Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019)
IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8

Machine Learning based Health Prediction System using IBM Cloud as PaaS

Asif Ahmed Neloy
dept. of electrical and computer engineering
North South University
Dhaka, Bangladesh
asif.neloy@northsouth.edu

Sazid Alam
dept. of electrical and computer engineering
North South University
Dhaka, Bangladesh
sazid.alam@northsouth.edu

Rafia Alif Bindu
dept. of electrical and computer engineering
North South University
Dhaka, Bangladesh
rafia.bindu@northsouth.edu

Nusrat Jahan Moni
dept. of electrical and computer engineering
North South University
Dhaka, Bangladesh
nusrat.moni@northsouth.edu

***Abstract*—Adaptable Critical Patient Caring system is a key concern for hospitals in developing countries like Bangladesh. Most of the hospital in Bangladesh lack serving proper health service due to unavailability of appropriate, easy and scalable smart systems. The aim of this project is to build an adequate system for hospitals to serve critical patients with a real-time feedback method. In this paper, we propose a generic architecture, associated terminology and a classificatory model for observing critical patient's health condition with machine learning and IBM cloud computing as Platform as a service (PaaS). Machine Learning (ML) based health prediction of the patients is the key concept of this research. IBM Cloud, IBM Watson studio is the platform for this research to store and maintain our data and ml models. For our ml models, we have chosen the following Base Predictors: Naïve Bayes, Logistic Regression, KNeighbors Classifier, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, and MLP Classifier. For improving the accuracy of the model, the bagging method of ensemble learning has been used. The following algorithms are used for ensemble learning: Bagging Random Forest, Bagging Extra Trees, Bagging KNeighbors, Bagging SVC, and Bagging Ridge. We have developed a mobile application named "Critical Patient Management System - CPMS" for real-time data and information view. The system architecture is designed in such a way that the ml models can train and deploy in a real-time interval by retrieving the data from IBM Cloud and the cloud information can also be accessed through CPMS in a requested time interval. To help the doctors, the ml models will predict the condition of a patient. If the prediction based on the condition gets worse, the CPMS will send an SMS to the duty doctor and nurse for getting immediate attention to the patient. Combining with the ml models and mobile application, the project may serve as a smart healthcare solution for the hospitals.**

Keywords*—*Patient Care System, Naïve Bayes, Logistic Regression, Ensemble Methods, IBM Cloud, IBM Watson Studio

I. 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 and dialysis machines, and some others more 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 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.

II. MOTIVATION

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.

III. RELATED WORK

There is been significant progress in patient monitoring systems using different embedder and a real-time operating system (RTOS). In 2016, R. Kumar and M. Pallikonda Rajasekaran discussed, monitoring patient's body temperature, respiration rate, heartbeat, and body movement using a Raspberry Pi board [6]. They provided an IoT based patient monitoring system. Vaibhavi A. Nejkar, Shambhavi R. Nimbhorkar, Jyoti K. Paliwal, A. A. Shrivastav proposed a nanny nurse for the baby monitoring system. This prototype can also serve critical patients [7]. In the case of machine learning, there are many novel works describing mortality prediction of the patients using Cardiac-intensive-care Warning Index (C-WIN) system [8], Predicting in-hospital mortality of patients with acute kidney injury in the ICU using random forest model [9] and also different models validate the models that are being used to predict the mortality [10]. Studying different novel works and approaches, we came to a conclusion to bridge the gap between the automated patient caring system with machine learning and real-time feedback facility. Also, we had to consider the state of the art that can be done in all the requirements and effective costings.

IV. PROBLEM DEFINITION AND PROPOSED SOLUTION

In a country like Bangladesh, where there are only 3 doctors for 10000 people [11], it's quite impossible to manage so many patients at a time and also the top experienced doctor has a high demand of the patient. Also, there is a massive 19% rate of producing the wrong treatment to the ICU and cabin Patients [12]. From the hospital survey, we came to know about the parameters that define the medical condition of a patient, the natural health conditions, the critical health conditions and most importantly the crucial time to make adequate decisions. Health conditions and parameters differ from age to age and from patient to patient [13]. Also, a single instance can affect others, like heart disease, can trigger lower pulse or maybe higher pulse based on the high pressure or low pressure. Considering all the survey results, we have created the project plan from step by step by building the core components first.

