International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Health Care Prediction

Vallam Krishna Bharath[1] , D. Suresh Reddy[2]
M Tech Student[1] ,Assistant Professor[2]
[1&2]Department of Computer Science and Engineering
[1&2]Siddhartha Educational Academy Groups of Institutions, Gollapali, Tirupathi, Andhra Pradesh, India

Article Info

Publication Issue :
Volume 8, Issue 6
November-December-2022

Page Number : 258-267

Article History
Accepted: 10 Nov 2022
Published: 23 Nov 2022

ABSTRACT

This study's goal was to create a patient discharge roster using machine learning methods. The discharge forecasting model suggested was then verified using the risk-based evaluation indicators from the ML tools. The study on a composed health data set of actual data, which includes aspects of severity, department, etc., We created a machine learning model to predict a patient's utilising the model types: Logistic Regression, Random Forests, and Light Gradient Boosting with the optimal parameters with various feature combinations and pre-processing techniques. When compared to all other systems, the suggested model was more accurate. The suggested model has improved accuracy when compared to SVM and NN models.

Keywords: Machine learning, LGBM, Random Forests, detection.

I. INTRODUCTION

Nearly all aspects of life, including medicine, have been automated, and illness diagnosis is no exception. Technologies for computer-aided sickness diagnosis make it simpler, faster, and less expensive to diagnose diseases. They facilitate decision-making by saving time and effort. Additionally, the automated diagnosis approach is less vulnerable to variation in doctor's preferences. Not only in the diagnosis and prognosis of disease, but in other areas as well. Additionally, it will assist us in providing discharge summaries that are based on the conditions. The bulk of clinical decision-making tools employ a better model. However, one should be sure the mower is providing or producing greater precision in decision-making.

Machine learning algorithms are quite good at forecasting the data. Using ID3, C4.5, and CART, Saba Bashir, Usman Qamar, Farhan Hassan Khan, and M. Younus Javed developed rule-based techniques for diabetes prediction in 2014. Using machine learning approaches, Wangshu Zhang, Feng Zeng, Xuebing Wu, Xuegong Zhang, and Rui Jiang predicted breast cancer metastases in 2009. Using machine learning techniques, Tao Zhang, Jiansheng Wu, and Haifeng Hu predicted text in 2014. A novel classification technique was developed in 2012 by Meng Wang, Kun Gao, Li-jing Wang, and Xiang-hu Miu using a C5.0 decision tree with several coupled classifiers. Lei Gao utilised a neural network to work on numerous forecasts during the 2013 Yuan-Dong LAN. In 2012, Magudeeswaran Veluchamy, Karthikeyan Perumal, and Thirumurugan Ponuchamy trained a back propagation neural network to predict blood cells with prediction efficiencies of 80% and 66.6% for normal and abnormal cells, respectively, using 4 geometrical features, 16 statistical features, and 7 moment invariant features. MLP (Multilayer Perceptron) was utilised by J. Poomcokrak and C. Neatpisarnvanit in 2008 to predict normal RBCs and Sickle cells.

With studies relating to patient safety, access, and financial performance, patient flow is a highly scrutinised indicator of hospital efficiency. 1-3 Preventable delays during hospitalization expose patients to higher hospitalization risk (such as hospital-acquired infections, unfavorable medicine responses, and pressure ulcers) and worsen patient experience. The same delays cause boarding in the emergency department, post-anesthesia care unit, and operating room because they prevent patient access downstream from internal and external admission sources (such as transfers). Each of these situations necessitates providing patient care in subpar facilities. Hospital patient flow issues are related to a changing legislative climate that supports publicly disclosed quality metrics and value-based incentive schemes aimed at lowering hospital-based care cost, which in the USA topped $1 trillion in 2017. 1 This has led many hospitals to concentrate all of their efforts on better managing capacity and patient flow. Each of these situations necessitates administering care to patients in subpar facilities. Hospital patient flow issues are related to a changing legislative climate that supports publicly disclosed quality metrics and value-based incentive schemes aimed at lowering hospital-based care cost, which in the USA topped $1 trillion in 2017. This has caused several hospitals to concentrate their efforts on better managing capacity and patient flow.

