ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

MACHINE LEARNING-BASED HEALTH PREDICTION SYSTEM USING IBM CLOUD AS PAAS

Mr.A.DSivarama Kumar[1], I.Savya[2], U.Venkata Nithin[3], A.Mani Ranga Sai Reddy[4]

[1]Assistant Professor,[2][3][4]Student

[1]sivaram.cse@svrec.ac.in,[2]savyainjeti9949@gmail.com,[3]ummadinithin999@gmail.com,

[4]maniambati4@gamil.com

Department of CSE, SVR Engineering College, Ayyaluru Metta, Nandyal(DIST)

Andhra Pradesh, India

Abstract: Develop a system that addresses healthcare gaps by focusing on responsiveness and scalability. The goal is to enhance service delivery in hospitals through improved critical patient care. Utilize machine learning and Cloud (PaaS) technologies to monitor real-time health conditions of critical patients. The emphasis is on enhancing healthcare decision-making and monitoring capabilities. Notably, the IBM Cloud component is simulated locally to overcome financial constraints. The model incorporates ensemble learning with predictors like Naïve Bayes, Logistic Regression, and Decision Tree Classifier. This approach aims to provide a robust and adaptable framework for predicting critical health conditions. Develop the "Critical Patient Management System" (CPMS) mobile app to facilitate remote monitoring of patient conditions in real-time. The app is designed to empower doctors with efficient tools for healthcare management, allowing them to monitor critical patients even when not physically present.

Index terms *- Patient Care System, Naïve Bayes, Logistic Regression, Ensemble Methods, IBM Cloud, IBM Watson Studio.*

1. INTRODUCTION

Critical Patient Caring or Monitoring System is a process where a doctor can continuously monitor more than one patient, for more than one parameter at a time in a remote place and also can have control over medicine dosage [1]. Development and evaluation of the ICU decision-support systems would be greatly facilitated by these systems. Devices such as vital sign monitors, mechanical ventilators dialysis machines, and some others are used to support critical patients whose bodies need time to recover and repair. Most of the machines are managed manually by supervising the patient's condition and test reports. So, we thought to automate the process and decision-making ability with the help of modern technology, especially the auto deployable machine

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

learning models and cloud computing. Machine learning models can predict the near future condition of the patients, whether their condition will increase or decrease, whether they need any immediate support or not. To generalize our models and data, we have selected IBM Cloud as a PaaS which altogether spans public, private and hybrid environments [2]. As initially, we cannot deploy our models directly, we had to use IBM Cloud, IBM Watson Studio for storing, testing and deploying our whole system. The ml models run within the cloud service and also trains with the auto-deployed data, the CPMS also can access the Cloud services through Bluemix [3]. The most significant of this paper carries the auto deployable machine learning model within the cloud storage with noteworthy accuracy. Also, testing and tuning approaches and parameter choosing, setting for different machine learning algorithms.

Health sector seems to be one of the neglected fields in terms of usage of technology in Bangladesh [4]. Although other sectors have adequately taken this advantage, health sector seems to be lagging behind. Government projects to integrate technology into the health sector has mostly failed. Due to inefficient handling of patients during an emergency, most of the cases result in death or permanent physical/mental damage to the patients, the main reason being the attending physician's inability to monitor the patient's vitals immediately [5]. The main method of communication is a mobile phone when the doctor is absent, resulting in communication mismatch. Our research installs the mechanism where the doctor can monitor the patient's vitals remotely, taking full advantage of Machine Learning to prescribe an advanced course and Cloud Computing to access the patient's vitals from any remote location. So, doctors can monitor multiple patients within a short span of time. Patients' relatives can get regular updates without having to visit the hospital every now and then.