V. SYSTEM DESIGN

The whole system is comprised of six major layouts. ML models are the core analysis of the project. Based on the models, the project operates through the IBM Cloud and Watson studio for deployment service. Fig. 1 illustrates the overview of system design. Each of them is briefly described below.

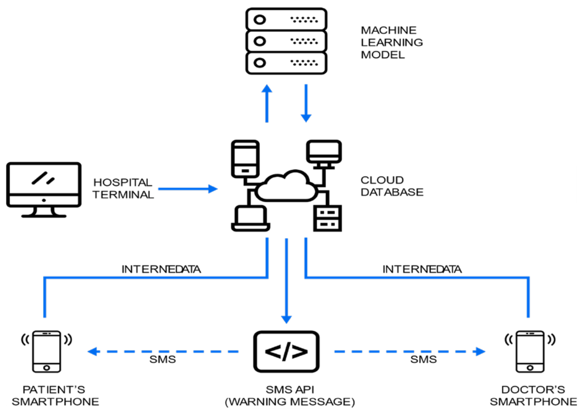

Fig 1. System Architecture

A. Cloud Architecture

To establish the system proposed by the author's research, IBM Cloud with IBM Watson was required. In the Bluemix Console of IBM Cloud, an instance of the Db2 Warehouse on Cloud service was created. Db2 Warehouse service supports SQL based database [14], thus it was selected. The NDX9073 schema was selected to build up the database. Tables for Patient Registration, Patient Update, Doctor's info, ML Result were created, where concerned data were stored.

B. Dataset Pre-processing

The training dataset for the ml models will be redirected from IBM Cloud storage. We cannot perform manual pre-processing within our auto-deployed database [15]. To perform pre-processing from IBM Cloud we have to use the "Refine" function from model building. From Refine function, a set of instruction can be set to pre-process automatically. Table. 1 illustrates the sample dataset after refinement. The dataset contains primarily 5000 sample data for our training and later it has been divided with the required proportion for testing and validation.

The dataset contains 17 features, among them 10 are categorical features and 7 are numerical features. For our model buildup, the "Yes", "Male", "ICU", "High" values are replaced with "1" and the "No", "Female", "Word", "Low" values are replaced with "0". After this value transformation, the dataset comprised of 14 numerical features and 3 categorical features (Primary_Disease, Last_Feedback, B/P).

C. ML Models Selection:

The ubiquitous monitoring of ICU patients has generated a wealth of data which presents many opportunities but also great challenges. The following algorithms are chosen as the base predictors for our models to convey the optimum accuracy.

1) Naïve Bayes: The naïve Bayes algorithm is constructed based on the Bayes theorem. In our model, this algorithm is used as information retrieval [16]. Naive Bayes retrieves correlation among two conditional probability for each class given each 'x' value. In this case, all the correlation between features like pulse, admitted_to, ventilator is obtained with Naïve Bayes

Table 1. Sample Dataset

Patient_ No	Age	Sex	Weight	Primary_Disease	Admitted _to	Ventilator	Internal_ Bleeding	Last_ Feedback
1	13	Male	23	Unknown	Word	No	No	Normal
2	37	Female	56	Kidney Diseases	ICU	Yes	No	Increasing
3	62	Male	60	Kidney Diseases	Word	No	No	Normal
4	45	Male	81	Stroke	ICU	Yes	Yes	Decreasing
Systole	**Diastole**	**B/P**	**Send_SMS**	**Glucose_Level**	**Oxygen_ Supply**	**Pulse**	**Blood_ Circulation**	
120	80	Normal	No	High	0	75	Normal	
145	90	High	No	Low	80	78	Low	
110	70	Low	No	High	0	72	Normal	
150	95	High	Yes	High	100	80	Low	

2) Logistic Regression: Logistic Regression is a widely used algorithm for classification problems. Predictive algorithms are built from “experience”, which constitutes data acquired from actual cases[17].“Send_SMS” column is classified for critical patients. Based on this prediction, CPMS sends the SMS according to the patient no.

3) Decision Tree Classifiers: Decision tree builds condition based on the independent variables and creates an n-height tree to classify targeted output[18]. for our dataset, to build an optimal decision tree, we will only consider mostly correlated variables for independent variables.