The maintenance of patient flow through the hospital has long been seen as a significant issue. Patients may need to be turned away if patient occupancy is too high, while maintaining preparedness is squandered if patient occupancy is too low. As a result, hospital management are curious about whether beds will be filled or available soon so that additional patients can be brought in or given discharge priority as needed.

Operational best practices and the potential for health information technology to support and/or allow innovative models that decrease patients' non-value-added time in hospitals serve as the foundation for these projects. Real-time demand capacity (RTDC) management and interdisciplinary discharge-focused rounds are two related best practices that have showed promise but inconsistent results. As part of rounds or morning bed huddles, clinician stakeholder groups have historically provided discharge forecasts. This manual procedure has shown considerable variability and low predictive accuracy, devoting clinician time away from direct patient care tasks. Operations-focused researchers have been inspired by this to create automated forecasts of individual discharges. A collaborative effort between surgeons, nurses, case managers, physical therapists, and others is necessary for timely patient release. They must recognize patients who are candidates for early release as well as the obstacles preventing the patient from leaving the hospital. Some hospitals expect the clinical staff to list and rank the patients who will be released each day, despite the fact that this is a laborious, difficult, and inconsistent process. Successful coordination of activities linked to release is challenging without accurately identifying patients and their discharge impediments.

Two algorithms, Random Forest, K Nearest Neighbors, and LGBM, were utilised. The suggested method's flowchart is shown here as defined.

II. RELATED WORKS

Operationally-Informed Hospital Wide Discharge Prediction: For the administration of hospital operations, precise patient discharge time estimations are essential. For the effective and efficient scheduling of hospital resources, such as beds and personnel, they are essential. In addition to hurting patient families and caregivers, unexpected discharges decrease hospital efficiency. Predictive models may be used to not only provide clinical decision assistance but also to enhance hospital operations because electronic health record data is becoming more widely available. In this study, we use a unique dataset that includes hourly data from the electronic health records of 14 distinct Kaiser Permanente hospitals to construct a prediction model for patient discharge by incorporating clinical expertise from operational executives at Kaiser Permanente Northern California. To forecast patient-level discharges for the following day at operationally pertinent moments on the hospital-centric timescale, we train and evaluate a number of algorithms with increasing degrees of complexity. With a gradient boosted model, we are able to attain the maximum AUC of 0.729, which is much better than the baseline model without hourly data and the existing estimates currently being used in these 14 facilities. We do a feature permutation significance evaluation and find that the introduction of the comprehensive, hourly data is mostly responsible for the improvement.

Summary: In this research, Andrew Ward and colleagues discuss how they used a gradient boosting model to get the greatest AUC of 0.729, which is much better than both the existing estimates used in these 14 facilities and the baseline model without hourly data.

A systematic review of research design and modelling techniques in inpatient bed management: The efficient distribution of scarce hospital bed resources is a challenging issue brought on by variable patient lengths of stay, shifting demand, unforeseen admissions, patient recovery state, and other elements. Inpatient units like intensive care units (ICU), step-down units, and medical or surgical units all have dedicated bed allocation teams that assign the beds and serve as a link between resource departments like the emergency department (ED), operating rooms (OR), labour and delivery (L&D), and referrals and inpatient units like those. Hospital costs are increased by prolonged inpatient stays caused by non-clinical needs, which can also lower patient satisfaction. This study offers a comprehensive evaluation of current inpatient bed management research and assesses the studies' use of various issue categories, measurement measures, and decision support strategies. We look for relevant publications from 2013 to 2017 using keywords like "bed management," "bed assignment," "bed planning," and "bed allocation" in the Google Scholar, PubMed, and Levy Library databases. The problem contexts (such as different practises, measurement measures, or in-scope units) and applicable approaches are used to identify and classify these articles (e.g. queueing theory, simulation, integer programming, etc.). At the conclusion of this study, a summary of current research trends, research gaps, and future research directions on inpatient bed management is provided. Overall, simulation modelling is the method of choice for researching inpatient bed management, but there are also chances to take system-wide inpatient flow into account, employ heuristic techniques, and include predictive models into inpatient bed management and allocation optimization.