2. LITERATURE SURVEY

This paper studies how to build a decision tree classifier [18] under the following scenario: a database is vertically partitioned into two pieces, with one piece owned by Alice and the other piece owned by Bob. Alice and Bob want to build a decision tree classifier based on such a database, but due to the privacy constraints, neither of them wants to disclose their private pieces to the other party or to any third party. We present a protocol that allows Alice and Bob to conduct such a classifier building without having to compromise their privacy. Our protocol uses an untrusted third-party server, and is built upon a useful building block, the scalar product protocol. Our solution to the scalar product protocol is more efficient than any existing solutions [32].

The experiments, aimed to compare three methods to create ensemble models implemented in a popular data mining system called WEKA, were carried out. Six common algorithms comprising two neural network algorithms, two decision trees [18, 32] for regression, linear regression, and support vector machine were used to construct ensemble models. All algorithms were employed to real-world datasets derived from the cadastral system and the registry of real estate transactions. Nonparametric Wilcoxon signed-rank tests to evaluate the differences between ensembles and original models were conducted. The results obtained show there is no single algorithm which produces the best ensembles and it is worth to seek an optimal hybrid multi-model solution. Keywords ensemble models-bagging-stacking-boosting-property valuation [29].

The naive Bayes classifier greatly simplify learn-ing by assuming that features are independent given class. Although independence is generally a poor assumption, in practice naive Bayes often competes well with more

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

sophisticated classifiers. Our broad goal is to understand the data character-istics which affect the performance of naive Bayes [25]. Our approach uses Monte Carlo simulations that al-low a systematic study of classification accuracy for several classes of randomly generated prob-lems. We analyze the impact of the distribution entropy on the classification error, showing that low-entropy feature distributions yield good per-formance of naive Bayes. We also demonstrate that naive Bayes works well for certain nearly-functional feature dependencies, thus reaching its best performance in two opposite cases: completely independent features (as expected) and function-ally dependent features (which is surprising). An-other surprising result is that the accuracy of naive Bayes is not directly correlated with the degree of feature dependencies measured as the class-conditional mutual information between the features. Instead, a better predictor of naive Bayes ac-curacy is the amount of information about the class that is lost because of the independence assumption.

Machine learning algorithms [16, 27, 30, 32] frequently require careful tuning of model hyperparameters, regularization terms, and optimization parameters. Unfortunately, this tuning is often a "black art" that requires expert experience, unwritten rules of thumb, or sometimes brute-force search. Much more appealing is the idea of developing automatic approaches which can optimize the performance of a given learning algorithm to the task at hand. In this work, we consider the automatic tuning problem within the framework of Bayesian optimization, in which a learning algorithm's generalization performance is modeled as a sample from a Gaussian process (GP). The tractable posterior distribution induced by the GP leads to efficient use of the information gathered by previous experiments, enabling optimal choices about what parameters to try next. Here we show how the effects of the Gaussian process prior and the associated inference procedure can have a large impact on the success or failure of Bayesian optimization. We show that thoughtful choices can lead to results that exceed expert-level performance in tuning machine learning algorithms. We also describe new algorithms that take into account the variable cost (duration) of learning experiments and that can leverage the presence of multiple cores for parallel experimentation. We show that these proposed algorithms improve on previous automatic procedures and can reach or surpass human expert-level optimization on a diverse set of contemporary algorithms including latent Dirichlet allocation, structured SVMs and convolutional neural networks.

For many computer vision problems, the most time consuming component consists of nearest neighbor matching in high-dimensional spaces. There are no known exact algorithms for solving these high-dimensional problems that are faster than linear search. Approximate algorithms are known to provide large speedups with only minor loss in accuracy, but many such algorithms have been published with only minimal guidance on selecting an algorithm and its parameters for any given problem. In this paper [28], we describe a system that answers the question, “What is the fastest approximate nearest-neighbor algorithm for my data?” Our system will take any given dataset and desired degree of precision and use these to automatically determine the best algorithm and parameter values. We also describe a new algorithm that applies priority search on hierarchical k-means trees, which we have found to provide the best known performance on many datasets. After testing a range of alternatives, we have found that multiple randomized k-d trees provide the best performance for other datasets [18, 29, 32]. We are releasing public domain code that implements these approaches. This library provides about one order of magnitude improvement in query time over

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

the best previously available software and provides fully automated parameter selection.