4) Neural Network Classifier: Neural Network is a supervised machine learning algorithm which process record at a time and learn by comparing the classification of that record. The dataset we obtained is evaluated by the feedforward neural network and the multilayer perceptron (MLP). The MLP flows through the function being evaluated from feature ‘X’, through the intermediate computations used to define function ‘f’, and finally to the output ‘Y’ [19].

5) Ensemble Methods: Ensemble methods are used in this paper to increase the accuracy of the models. Ensemble method is comprised of three methods- Stacking, Bagging and Boosting [20]. Ensemble Bagging of Random Forest, Extra Trees, K-Neighbors, SVC and Ridge Classifier is applied in the models.

D. Watson Machine Learning models:

IBM Watson provides flexible options to create and run ml models. A model can be created from the “Model Builder” option. To use any existing trained model from local storage or to import any ml models, the “From File” option is present. For our project, we will use Jupyter Notebook models by setting the training parameter as a manual.

E. Deployment of the models:

For ensuring the continuous evolution of our dataset on a large scale and in a selected time scale, we needed to set a deployment of the ml models. For our mobile app and automated dataset series, the “Batch Prediction” performed the best [21]. Batch Prediction deployment is auto-deployed in a selected time interval with all the models that we selected for training. For the deployment, the parameters are following-

Table 2. Deployment Connection

Input Connection	Output Connection
Type - blue mix cloud object storage	Type - dashdb

{
"firstlineheader": true,
"file_name": "REGISTRATION.csv",
"infer_schema": "1",
"file_format": "csv",
"type": "bluemixcloudobjectstorage"
}
	{
	"type": "dashdb",
	"tablename": "MLRESULT",
	"schemaname": "NDXXXXX",
	"writemode": "write"
	}

Once we selected the parameters for the deployment, the models auto-deployed in selected time series. We can evaluate the deployment from the “Status” section, if the Status is “Successful” then the deployment is running successfully and if not, we need to debug and fix the prominent errors to make it running.

F. Connecting the Android Application with IBM:

To make the output visible, more user-friendly and under stable, we introduced our mobile application CPMS. Which is for both doctor and patients relative. The app is only utilized as an output device. It is not for entering new data in the database.

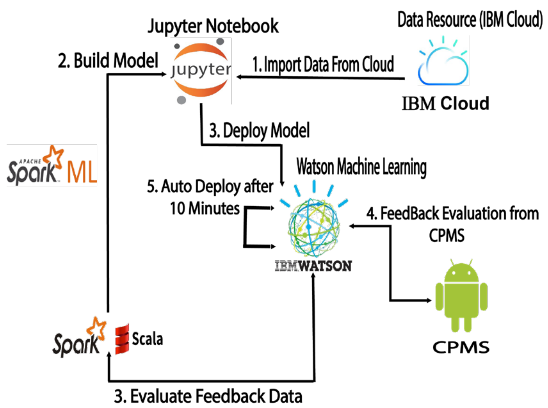

Fig 2. IBM Aggregation with CP

1) Collect dataset from IBM Cloud: The full app is based on a database from IBM Cloud. The admin will provide "Username" "password" "Hostname or IP Address" to connect After connecting the database from login to patient all will be displayed as an output of ml models.

2) Integration of SMS Sending System: In order to inform about the condition of a patient to doctors and patient relatives, we regulated an SMS system. For this, we connected an SMS API named "BD Bulk SMS". The API can send bulk SMS to any local operators. With the API we can send unlimited numbers of SMS from our system with negligible time difference because of using the IBM Watson SMS API gateway [22].

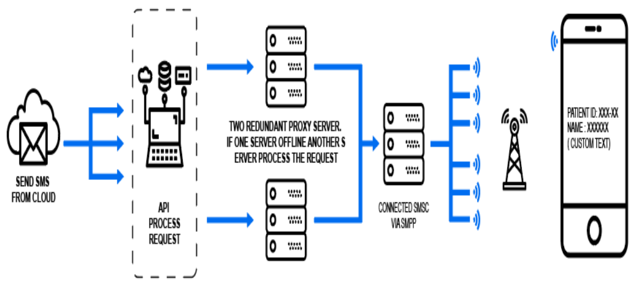

Fig 3. SMS API working process

VI. TRAINING AND TESTING OF ML MODELS

For setting up the parameters, all the models are trained in the Watson Studio and stored in the IBM Cloud. This auto deployment and training is the key concept of this proposed methodology.