Summary: Inpatient bed management research is dominated by simulation modelling, according to L. He, S. C. Madathil, and colleagues. However, there are chances to include predictive models, heuristic approaches, and system-wide inpatient flow into inpatient bed management and allocation optimization. 10612

ICU physicians are unable to accurately predict length of stay at admission: a prospective study: To assess the precision of the duration of stay estimate provided by doctors when a patient is admitted to an intensive care unit. Prospective cohort study is the design. Location: An cancer hospital with three medical-surgical intensive care units. All patients admitted between January and December 2014 are listed here. There were none. Principal outcome metrics: The doctors in charge of patient admission determined the length of stay in the intensive care unit (ICU) and divided it into three categories: 48 hours or less, 2 to 5 days, and more than 5 days. The agreement between the anticipated and actual length of stay in the critical care unit was calculated. Throughout the course of the trial, 2955 patients were hospitalized in total. In 1557 (52.7%) admissions, doctors correctly projected the average length of stay in the ICU. ICU stay duration was overestimated in 534 (18.1%) patients and underestimated in 864 (29.2%) cases. Poor (Kappa = 0.22) agreement between anticipated and actual critical care unit duration of stay was not related to physician characteristics. Compared to those of 48 h and 2-5 days, predictions of an intensive care unit duration of stay of >5 days were substantially less accurate (31.1, 59.8, and 53.1%, respectively, P 0.001). Conclusions: In these oncological critical care units, the prognosis of the length of stay is erroneous and, ideally, should not be given upon admission.

Summary: The intensive care unit length of stay forecast in these oncological intensive care units is unreliable, according to A. P. Nassar and P. Caruso, and should preferably not be given upon admission.

Pursuing Optimal Prediction of Dis-charge Time in ICUs with Machine Learning Methods: Patients in hospital critical care units (ICU) undergo ongoing examination. The evaluation's goal includes calculating the anticipated number of days until release. When managing ICUs, this value is crucial. According to certain research, medical practitioners are adept at forecasting short-term release dates but less so at making long-term forecasts. 1.79-day average prediction error may be achieved using machine learning techniques. In order to develop a data-driven model to forecast the timing of ICU patients' release on a daily basis, we conducted a research on 3,787 patient-days in the ICU of the Hospital Joan XXIII (Spain). The inaccuracy for our model, which is based on random forest technology, was 1.34 days. The number of occurrences needed to lower the inaccuracy below one day, according to our analysis of the model's development as more data became available, is 4,745. When we used all of the available data to train the model, we were able to predict whether an individual would survive an ICU stay with a mean error of less than half a day and a coefficient of determination (R2) above 97%. Similar outcomes were found when patients' gender and age were taken into account, indicating that our strategy is a suitable way to attain optimal performance if additional information is available.

Summary: In their forecasts of either ICU survivors or non survivors, D. Cuadrado and colleagues attained a mean error of less than half a day with a coefficient of determination (R2) over 97%.