3. METHODOLOGY

i) Proposed Work:

The proposed system is an advanced Machine Learning (ML) [16, 27, 30, 32] based Health Prediction System utilizing Cloud as a Platform as a Service (PaaS). Employing ML models such as Naïve Bayes, Logistic Regression, and others, the system facilitates real-time training and deployment for accurate health predictions. A mobile application named "Critical Patient Management System (CPMS)" offers healthcare professionals real-time access to patient data. With a focus on smart healthcare, this system aims to provide immediate feedback, proactive intervention, and ultimately enhance patient care in hospital settings. The system employs advanced machine learning models for real-time health predictions, providing instant insights into patient conditions and enabling prompt interventions by healthcare professionals. Incorporating a diverse range of machine learning models, including Naïve Bayes and Logistic Regression, ensures a thorough analysis of health data. This approach enhances the accuracy and reliability of predictions, covering various aspects of patient well-being. [3, 17] The "Critical Patient Management System (CPMS)" mobile app is developed to offer healthcare professionals a user-friendly interface, facilitating real-time monitoring and seamless access to patient data. This promotes efficient healthcare management.

ii) System Architecture:

The system architecture for the proposed Critical Patient Management System (CPMS) integrates a mobile app interface for healthcare professionals to input patient vitals, a locally simulated IBM Cloud (PaaS) for storing and executing diverse machine learning models (Naïve Bayes, Logistic Regression, Decision Tree Classifier) to predict critical health conditions in real-time. Result feedback is provided through the mobile app, allowing doctors to make timely decisions based on real-time predictions, and the scalable cloud-based architecture ensures accessibility and efficient monitoring of patients from various locations.

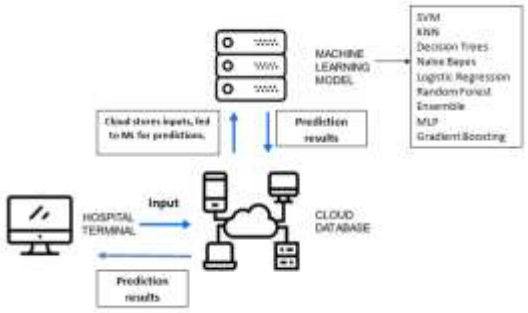


Fig 1 Proposed Architecture

iii) Data collection:

HEALTHCARE DATASET

This is the dataset on which we have trained the machine learning models. [15] We are displaying few rows here and columns representing various attributes such as age, sex, chest pain type (cp), resting blood pressure (trestbps), cholesterol levels (chol), fasting blood sugar (fbs), rest electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise relative to rest (oldpeak), slope of the peak exercise ST segment (slope), number of major vessels colored by fluoroscopy (ca), thalassemia type (thal), and a class label (class) indicating whether the patient's condition is stable (0) or abnormal (1). Using this data we will train the machine learning models.

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

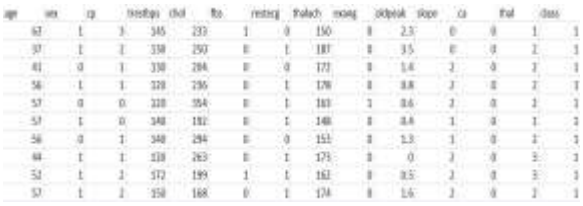


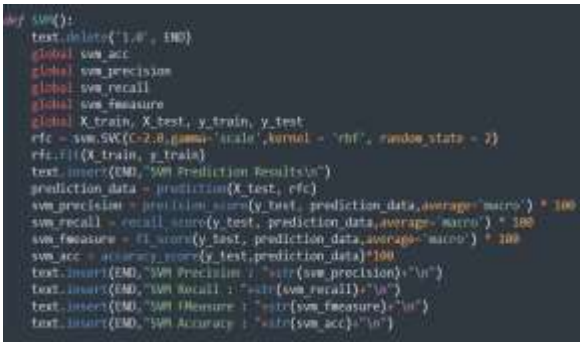
Fig 2 Dataset

The process begins with the collection of patient vitals, including body temperature, blood pressure, and other relevant parameters. This data can be manually input or obtained through sensor-equipped devices.