1) Visualize the Dataset: Visualization is the core part to understand the dataset. The dataset contains mostly categorical data with "yes" and "no" values. Considering the data type, Bar chart, Histogram, and Correlation matrix are proposed to visualize the dataset [23]. An important finding from the "Send_SMS" feature is the rate of sending SMS for adequate diseases. From Fig. 4 bar chart, we can see the frequency of SMS sending in terms of the diseases.

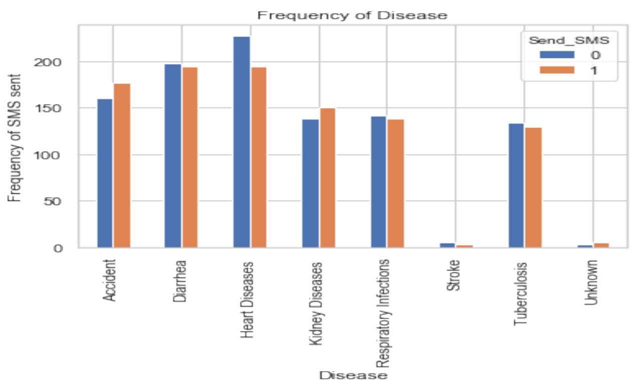

Fig 4. The frequency of Sending SMS in terms of Diseases.

The correlation matrix stated the important features selection process for choosing the highest correlated features [24]. The matrix used 2 kinds of data points, with only the numerical features and with numerical and categorical features. Fig. 5 shows the correlation matrix with both values.

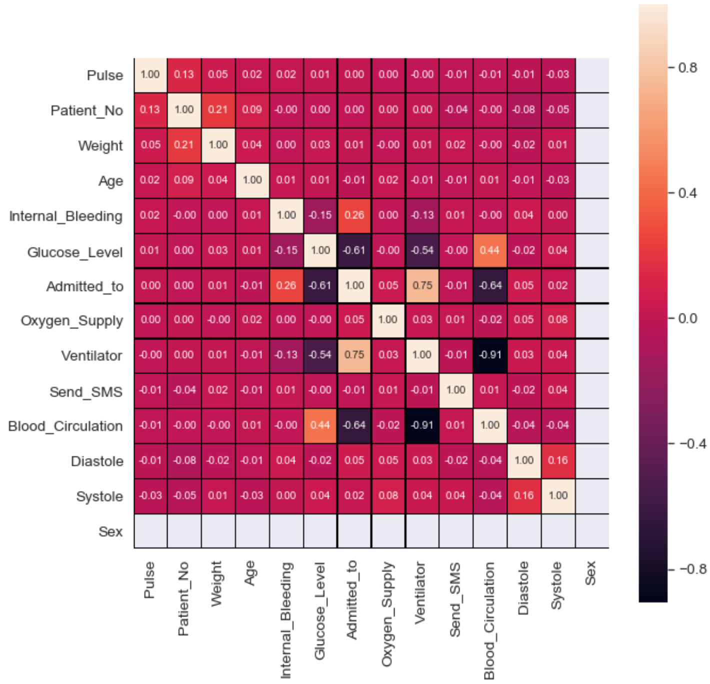

Fig 5. Correlation Matrix

2) Selecting and Training the Base Predictors: As we mentioned earlier, the ml models that are proposed in the paper are divided into two portions, the first evaluating algorithms are single base predictors and the second portion are stacked as the ensemble bagging method. Based on the dataset visualization, the paper suggests the following

- Bayes theorem states- $P(X|Y) = (P(Y|X) * P(X))/(P(Y))$ So, Naïve Bayes predicts the dependency between features X and Y and finds the inter-relation among the targeted variables [25].

- Logistic Regression Classifies the column "Send_SMS" whether to send the SMS or not. this classification is done based on the sigmoid function to normalize the values - $\frac{1}{1+e^{-x}}$, where x is the features or independent values [26]. This is a simple classification task but a very important one.

