International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 56-61 www.ijaem.net ISSN: 2395-5252

Precision clinical medicine using machine learning

Mohit, Dr. Banita,
Mr. Rahul

Date of Submission: 01-06-2023 Date of Acceptance: 10-06-2023

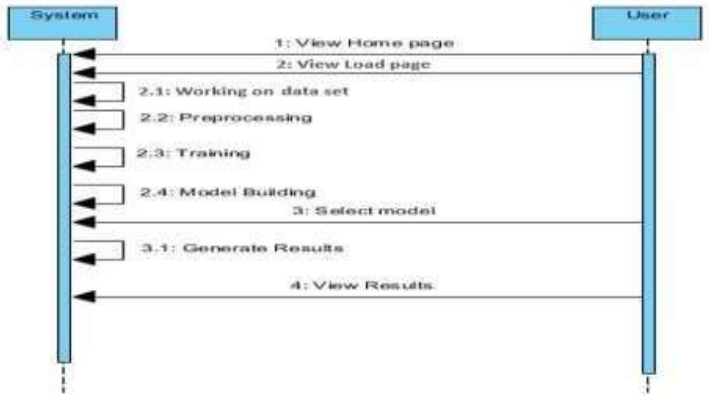
ABSTRACT
Remote patient monitoring in the intensive care unit (ICU) is a critical observation and assessment duty required for precision medicine. We recently developed a cloud-based intelligent remote patient monitoring (IRPM) framework in which we adhere to the state-of-the-art in risk stratification via machine learning-based prediction, but with minimal features that rely on vital signs, the most commonly used physiological variables obtained inside and outside hospitals. In this paper, we create three machine learning models for readmission, abnormality, and next-day vital sign readings to considerably improve the efficacy of the basic IRPM framework. We give a formal representation of a feature engineering technique and describe the creation and testing of three replicable machine learning prediction models. CU patient readmission, irregularity, and vital sign measures the next day We provided two solutions for data with unbalanced classes for the readmission model and applied five binary classification techniques to each approach. We used the same five algorithms to predict whether a patient will have abnormal health conditions in the abnormality model. Our findings show that by focusing on low and high quantile ranges of vital signs, we may still get an acceptable performance with these machine learning models. We discovered that using the most current vital sign values results in the lowest prediction error. Given the medical industry's large investment in patient monitoring devices, the developed models will be incorporated into an Intelligent ICU Patient Monitoring (IICUPM) module.

Aim and Objective

In this study, we aim to apply a human-centered design approach to evaluate the usability of a remote patient-monitoring system user interface (UI) in the ICU context and conceptualize and evaluate design changes.

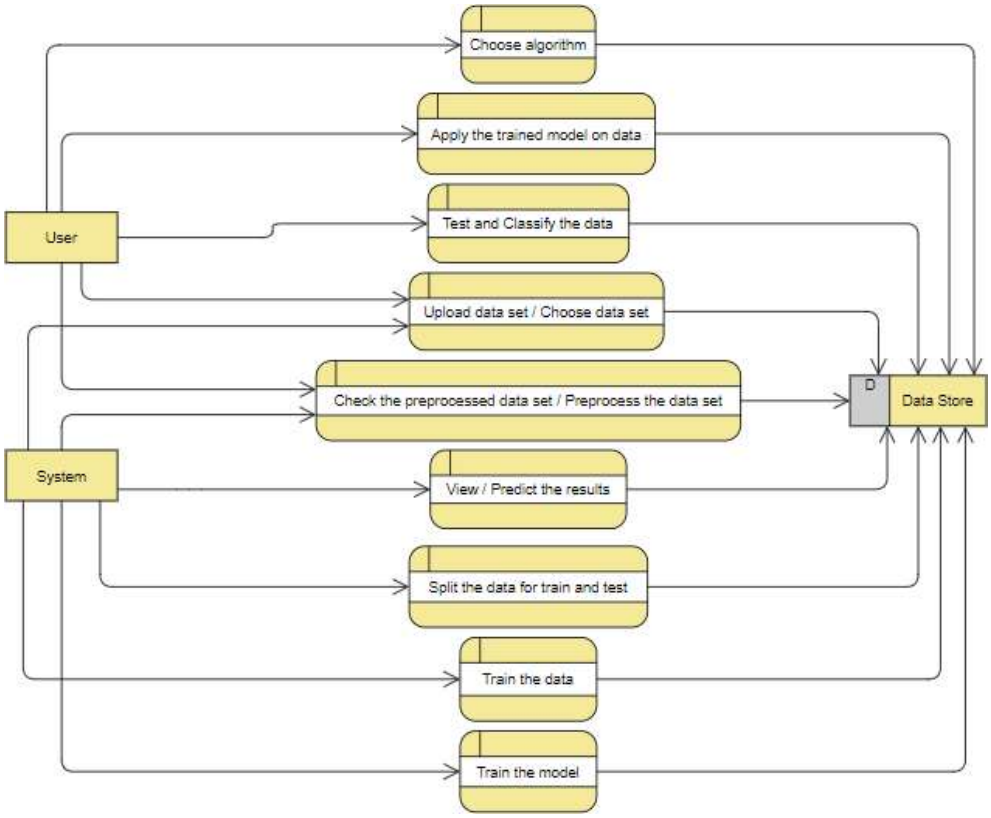
We aim to evaluate the usability of a remote patient-monitoring system and, specifically, identify usability problems, positive findings, and user ideas. We hypothesize that an HCD approach will help to implement evidence-based design changes that will improve the subjectively perceived usability and objective measures of the effectiveness and efficiency of the technology