- Healthcare professionals use the [3] CPMS mobile app to input patient vitals. The app serves as a user-friendly interface, facilitating the seamless submission of data and real-time interaction with the system.
- The CPMS [17] mobile app uploads the collected patient vitals to the locally simulated IBM Cloud (PaaS). This cloud serves as the central repository for storing machine learning algorithms and processing patient data.
- Diverse machine learning models, such as Naïve Bayes, Logistic Regression, and Decision Tree Classifier, analyze the uploaded patient vitals independently. These models form the predictive engine for real-time health predictions.
- The system provides real-time feedback to healthcare professionals through the CPMS mobile app. Predictions, alerts, and relevant insights based on the ensemble learning model are displayed, empowering doctors to make informed and timely decisions.
- Historical patient data is stored in a secure database integrated into the cloud infrastructure. This database supports continuous learning, allowing the machine learning models to improve and adapt over time based on past experiences.
- The cloud-based architecture ensures scalability, accommodating an increasing volume of patient data. Additionally, the system is designed to be accessible remotely, providing healthcare professionals with the flexibility to monitor patients from various locations.

vi) Algorithms:

Support Vector Machine (SVM): Support Vector Machine (SVM) is selected for its efficacy in classifying patient vitals and predicting critical health conditions. By finding the optimal hyperplane to separate data into classes, SVM enhances the precision of real-time health predictions in the Critical Patient Management System, particularly suited for handling complex and high-dimensional data.



```
def svm():
    text.delete('1.0', END)
    global svm_acc
    global svm_precision
    global svm_recall
    global svm_fmeasure
    global X_train, X_test, y_train, y_test
    rfc = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state = 2)
    rfc.fit(X_train, y_train)
    text.insert(END,"SVM Prediction Results\n")
    prediction_data = prediction(X_test, rfc)
    svm_precision = precision_score(y_test, prediction_data,average='macro') * 100
    svm_recall = recall_score(y_test, prediction_data,average='macro') * 100
    svm_fmeasure = f1_score(y_test, prediction_data,average='macro') * 100
    svm_acc = accuracy_score(y_test,prediction_data)*100
    text.insert(END,"SVM Precision : "+str(svm_precision)+"\n")
    text.insert(END,"SVM Recall : "+str(svm_recall)+"\n")
    text.insert(END,"SVM FMeasure : "+str(svm_fmeasure)+"\n")
    text.insert(END,"SVM Accuracy : "+str(svm_acc)+"\n")
```

Fig 3 AVM

K-Nearest Neighbours (KNN): K-Nearest Neighbors (KNN) is employed in the Critical Patient Management System to predict patient health conditions based on similarity to nearby data points. Each patient's vitals are represented as a data point, and predictions are made by considering the majority class of its k-nearest neighbors in the feature space. KNN contributes to real-time health predictions by dynamically adapting to local context and is part of the ensemble learning approach for robust predictions in the project.

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

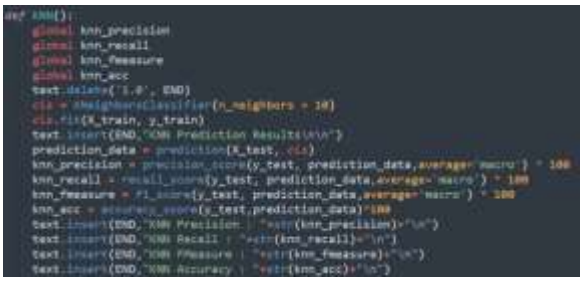


Fig 4 KNN

Decision Tree: Decision Trees are utilized in the Critical Patient Management System to analyze patient vitals and predict critical health conditions. These machine learning algorithms create a tree-like structure based on features, allowing for interpretable decision-making. Their suitability lies in providing healthcare professionals with transparent insights into the logic behind predictions, crucial for real-time patient monitoring and informed decision-making [18].