- Tree-Based Classifier creates the decision trees to find out the condition of the patient based on the following features – Age, Weight, Primary Disease, Ventilator, Last Feedback, Admitted to, B/P, Pulse, Systole, Diastole, and Internal Bleeding.

3) Model Parameters: Models parameter are configurable and these can be estimated from data. From our dataset, models parameter are set by tuning for delivering the best accuracy and performance [27]. The parameters and hyperparameters for our models are following –

- Logistic Regression: For the strong Regularization, we tested 3 C values. For C=1,100 and .001, the coefficient parameters are shown in Fig. 6.

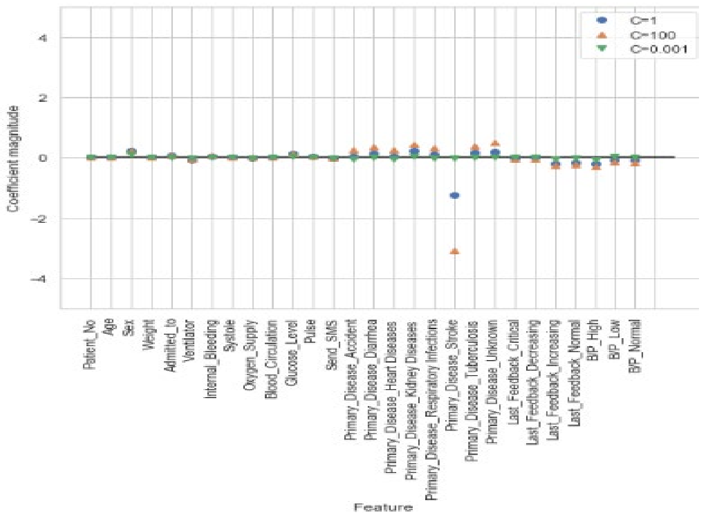

Fig 6. Coefficient Magnitude

- KNeighbors Classifier: The train and Test accuracy for n_neighbors range 1 to 10 is following –

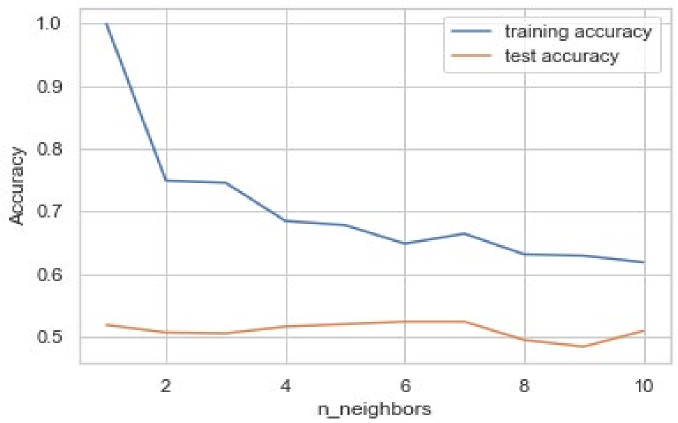

Fig 7. Train and Test accuracy of KNeighbors Classifier

Based on the graph, the optimum *n_neighbors = 6* is selected for the model [28] as at point 6, the error is optimum.

- Random Forest Classifier: Decision Tree classifiers performs well under optimum tree height [29]. From the Correlation Matrix (Fig. 5) we selected the most important features and for the Random forest. The features as shown in Fig 8 are combined for the model. From the Feature importance, the following are selected - *max_depth = 4, n_estimators = 100, random_state = 0.*

- Gradient Boosting Classifier: Grid Search evaluation of Scikit Learn is used here to turn the learning rate [30]. The following parameters were set after tuning with different grid - *random_state=0, max_depth=3.*

- MLP Classifier: MLP is sensitive to the dataset. Before training the data, built-in StandardScaler was applied for standardization and normalization of data [31]. The stats for standardization are - *copy=True, with _mean=True, with_std=True*.

  After the scaling, the training parameters are set after several tuning- *activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, learning _rate = 'constant', learning _rate_init =0.001, max_iter=1000.*

- Ensemble Methods: Ensemble Bagging procedure to a high-variance machine learning algorithm, so the parameters are defined based on the number of samples and hence the number of trees to include [32]. This parameter can be chosen by increasing or decreasing the number of trees on run after run until the accuracy begins to stop showing improvement. Table. 3 shows the parameters for ensemble bagging.