Patient length of stay and mortality prediction: A survey: The use of data mining and machine learning techniques to enhance hospital performance has grown over the past several years. In particular, hospitals are looking to boost the statistics of their critical care units by lowering the number of patients who pass away there. Prediction of quantifiable outcomes, such as risk of complications, death, and duration of hospital stay, has been the subject of research. For healthcare professionals and patients alike, the duration of stay is a crucial parameter that is affected by a variety of circumstances. The length of stay in critical care is particularly important for patient experience and care costs, and it is impacted by elements unique to the extremely complex environment of the intensive care unit. When other outcomes cannot be monitored, the duration of stay is sometimes used as a substitute. One example is as a substitute for hospital or critical care unit mortality. The length of stay is another factor that has been used to gauge the severity of diseases and the usage of medical resources. This paper looks at various applications of length of stay and mortality prediction in acute medicine and the intensive care unit. It also emphasizes how to predict mortality and analyse length of stay. In addition, the study offers a categorization and assessment of the analytical approaches used to forecast mortality and the duration of stay in relation to a collection of pertinent research publications published between 1984 and 2016 in the field of survival analysis. The article also identifies a few of the domain's gaps and issues.

Summary: Based on a collection of pertinent research articles published between 1984 and 2016 in the field of survival analysis, the study categorises and assesses the analytical methodologies for forecasting length of stay and mortality.

Computerized prediction of intensive care unit discharge after cardiac surgery: development and validation of a Gaussian processes model: Cardiothoracic surgery patients' intensive care unit (ICU) length of stay (LOS) can vary greatly and is sometimes impossible to estimate in the first few hours following admission. A patient undergoing heart surgery may have an early clinical evolution that is indicative of his LOS. The goal of the current study was to create a predictive model for ICU discharge following non-emergency cardiac surgery by using Gaussian processes (GP), a machine learning approach, to analyse the first four hours of data in the electronic medical record of these patients.

Summary: The GP model showed much higher discriminative power than the Euro SCORE, ICU nurses, and ICU doctors' forecasts, and at least as excellent as theirs. The only accurate model was the GP model.

III. Methodology

Proposed system:

We suggest this application, which may be seen as a valuable system since it aids in reducing the constraints brought about by conventional and other existing ways. The goal of this project is to provide a quick and accurate approach for identifying and categorizing discharge days. We developed a potent algorithm to create this system in a Python-based environment using the flask framework.

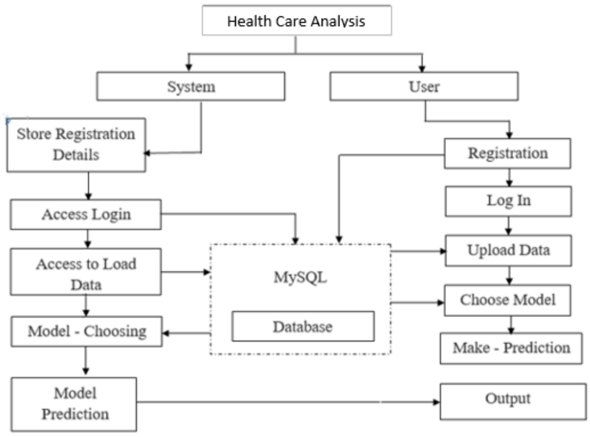


Figure 1 : Fake reviews dataset block diagram

IV. Implementation

1. Random Forest:

A random forest is a machine learning method for tackling classification and regression issues. It makes use of ensemble learning, a method for solving complicated issues by combining a number of classifiers.

In a random forest algorithm, there are many different decision trees. The random forest algorithm creates a "forest" that is trained via bagging or bootstrap aggregation. The accuracy of machine learning algorithms is improved by the ensemble meta-algorithm known as bagging.

Based on the predictions of the decision trees, the (random forest) algorithm determines the result. It makes predictions by averaging or averaging out the results from different trees. The accuracy of the result grows as the number of trees increases.

The decision tree algorithm's shortcomings are eliminated with a random forest. It improves precision and decreases dataset overfitting. Without numerous adjustments in packages (like Scikit-learn), it generates predictions.