Fig 5 Decision tree

Naïve Bayes: Naïve Bayes, chosen for its simplicity and efficiency, is a probabilistic algorithm in the Critical Patient Management System project. Despite assuming feature independence, it effectively calculates the probability of abnormal patient conditions based on vitals. Its inclusion in the ensemble learning model adds diversity, contributing to accurate real-time predictions in the healthcare data analysis [16].

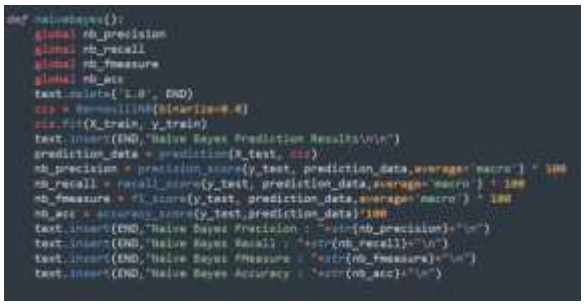


Fig 6 Naïve bayes

Logistic regression: Logistic Regression is chosen for the Critical Patient Management System project to predict critical health conditions from patient vitals. Specifically designed for binary classification, it calculates the probability of a patient's condition being abnormal or stable based on input features. This makes Logistic Regression well-suited for real-time health predictions, facilitating prompt decision-making by healthcare professionals [17].

Fig 7 Logistic regression

Random forest: Random Forest, a key component in the Critical Patient Management System, improves prediction accuracy by creating multiple decision trees during training and combining their outputs. This ensemble method enhances robustness by mitigating overfitting through the use of random subsets of the dataset for each tree. In healthcare predictions, Random Forest's versatility and reliability make it well-suited for handling the complexity of patient data, contributing significantly to the system's overall predictive capabilities [20].

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

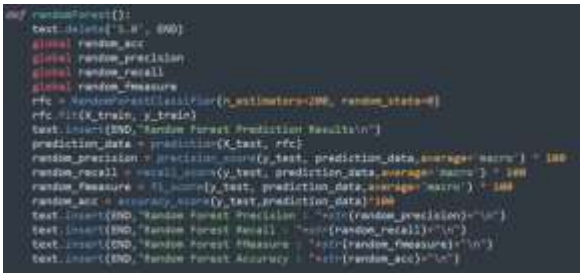


Fig 8 Random forest

Ensemble Algorithm: The ensemble algorithm in this project combines Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT) classifiers to enhance predictive accuracy. Leveraging the unique strengths of each algorithm, SVM captures complex relationships, RF mitigates overfitting, and DT provides interpretable insights. This collaborative approach improves the overall reliability of real-time health predictions in the Critical Patient Management System.

Fig 9 Ensemble algorithm

MLP: In the Critical Patient Management System, Multilayer Perceptron (MLP) is chosen for its capacity to capture intricate patterns in patient vitals. As an artificial neural network, MLP's proficiency in handling non-linear relationships contributes to the ensemble learning model, enhancing the accuracy of real-time health condition assessments [19].