Table 3. Parameters for Ensemble Bagging algorithms

Algorithms	Parameters
Random Forest Classifier	max_samples=0.4, max_features=18, random_state =seed
Extra Trees Classifier	max_samples=0.25, max_features=20, random_ state=seed
KNeighbors Classifier	n_neighbors=5.
SVC	C=1.0, cache_size=200, class _weight= None, coef0 = 0.0, degree=3, gamma = 0.5, kernel='rbf', probability = False, scale_C = True, shrinking = True, tol = 0.001
Ridge Classifier	scorer=accuracy score, random_state=seed, Folds=10, verbose=2

4) Training, Testing, and Validation: The splitting of the dataset for train, test, validation done by considering 2 things -
1. The total number of samples in the dataset.
2. On the actual model which are training.

The models are weighted with many parameters and hyperparameters. So, the authors' suggested for cross-validation [33]. The number of K-folds deployed is 6. The Dataset splitting mentioned in Table 4.

Table 4. Split Ratio

Dataset	Ratio	Sample Number
Train	60	3000
Test	20	1000
Validation	20	1000

VII. RESULT TESTING AND ANALYSIS

As per the nature of this project, the authors had to follow the waterfall model to deliver each feature [34]. Firstly, we tested the IBM Cloud data entry process. In this process, several Cloud issues had to fix for avoiding data loss.

Table 5. Cloud Entry Testing

Service	Schema	Total Rows	Successful	Success Rate
NDX9073	Registration	2988	2788	93.30%
NDX9073	Patient Update	2800	2459	87.82%

IBM Cloud showed more than 90% data successfully converted from the terminal to cloud. From this data, the Watson studio carried out the machine learning model accuracy and validity.

Table 6. Machine Learning Models Accuracy (Base Predictors)

Algorithms	Precision	Recall	F1 score	Support
Naïve Bayes	0.92	0.90	0.87	24
Logistic Regression	0.90	0.88	0.86	20
KNeighbors Classifier	0.95	0.92	0.91	23
Decision Tree Classifier	0.91	0.91	0.90	21
Random Forest Classifier	0.92	0.93	0.92	22
Gradient Boosting Classifier	0.86	0.88	0.88	18
MLP Classifier	0.81	0.81	0.80	15

Table 7. Machine Learning Models Accuracy (Ensemble Methods)

Algorithms	Training Accuracy (%)	Testing Accuracy (%)
Random Forest Classifier	86	88
Extra Trees Classifier	90	89
KNeighbors Classifier	82	86
SVC	88	92
Ridge Classifier	91	90

Avg accuracy from the models varied from 80%-92%. For Validation of the models, we established the Confusion Matrix and the Receiver operating characteristic (ROC) curve.

Table 8. Validation Observation

Algorithm	Confusion Matrix
Naïve Bayes	[[50 2] [1 22]]
Logistic Regression	[[142 12] [10 48]]
Random Forest Classifier	[[12 20] [5 13]]

The ROC Curve was observed carefully, whether it could show the accurate areas under prediction. The ROC Curve brought out most of the areas under True Positive Rate.

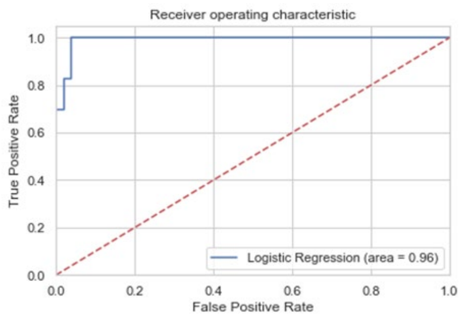

Fig 8. ROC Curve

Clustering with different Classes of values 1 and 0 can visualize the data points. Fig. 9 and Fig. 10 draws a clear scenario of clustering with different algorithms. Fig. 9 describes the clustering with all features and Fig. 10 illustrates clustering with only numerical features.