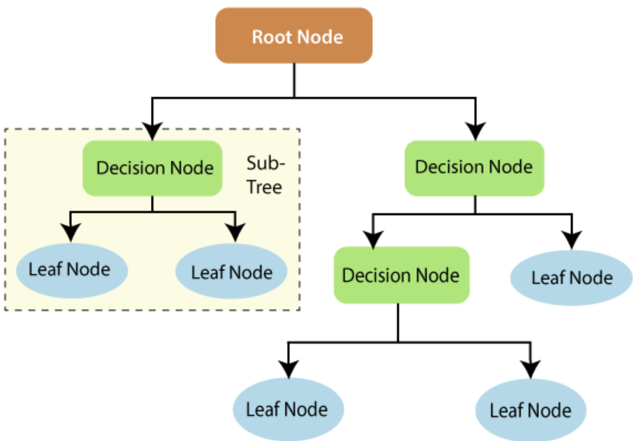
Features of a Random Forest Algorithm:

- Compared to the decision tree algorithm, it is more accurate.
- It offers a practical method for dealing with missing data.
- Without hyper-parameter adjustment, it can generate a fair prediction.
- It addresses the issue of decision trees' overfitting.
- At the node's splitting point in every random forest tree, a subset of characteristics is chosen at random.

Decision trees are the building blocks of a random forest algorithm. A decision support method that has a tree-like structure is called a decision tree. We will learn about decision trees and how random forest methods function.

Decision nodes, leaf nodes, and a root node are the three parts of a decision tree. A training dataset is divided into branches by a decision tree algorithm, which then separates those branches further. This process keeps on until a leaf node is reached. It is impossible to further separate the leaf node.

The qualities that are utilised to forecast the result are represented by the nodes in the decision tree. Links to the leaves are provided by decision nodes. The three categories of nodes in a decision tree are depicted in the diagram below.



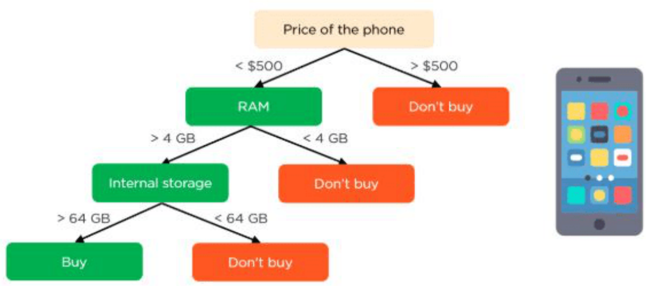
Information theory can provide further light on decision trees' operation. The foundation of a decision tree is information gain and entropy. An review of these key ideas will help us better comprehend the construction of decision trees.

Uncertainty may be measured using entropy. Given a collection of independent variables, information gain measures how much uncertainty in the target variable is decreased.

Using independent variables (features) to learn more about a target variable is the idea behind information gain (class). The information gain is calculated using the entropy of the target variable (Y) and the conditional entropy of Y (given X). In this instance, the entropy of Y is reduced by the conditional entropy. Decision trees are trained using information acquisition. It helps to make these plants' uneasiness lessened. A significant information gain denotes the removal of a large amount of uncertainty (information entropy). Splitting branches, a crucial step in the creation of decision trees, depends on entropy and information gain.

Take a look at a straightforward decision tree example. Let's say we want to forecast whether or not a consumer would buy a mobile phone. His selection is based on the phone's characteristics. A decision tree diagram can be used to display this study.

The above-mentioned phone features are represented by the decision's root node and decision nodes. The leaf node reflects the outcome, whether a purchase is made or not. The pricing, internal storage, and Random Access Memory are the primary criteria for selection (RAM). The following is how the decision tree will look.



Decision-tree application in random forest

The fundamental distinction between the random forest method and the decision tree algorithm is that the latter randomly selects the root nodes and groups the nodes. To get the necessary forecast, the random forest uses the bagging approach.