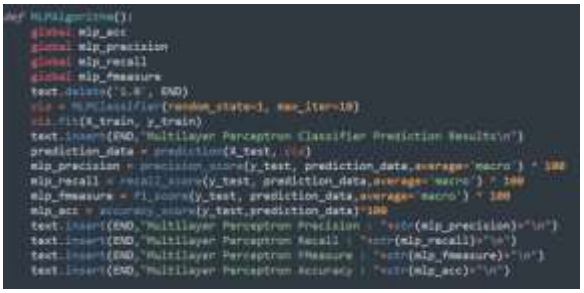


Fig 10 MLP

Gradient Boosting: Gradient Boosting is an ensemble learning method that sequentially improves predictive models, making it ideal for enhancing accuracy in the Critical Patient Management System project. It iteratively corrects errors to create a robust predictive model, well-suited for real-time analysis of diverse patient vitals and dynamic critical health conditions.

Fig 11 Gradient boosting

4. EXPERIMENTAL RESULTS

Fig 12 Home screen

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

Fig 13 Upload healthcare dataset and preprocess

Fig 14 Run Algorithm

Precision Measures the accuracy of positive predictions, indicating how many predicted positives were actually correct.

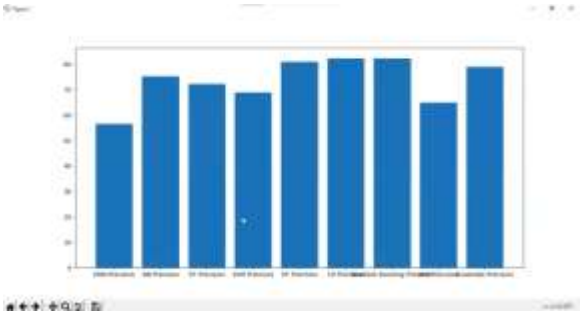


Fig 15 Precision graph

Recall Measures the ability to identify all relevant instances, showing how many actual positives were correctly predicted.

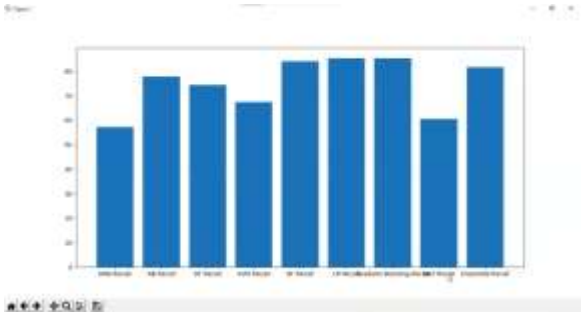


Fig 16 Recall graph

F1 score Combines precision and recall into a single metric, balancing accuracy and completeness in predictions.

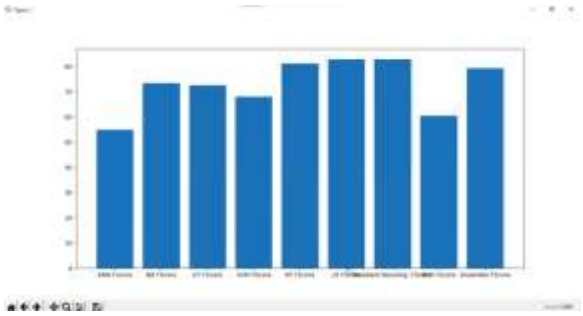


Fig 17 Fscore graph

Accuracy Measures the overall correctness of predictions, showing the percentage of correctly classified instances.