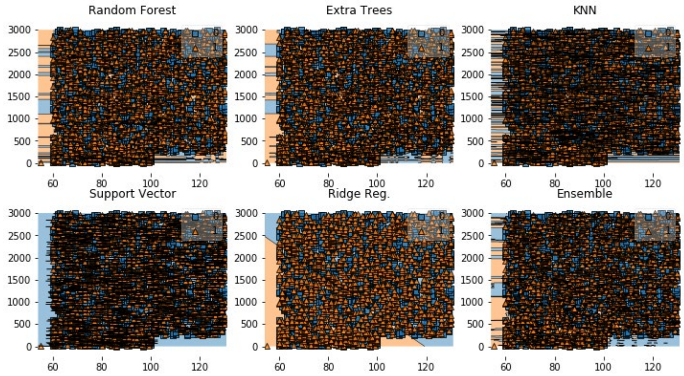

Fig 9. Clustering with all features

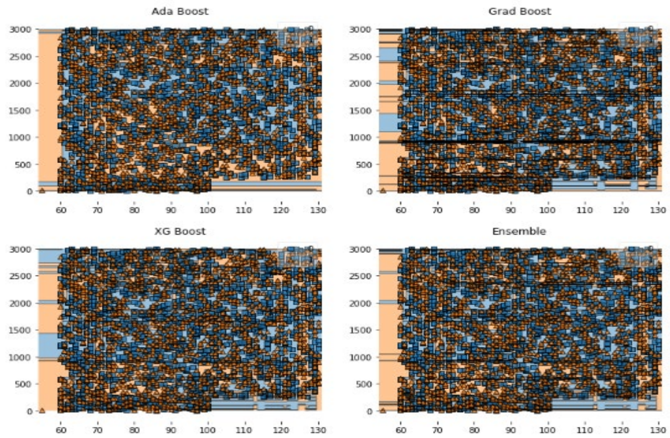

Fig 10. Clustering with only numerical features

Deploying the machine learning models was the last test to satisfy all the Watson Studio process. After deploying the models with a 1-hour interval, we wanted to see the result in person by examining in ICU. To do so, we collected live samples reading. After examining 12 hours we collected results from 120 patients with more than 2000 data, some of the results we got are illustrated in the Table. 9. Among the condition of 120 patients, 98 were predicted correctly.

Table 9. Hospital testing reports

Patient age	Primary Disease	Model Prediction	Actual Condition
56	Heart Disease	Decreasing	Decreasing
07	Unknown	Normal	Increasing
38	Stroke	Increasing	Normal
21	Accident	Decreasing	Decreasing
46	Kidney Diseases	Normal	Normal

As the predicting results showed satisfactory scenario, we moved on to test this results in predicting health condition and sending SMS task. Logistic regression predicted 560 SMS needed to be sent as per as the condition forecast, among them 485 was the valid situation and 48 times the model predicted wrong and 27 times the SMS sending was failed.

Table 10. SMS Testing

Total SMS	Sent	Right Prediction	Wrong Prediction
560	533	485	48

Nearly 90% accuracy rate also worked for this SMS API Testing process.

VIII. CONCLUSION

To provide better treatment we require more advanced technologies at very low cost. We started this project to bring out a good result in the hospitals to serve the patient. We used some of the existed techniques and technologies to give a new shape in the hospital and nursing sector. Most of the ml models accuracy varied from 80% to 92%. The lowest accuracy obtained is 80%. An important finding of this project is the appropriate uses of machine learning models for medical patients and categorical data manipulations. The IBM Cloud showed good promising actions by keeping more than 90% success rate. Altogether the results we obtained from our project and experiments are showing promise to rise this system in large scale for urban and low economical side peoples. With the help of this project, a virtual doctor can be established to serve the people better and monitor patients with appropriate care. This is also a decision-making assistant for the doctor as a smart health care system. As we have established this project with very few parameters of the physical segments, we can improve this project more by adding full parameters to measure the human body circulations. In the future, we are planning to install an embedded system to take a live reading from Ventilator, Medicine Pump, Heart Monitor, and other ICU machines. This will also increase the overall working accuracy of this project.

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.

[3] "Rational Unified Process", URL: [online] Available: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_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/vital-signs-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. i-Manager'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/us-en/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://jupyter-notebook-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_gettingstarted.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.

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