Bagging entails using many samples of data (training data) as opposed to a single sample. Predictions are made using characteristics and observations from a training dataset. Depending on the training data that the random forest algorithm receives, the decision trees provide a variety of results. The highest ranking of these outputs will be chosen as the final output.

The operation of random forests may still be explained using our initial example. The random forest will contain several decision trees rather than just one. Assume that there are only four decision trees in all. In this instance, four root nodes will be created using the training data, which consists of the phone's observations and characteristics.

The four features that potentially affect the customer's choice are represented by the root nodes (price, internal storage, camera, and RAM). By randomly choosing characteristics, the random forest will divide the nodes. The results of the four trees will be used to choose the final forecast.

The majority of decision trees will select the ultimate result. The ultimate forecast will be purchasing if three trees predict buying and one tree predicts not buying. It is anticipated that the client will purchase the phone in this instance.

2. KNN:

- One of the simplest machine learning techniques based on supervised learning is K-Nearest Neighbour.
- The K-NN method assumes that the new case/data and the existing cases are comparable and places the new case in the category that is most similar to the existing categories.
- The K-NN algorithm saves all the information that is accessible and categorises fresh input based on similarity. This means that utilising the K-NN method, fresh data may be quickly and accurately sorted into a suitable category.
- The K-NN approach may be used for both classification and regression problems, however it is more frequently utilised for classification issues.
- Because K-NN is a non-parametric method, it makes no assumptions about the underlying data.
- It is also known as a lazy learner algorithm since it keeps the dataset rather than learning immediately from the training set. Instead, it uses the dataset to execute an action when classifying data.
- The KNN method simply saves the dataset during the training phase and classifies fresh data into a category that is quite similar to the training data.
- Example: Let's say we have a picture of a creature that resembles both a cat and a dog, but we're not sure which one it is. Therefore, since the KNN method is based on a similarity metric, we may utilise it for this identification. By comparing the new data set's characteristics to those in the photographs of cats and dogs, our KNN model will determine which category the images belong to: cat or dog.

The K-NN working can be explained on the basis of the below algorithm:

Step 1: Decide on the neighbours' K-numbers.
Step 2: Calculate the Euclidean distance between K neighbours in step two.
Step 3: Based on the determined Euclidean distance, select the K closest neighbours.
Step 4: Count the number of data points in each category among these k neighbours.
Step 5: Assign the fresh data points to the category where the neighbour count is highest.
Step 6: Our model is complete

3. LGBM:

- The distributed gradient boosting framework for machine learning known as LightGBM, or Light Gradient Boosting Machine, was created by Microsoft and is free and open source. It is used for classification, ranking, and other machine learning applications and is based on decision tree algorithms. Performance and scalability are the main development priorities. The GBT, GBDT, GBRT, GBM, MART, and RF algorithms are among those supported by the LightGBM framework. Many of XGBoost's benefits, such as sparse optimization, parallel training, various loss functions, regularization, bagging, and early stopping, are also included in LightGBM. The way trees are built in the two differs significantly. Unlike the majority of previous implementations, LightGBM grows a tree row by row rather than level by level. Trees are instead grown leaf-wise. It selects the leaf it thinks will result in the greatest reduction in loss. Additionally, unlike XGBoost and other implementations, LightGBM does not employ the popular sorted-based decision tree learning technique, which looks for the optimum split point on sorted feature values. Instead, LightGBM uses a highly optimized decision tree learning method based on histograms, which has significant advantages in terms of both efficiency and memory use. The Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) approaches used by the LightGBM algorithm enable the algorithm to run more quickly while retaining a high degree of accuracy.

1) Control Parameters:

- **Max depth**: It gives the depth of the tree and also controls the over fitting of the model. If you feel your model is getting over fitted lower down the max depth.
- **Min_data_in_leaf**: Leaf minimum number of records also used for controlling overfitting of the model.
- **Feature fraction**: It decides the randomly chosen parameter in every iteration for building trees. If it is 0.7 then it means 70% of the parameter would be used.
- **Bagging fraction**: It checks for the data fraction that will be used in every iteration. Often, used to increase the training speed and avoid overfitting.
- **Early_stopping_round**: If the metric of the validation data does show any improvement in last early_stopping_round rounds. It will lower the imprudent iterations.
- **Lambda**: It states regularization. Its values range from 0 to 1.
- **Min_gain_to_spli**t: Used to control the number of splits in the tree.