Sequence Diagram



International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 56-61 www.ijaem.net ISSN: 2395-5252

Data Flow Diagram



IMPLEMENTATION

Algorithms

Random Forest Classifier

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the over fitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like Scikit-learn).

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

KNN

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 56-61 www.ijaem.net ISSN: 2395-5252

it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is one of the commonly used dimensionality reduction techniques in machine learning to solve more than two-class classification problems. It is also known as Normal Discriminant Analysis (NDA) or Discriminant Function Analysis (DFA).This can be used to project the features of higher dimensional space into lower-dimensional space in order to reduce resources and dimensional costs. In this topic, "Linear Discriminant Analysis (LDA) in machine learning", we will discuss the LDA algorithm for classification predictive modeling problems, limitation of logistic regression, representation of linear Discriminant analysis model, how to make a prediction using LDA, how to prepare data for LDA, extensions to LDA and much more. So, let's start with a quick introduction to Linear Discriminant Analysis (LDA) in machine learning. Although the logistic regression algorithm is limited to only two-class, linear Discriminant analysis is applicable for more than two classes of classification problems.Linear Discriminant analysis is one of the most popular dimensionality reduction techniques used for supervised classification problems in machine learning. It is also considered a pre-processing step for modeling differences in ML and applications of pattern classification

Whenever there is a requirement to separate two or more classes having multiple features efficiently, the Linear Discriminant Analysis model is considered the most common technique to solve such classification problems. For e.g., if we have two classes with multiple features andneed to separate them efficiently. When we classify them using a single feature, then it may show overlapping.

5.2.4 Multilayer Perceptron

In the world of deep learning, TensorFlow, Keras, Microsoft Cognitive Toolkit (CNTK), and PyTorch are very popular. Most of us may not realize that the very popular machine learning library.

We cannot fine-tune the parameters like different activation functions, weight initializers etc. for each layer.

MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable

The term "multilayer perceptron" does not refer to a single perceptron that has multiple layers. Rather, it contains many perceptrons that are organized into layers. An alternative is "multilayer perceptron network". Moreover, MLP "perceptrons" are not perceptrons in the strictest possible sense. True perceptrons are formally a special case of artificial neurons that use a threshold activation function such as the Heaviside step function. MLP perceptrons can employ arbitrary activation functions. A true perceptron performs binary classification, an MLP neuron is free to either perform classification or regression, depending upon its activation function.

The term "multilayer perceptron" later was applied without respect to nature of the nodes/layers, which can be composed of arbitrarily defined artificial neurons, and not perceptrons specifically. This interpretation avoids the loosening of the definition of "perceptron" to mean an artificial neuron in general.

MLPs are useful in research for their ability to solve problems stochastically, which often allowsapproximate solutions for extremely complex problems like fitness approximation.

5.2.5 Neural Networks

An artificial neural network (ANN) is the piece of a computing system designed to simulate the way the human brain analyzes and processes information. It is the foundation of artificial intelligence (AI) and solves problems that would prove impossible or difficult by human or statistical standards. ANNs have self-learning capabilities that enable them to produce better results as more data becomes available.

An ANN has hundreds or thousands of artificial neurons called processing units, which are interconnected by nodes.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 56-61 www.ijaem.net ISSN: 2395-5252

These processing units are made up of input and output units. The input units receive various forms and structures of information based on an internal weighting system, and the neural network attempts to learn about the information presented to produce one output report. Just like humans need rules and guidelines to come up with a result or output, ANNs also use a set of learning rules called backpropagation, an abbreviation for backward propagation of error, to perfect their output results.