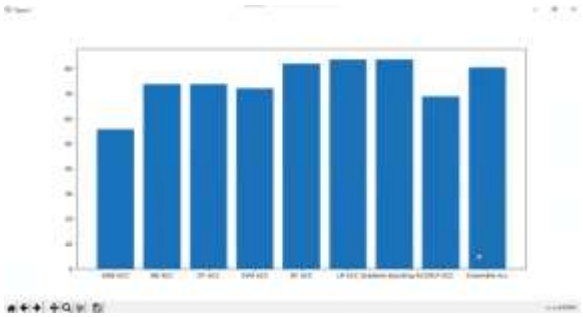


Fig 18 Accuracy graph

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

Fig 19 Start cloud server

Fig 20 Upload files

5. CONCLUSION

The project achieves a seamless fusion of cloud technology and machine learning [16], streamlining the process of identifying abnormal health conditions. Collaborative modules work unitedly to provide a holistic approach to health predictions. By strategically utilizing a locally simulated dummy cloud, the project ensures practicality and accessibility. This approach allows for thorough testing without imposing financial burdens on students and developers. The mobile module, particularly the patient monitoring app, facilitates remote tracking of vital signs. Its adaptability to real-world scenarios, such as allowing test data uploads, enhances practicality and user-friendliness in diverse healthcare settings. Through systematic validation, the project evaluates the accuracy of various machine learning algorithms [27, 30]. The standout performer is the ensemble algorithm, surpassing others and confirming its efficacy in predicting patient conditions based on a range of health parameters.

6. FUTURE SCOPE

Future enhancements may involve installing embedded systems to capture live readings from ICU machines like Ventilators, Medicine Pumps, and Heart Monitors, enabling a more comprehensive patient health monitoring approach. The system's potential growth lies in integrating additional patient monitoring systems through diverse embedded systems and real-time operating systems. This expansion can contribute to a more inclusive and adaptable healthcare monitoring infrastructure [2]. Ongoing research and development opportunities exist in machine learning for predicting patient mortality. Future work aims to enhance model accuracy and performance, keeping pace with evolving healthcare demands and technological advancements. The system's scalability can be leveraged to enable doctors to remotely monitor multiple patients efficiently, providing timely updates to relatives without requiring frequent hospital visits. This expansion enhances patient care accessibility and reduces the burden on healthcare facilities.

REFERENCES

[1] Gardner R.M., Shabot M.M. (2006) Patient-Monitoring Systems. In: Shortliffe E.H., Cimino J.J. (eds) Biomedical Informatics. Health Informatics. Springer, New York, NY

[2] Aggarwal, M., & Madhukar, M. (2017). IBM's Watson Analytics for Health Care: A Miracle Made True. In Cloud Computing Systems and Applications in Healthcare (pp. 117-134). IGI Global.

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

[3] "Rational Unified Process", URL: [online] Available: https://www.ibm.com/developerworks/rational/library/content/03July/10 00/1251/1251_bestpractices_TP026B.pdf.

[4] Anwar Islam, Tuhin Biswas. Health System in Bangladesh: Challenges and Opportunities. American Journal of Health Research. Vol. 2, No. 6, 2014, pp. 366-374. doi: 10.11648/j.ajhr.20140206.18

[5] P. Griffiths, A. R. Saucedo, P. Schmidt, G. Smith. Vital signs monitoring in hospitals at night. (n.d.). Retrieved from https://www.nursingtimes.net/clinical-archive/assessment-skills/vitalsigns-monitoring-in-hospitals-at-night/5089989.article.

[6] An Embedded, GSM based, Multiparameter, Realtime Patient Monitoring System and Control – An Implementation for ICU Patients. Kumar, R., & Rajasekaran, M. P. (2016, January). An IoT based patient monitoring system using raspberry Pi. In 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16) (pp. 1-4). IEEE.

[7] Nejkar, V. A., Nimbhorkar, S. R., Paliwal, J. K., & Shrivastav, A. A. (2018). Smart Nanny an IoT Based Baby Monitoring System. iManager's Journal on Computer Science, 6(1), 28.

[8] Ruiz, V. M., Saenz, L., Lopez-Magallon, A., Shields, A., Ogoe, H. A., Suresh, S., & Tsui, F. R. (2019). Early Prediction of Critical Events for Infants with Single Ventricle Physiology in Critical Care Using Routinely Collected Data. The Journal of Thoracic and Cardiovascular Surgery.

[9] Lin, K., Hu, Y., & Kong, G. (2019). Predicting In-hospital Mortality of Patients with Acute Kidney Injury in the ICU Using Random Forest Model. International Journal of Medical Informatics.