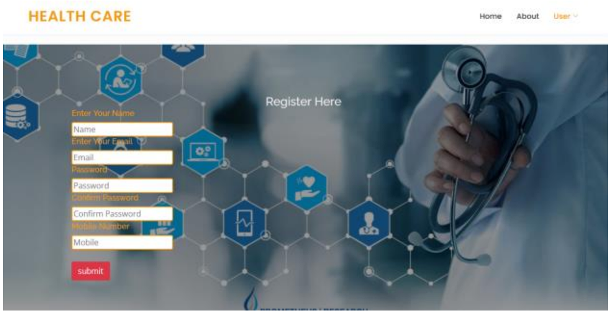
V. Results and Discussion

The following images will visually depict the process of our project.

Home page: In our project, we are detecting the anemia from the details entered by the user.

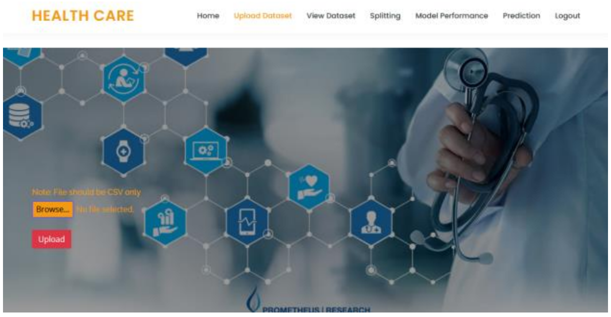
About page: Here the application describes what main objective of this project is.

Registration: Registration page in which user need to register to get started.

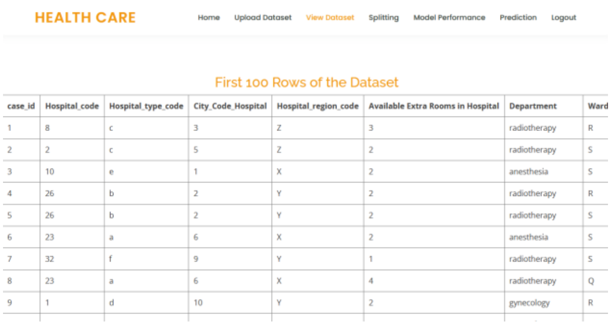


Login: Login page, user need to enter valid credentials in order to enter.

Upload page: Upload Page in order to upload the dataset.



View data: User views the data which he was uploaded to the system.

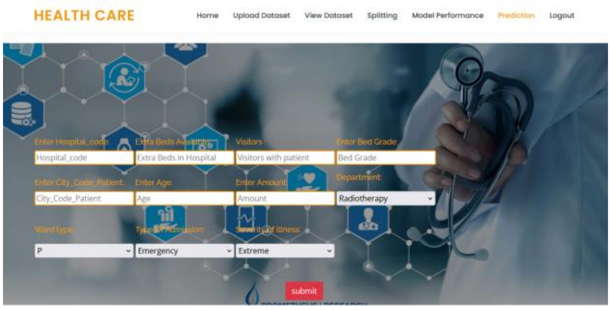


case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital	Department	Ward
1	8	c	3	Z	3	radiotherapy	R
2	2	c	5	Z	2	radiotherapy	S
3	10	e	1	X	2	anesthesia	S
4	26	b	2	Y	2	radiotherapy	R
5	26	b	2	Y	2	radiotherapy	S
6	23	a	6	X	2	anesthesia	S
7	32	f	9	Y	1	radiotherapy	S
8	23	a	6	X	4	radiotherapy	Q
9	1	d	10	Y	2	gynecology	R

Split data: Splits the data into train and test sets before training our model.