An ANN initially goes through a training phase where it learns to recognize patterns in data, whether visually, aurally, or textually. During this supervised phase, the network compares its actual output produced with what it was meant to produce—the desired output. The difference between both outcomes is adjusted using backpropagation. This means that the network works backward, going from the output unit to the input units to adjust the weight of its connections between the units until the difference between the actual and desired outcome produces the lowest possible error.

Whenever we increase the layers in our ANN then it is nothing but our Deep Neural Networks.

Black Box Testing

Black Box Testing is a software testing technique in which the tester is not aware of the internal structure, design, or code of the software being tested. This approach tests the software based on its functionality, specifications, and requirements, without any knowledge of its internal workings. The tester provides inputs to the system and observes the outputs without knowing how the software produces them. Black Box Testing relies on predetermined requirements and specifications to validate the accuracy and completeness of the software. The test cases are designed to ensure that the software functions as expected and meets the user's needs.

Test Validation

Attributes	On Module	Validation
Upload CSV	Spi.h	Should successfully test and train
Run Python code	CV2 & numpy	Should Properly recognize the image and convert them to numeric expression
Classify the data	LinearSVC	Should Classify positive from negative reviews
Text Validation	TF-IDF	Should transform text into a meaningful representation of numbers
Merge file	Pandas	Should run code and merge the file

Test Cases

To evaluate the performance and reliability of our application, we have designed several test cases that cover various input scenarios. These test cases ensure that the application can handle different types of datasets. The test cases are as follows:

Test Case 1:

Description: This test case assesses the model's ability to classify datasets pertaining to stablevital signs of the patients. We expect the model to perform well and provide accurate results as this scenarios represents the ideal input for a monitoring system.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 56-61 www.ijaem.net ISSN: 2395-5252

Figure 6.4.1 Positive Test Case

Test Case 2:

Description: This test case evaluates the model's performance to classify datasets where the patient enters a precarious state. Since this case involves the need for immediate attention we expect the employed trigger to activate.

Figure 6.4.1 Negative Test Case

Test Case 3:

Description: This test case evaluates the model's performance on datasets with illogical values. Since the quality of the input is not optimal, the model might struggle to provide accurate results. However, the monitoring system should still attempt classify the patients details and provide the best possible prediction.

CONCLUSION

Machine learning methods applied to clinical datasets providehuge promise for tailored drug deliveryHuman illness therapy that is aimed Adding to our foundation we propose three replicable risk indicators as part of our recent creation of an Intelligent ICU Patient Monitoring (IICUPM) PlatformMLmodels for stratification. Our findings suggest that we canconstruct balanced predictive models for ICU patient readmissionand anomaly with more precision by combining.Combining ML and a quantiles technique that focused solely on important variablessigns. To prevent incorrect findings and low precision in the we presented two data options for the readmission model.Having unequal classes: one that employs under-samplingapproach, as well

as one that use the clustering re-sampling method. We also provide three methods for identifying predictors ofVital sign measures taken the next day in relation to a baseline.We created regression models utilizing two distinct classificationsalgorithms to each method. In general, we discovered that theerror compensation strategy outperformed the average.The measured method fared the poorest. The outcome suggeststhat employing the most current vital sign measures achieves the least error, especially when accounting for prior blunders in addition to offering three clear.This work adds a functionality to replicable ML modelsdeveloping an algorithm that may be used in a variety of critical care contexts.

FUTURE ENHANCEMENTS

Advanced Biometric Sensors: Future remote patient monitoring systems may incorporate more advanced biometric sensors capable of capturing a broader range of physiological data

Wearable and Implantable Devices: Remote patient monitoring systems will increasingly utilize wearable devices and even implantable sensors to provide continuous and real-time monitoring of patients' health.

Remote Medication Management: Future remote patient monitoring systems may include features to monitor medication adherence and remotely manage medication schedules

Contextual Data Analysis: Remote patient monitoring systems will gather contextual data beyond traditional health parameters. This may include environmental data, activity levels, sleep patterns, dietary information, and social factors.

Data Security and Privacy: As remote patient monitoring systems handle sensitive health data, future enhancements will focus on ensuring robust data security and privacy measures.

Predictive Analytics and Early Warning Systems: Future remote patient monitoring systems will employ predictive analytics and early warning systems to identify early signs of health deterioration or potential emergencies