[10] Teres, D., Lemeshow, S., Avrunin, J. S., & Pastides, H. A. R. R. I. S. (1987). Validation of the mortality prediction model for ICU patients. Critical care medicine, 15(3), 208-213.

[11] Ahmed, S. (n.d.). BREAST CANCER: PRESENTATION AND LIMITATION OF TREATMENT – BANGLADESH PERSPECTIVE. doi:10.4172/1948-5956.S1.041

[12] Clarke, F & Mcdonald, Ellen & Griffith, Lauren & Cook, D & Mead, M & Guyatt, G & Rabbat, Christian & Geerts, W & Arnold, D & Warkentin, T & Crowther, Mark. (2004). Thrombocytopenia in medical–surgical ICU patients. Critical Care. 8. 1-1. 10.1186/cc2592.

[13] Choi, N. G., DiNitto, D. M., & Kim, J. (2014). Discrepancy Between Chronological Age and Felt Age: Age Group Difference in Objective and Subjective Health as Correlates. Journal of Aging and Health, 26(3), 458–473. https://doi.org/10.1177/0898264314523449.

[14] Db2 Warehouse. (n.d.). Retrieved from https://www.ibm.com/usen/marketplace/db2-warehouse.

[15] Das, Sudipto & Nishimura, Shoji & Agrawal, Divyakant & El Abbadi, Amr. (2010). Live Database Migration for Elasticity in a Multitenant Database for Cloud Platforms.

[16] Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In European conference on machine learning (pp. 4-15). Springer, Berlin, Heidelberg.

[17] Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification

models: a methodology review. Journal of Biomedical Informatics, 35(5-6), 352-359.

[18] Du, W., & Zhan, Z. (2002, December). Building decision tree classifier on private data. In Proceedings of the IEEE International Conference on Privacy, security and data mining-Volume 14 (pp. 1-8). Australian Computer Society, Inc.

[19] Ghate, V. N., & Dudul, S. V. (2010). Optimal MLP neural network classifier for fault detection of three phase induction motor. Expert Systems with Applications, 37(4), 3468-3481.

[20] Graczyk, M., Lasota, T., Trawiński, B., & Trawiński, K. (2010, March). Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. In Asian conference on intelligent information and database systems (pp. 340-350). Springer, Berlin, Heidelberg.

[21] What is the Jupyter Notebook? (n.d.). Retrieved from https://jupyternotebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html.

[22] Getting started with SMS Gateway. (n.d.). Retrieved from https://www.ibm.com/support/knowledgecenter/en/SS4U29/sms_getting started.html.

[23] Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. Psychometrika, 38(1), 1-10.

[24] Lu, Y., Cohen, I., Zhou, X. S., & Tian, Q. (2007, September). Feature selection using principal feature analysis. In Proceedings of the 15th ACM International Conference on Multimedia (pp. 301-304). ACM.

[25] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

[26] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

[27] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Advances in neural information processing systems (pp. 2951-2959).

[28] Muja, M., & Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1), 2(331-340), 2.

[29] Naidoo, L., Cho, M. A., Mathieu, R., & Asner, G. (2012). Classification of savanna tree species, in the Greater Kruger National Park region, by integrating hyperspectral and LiDAR data in a Random Forest data mining environment. ISPRS journal of Photogrammetry and Remote Sensing, 69, 167-179.

[30] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Advances in neural information processing systems (pp. 2962-2970).

[31] Mistry, J., & Inden, B. (2018). An approach to sign language translation using the Intel Realsense camera.

[32] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine learning, 40(2), 139-157.

[33] Baumann, K. (2003). Cross-validation as the objective function for variable-selection techniques. TrAC Trends in Analytical Chemistry, 22(6), 395-406.

ISSN: 2278-4632
Vol-14, Issue-4, April: 2024

[34] Overview: Estimators, transformers and pipelines - spark.ml. (n.d.). Retrieved from https://spark.apache.org/docs/1.6.0/ml-guide.html.