Model training: Here training of your model takes place and display the model r2 score.

Prediction: User need to enter the required fields in order to get the response from the data whether the patient is anaemic or not.



VI. Conclusion

The technique we created to predict when a patient would be released from the hospital in terms of days is effective. This is made in a user-friendly setting using Flask and Python programming. In order to assess if the patient has anemia or not, the system is likely to collect data from the user.

VII. REFERENCES

[1]. A. d. Grood, K. Blades, and S. Pendharkar, “A Review of Discharge Prediction Processes in Acute Care Hospitals,” Healthcare Policy —

Politiques de Sant´e, vol. 12, no. 2, pp. 105–115, 2016.

[2]. L. He, S. C. Madathil, A. Oberoi, G. Servis, and M. T. Khasawneh, “A systematic review of research design and modeling techniques in inpatient bed management,” Computers & Industrial Engineering, vol.

[3]. 127, no. IJITR 3 2015, pp. 451–466, 2019.

[4]. A. P. Nassar and P. Caruso, “ICU physicians are unable to accurately predict length of stay at admission: a prospective study,” International Journal for Quality in Health Care, vol. 28, no. 1, pp. 99–103, 2016.

[5]. D. Cuadrado, D. Ria˜no, J. G´omez, M. Bod´ı, G. Sirgo, F. Esteban, R. Garcia, and A. Rodr´ıguez, “Pursuing Optimal Prediction of Dis- charge Time in ICUs with Machine Learning Methods,” pp. 150–154, 2019.

[6]. A. Awad, M. Bader–El–Den, and J. McNicholas, “Patient length of stay and mortality prediction: A survey,” Health Services Management Research, vol. 30, no. 2, pp. 105–120, 2017.

[7]. G. Meyfroidt, F. G ̈uiza, D. Cottem, W. D. Becker, K. V. Loon, J.-M. Aerts, D. Berckmans, J. Ramon, M. Bruynooghe, and G. V. d. Berghe, “Computerized prediction of intensive care unit discharge after cardiac surgery: development and validation of a Gaussian processes model,” BMC Medical Informatics and Decision Making, vol. 11, no. 1, p. 64, 2011.

[8]. L. Turgeman, J. H. May, and R. Sciulli, “Insights from a machine learning model for predicting the hospital Length of Stay (LOS) at the time of admission,” Expert Systems with Applications, vol. 78, no. International journal of cardiology 176 2014, pp. 376–385, 2017

[9]. R. Houthooft, J. Ruyssinck, J. v. d. Herten, S. Stijven, I. Couckuyt, B. Gadeyne, F. Ongenae, K. Colpaert, J. Decruyenaere, T. Dhaene,and F. D. Turck, “Predictive modelling of survival and length of stay in critically ill patients using sequential organ failure scores,” Artificial Intelligence in Medicine, vol. 63, no. 3, pp. 191–207, 2015.

[10]. A. Morton, E. Marzban, G. Giannoulis, A. Patel, R. Aparasu, and A. Loannis, “A Comparison of Supervised Machine Learning Techniques for Predicting Short-Term In-Hospital Length of Stay among Diabetic Patients,” pp. 428–431, 2014.

[11]. T. A. Daghistani, R. Elshawi, S. Sakr, A. M. Ahmed, A. Al-Thwayee, and M. H. Al-Mallah, “Predictors of in-hospital length of stay among cardiac patients: A machine learning approach,” International Journal of Cardiology, vol. 288, no. Heart 2017, pp. 140–147, 2019.

Cite this article as :

Vallam Krishna Bharath, D. Suresh Reddy, "Health Care Prediction", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 6, pp. 258-267, November-December 2022.

Journal URL : https://ijsrcseit.com/CSEIT228635