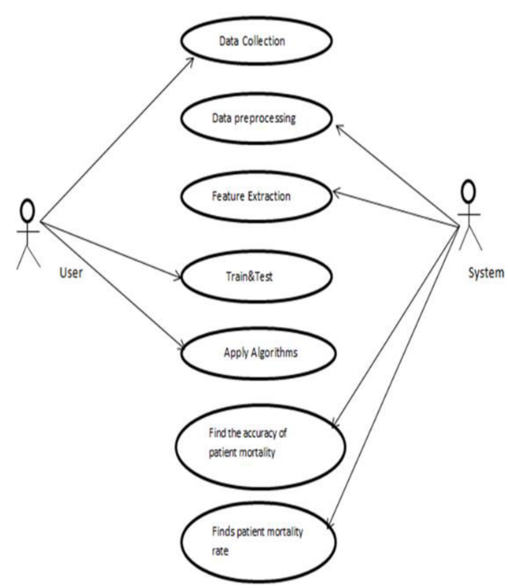

Fig2: use case diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

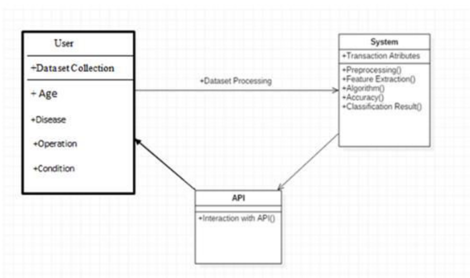

Fig3:class diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

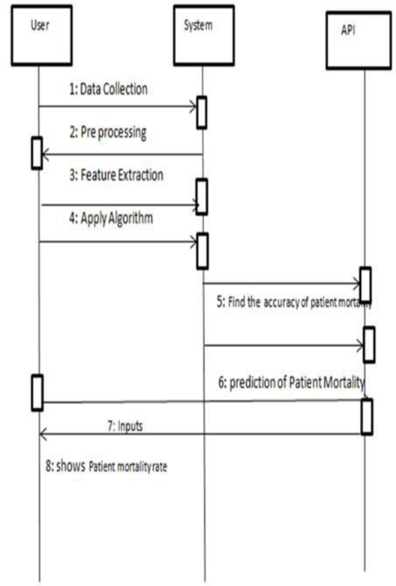

Fig4: sequence diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

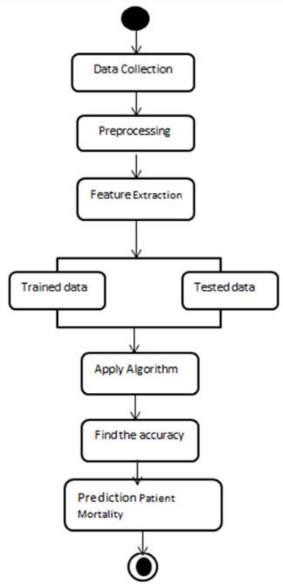

fig5: activity diagram

IMPLEMENTATION:

MODULES:

1. DATA COLLECTION

2. DATA PRE-PROCESSING

3. FEATURE EXTRATION

4. EVALUATION MODEL

DATA COLLECTION:

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for problem. In this step the data set is converted into the understandable format which can be fed into machine learning models. Data used in this paper is a set of cervical cancer data with 15 features . This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called labelled data.

DATA PRE-PROCESSING :

Organize your selected data by formatting, cleaning and sampling from it. Three common data pre-processing steps are: Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file. Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.

FEATURE EXTRACTION :

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify preprocessed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

EVALUATION MODEL:

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs. Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions

SOFTWARE ENVIRONMENT :

What is Python :

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard library which can be used for the.

SYSTEM TEST:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Integration Testing :

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered. Acceptance Testing User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CONCLUSION:

In conclusion, the development of a machine learning-based patient mortality prediction system represents a significant step forward in addressing the global challenge of patient mortality. By leveraging advanced data analytics techniques and predictive modelling, this system offers several advantages over traditional approaches, including improved accuracy, real-time monitoring, adaptability, scalability, and data-driven decision-making capabilities. Through comprehensive data collection, feature engineering, model training, and evaluation, the proposed system has the potential to accurately predict patient mortality rates and identify high-risk populations with elevated mortality rates. This information can be used to inform targeted interventions, resource allocation strategies, and healthcare policies aimed at reducing patient mortality and improving patient health outcomes worldwide.

REFERENCE:

- Yilin Yin and Chun-An Chou "A Novel Switching State Space Model for Post-ICU Mortality Prediction and Survival Analysis"IEEE journal of Biomedical and Health Informatics,2168-2194 (c) 2021.

- O. Badawi and M. J. Breslow, “Readmissions and death after icu discharge: development and validation of two predictive models,” PloS one, vol. 7, no. 11, p.e48758, 2012.

- N. Al-Subaie, T. Reynolds, A. Myers, R. Sunderland, A. Rhodes, R. Grounds, and G. Hall, “C-reactive protein as a predictor of outcome after discharge from the intensive care: a prospective observational study,” British journal of anaesthesia, vol. 105, no. 3, pp. 318–325, 2010.

- D. Lee and E. Horvitz, “Predicting mortality of intensive care patients via learning about hazard,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.

- V. Bari, E. Vaini, V. Pistuddi, A. Fantinato, B. Cairo, B. De Maria, L. A. Dalla Vecchia, M. Ranucci, and A. Porta, “Comparison of causal and non-causal strategies for the assessment of baroreflex sensitivity in predicting acute kidney dysfunction after coronary artery bypass grafting,” Frontiers in physiology, vol.10, p. 1319, 2019.

- Porta, R. Colombo, A. Marchi, V. Bari, B. De Maria, G. Ranuzzi, S. Guzzetti, T. Fossali, and F. Raimondi, “Association between autonomic control indexes and mortality in subjects admitted to intensive care unit,” Scientific reports, vol. 8, no. 1, pp. 1–13, 2018.

- Yilin Yin and Chun-An Chou "A Novel Switching State Space Model for Post-ICU Mortality Prediction and Survival Analysis"IEEE journal of Biomedical and Health Informatics,2168-2194 (c) 2021.

• O. Badawi and M. J. Breslow, “Readmissions and death after icu discharge: development and validation of two predictive models,” PloS one, vol. 7, no. 11, p.e48758, 2012.

• N. Al-Subaie, T. Reynolds, A. Myers, R. Sunderland, A. Rhodes, R. Grounds, and G. Hall, “C-reactive protein as a predictor of outcome after discharge from the intensive care: a prospective observational study,” British journal of anaesthesia, vol. 105, no. 3, pp. 318–325, 2010.

• D. Lee and E. Horvitz, “Predicting mortality of intensive care patients via learning about hazard,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.

• V. Bari, E. Vaini, V. Pistuddi, A. Fantinato, B. Cairo, B. De Maria, L. A. Dalla Vecchia, M. Ranucci, and A. Porta, “Comparison of causal and non-causal strategies for the assessment of baroreflex sensitivity in predicting acute kidney dysfunction after coronary artery bypass grafting,” Frontiers in physiology, vol.10, p. 1319, 2019.

• Porta, R. Colombo, A. Marchi, V. Bari, B. De Maria, G. Ranuzzi, S. Guzzetti, T. Fossali, and F. Raimondi, “Association between autonomic control indexes and mortality in subjects admitted to intensive care unit,” Scientific reports, vol. 8, no. 1, pp. 1–13, 2018.