Remote Health Monitoring System with Analytics Dashboard

By

Sisira Kumar Kapat

Lecturer (CSE)
UCP Engineering School,
Berhampur, Odisha

Table of Contents

1. Introduction

1.1 Overview

Remote health monitoring system plays a vital role especially in Covid-19 pandemic period. This project addresses the current demand of wearable which could be developed further to function and meet the need of patients effectively. The user enters to the system by providing his/her name, gender and age. Different IBM IoT sensors are associated with the system to measure different health parameters such as temperature, blood pressure, pulse of the user. Result is being generated in accordance with the age and different health parameters.

Utilization of sensors decreases the possibility of human mistake, ensures better care and treatment, reduces medical expenses, lessens the involved space of the room and improves overall performance [2]. This system is much practical in maintaining social distancing and to avoid spread of the Covid-19 or such contagious diseases.

1.2 Purpose

The purpose of this project is to use ICT in healthcare. Bustling time schedule and unpredictable situations of life increases the probability of health risk, independent of the age of a person. Though we cannot replace the healthcare system with this, but this project ultimately supplements the existing healthcare system.

2. Literature Survey

2.1 Existing Problem

There are many problems related to health care system, exist in current era. Some of the problems are described as follows.

People are very much busy. Most of the time people don't go to hospital, simply because they have to maintain a queue. The denial will increase the health issue and subsequently results into a health hazard.

The numbers of health professionals are also limited, which increases the personal overhead to each health professionals. For an instance, a single doctor can check or treat up to a certain number of patients in specific time duration.

Patients who met with an accident, patients at the time of child-birth (delivery cases) or such patients in critical condition need emergency attention. A wearable can be much helpful in observing the status of such patients. The doctor can observe remotely to such patients till they arrive the hospital.

2.2 Proposed Solution

In a solution to existing problem, we can integrate the iot devices or sensors at the frontend of the system such as a handheld device or a wearable, which can collect the health parameters such as temperature, BP and pulse of the user. Then the IoT data along with the age of the patient/user is sent to the cloud. A machine learning algorithm evaluates the parameters at the backend of the system which is deployed in the cloud. After evaluation, the result is shown to the user by means of the user interface. This improvises the speed and accuracy of the existing traditional method of health check-up.

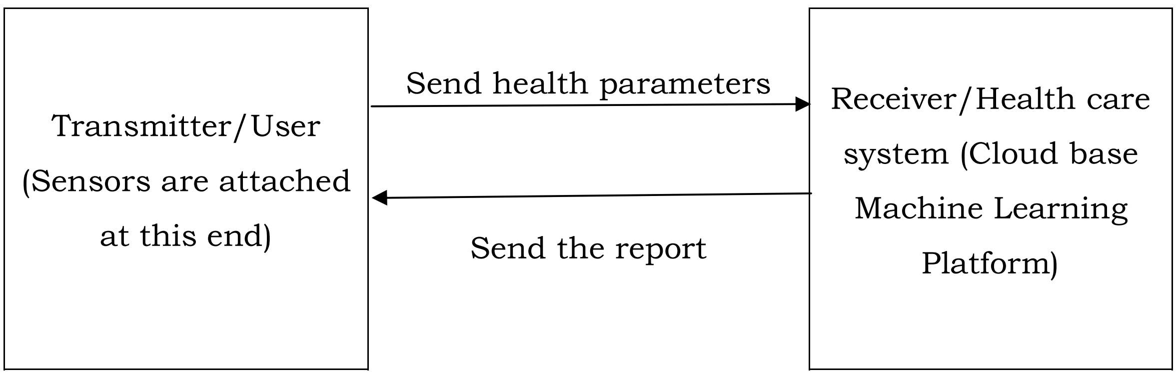


Figure-1 : Proposed Solution

3. Theoretical Analysis

The system uses some sensors based on the requirement. The number of sensors and type of sensors depends purely upon the purpose of the system. For an instant, a doctor uses thermometer to measure the temperature of a patient to confirm whether he is suffering from fever or not. Similarly diseases are confirmed by analyzing some health parameters like age, temperature, blood pressure etc. Sometimes X-Ray, ultrasound are needed for analyzing some critical conditions.

Mathematically we can represent,

$$Health\ Status\ Prediction\ (P) = f\left(\sum_{i=1}^{n} Parameter_i\right)$$

The user must enter his/her name, gender and age in the user interface. Name is a required parameter to the system to identify the person. Since there are some diseases which are specific to an age group, so age is a required parameter. Similarly some diseases are specific to a particular gender group, hence gender is also a required parameter in the system. After entering the three parameters, the system will collect the health parameters via IoT sensors.

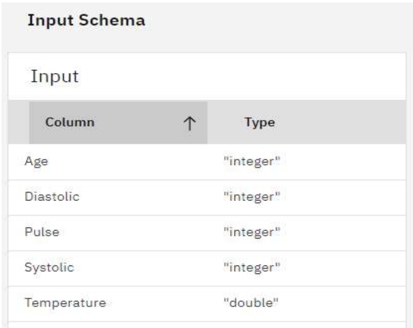
3.1 Block Diagram

The system is composed of three components such as,

i. User interface to enter name, gender and age
ii. IoT interface to collect the health parameters
iii. Machine learning model in the cloud to evaluate the data

IoT interface plays a vital role in the system to collect the health parameters. The Temperature sensor is used to collect the temperature in Celsius of the user. The Blood pressure sensor is used to collect both systolic and diastolic blood pressure of the user. The Pulse sensor is used to collect the pulse status of the user. The collected data are sent to the cloud database for analysis.

We have designed a health monitoring machine learning model, which is highly essential for decision making. We have used the Health monitoring dataset for training and testing the model. Our dataset contains 1015 records with five attributes; out of which 90% is used for training and 10% is used for testing.. The input schema is represented in figure-1. AutoAI experiment with Multi class classification is used for the experiment.


Input Schema

Input

Column	Type
Age	"integer"
Diastolic	"integer"
Pulse	"integer"
Systolic	"integer"
Temperature	"double"

Figure-1

The complete block diagram of the system is presented in figure-2.

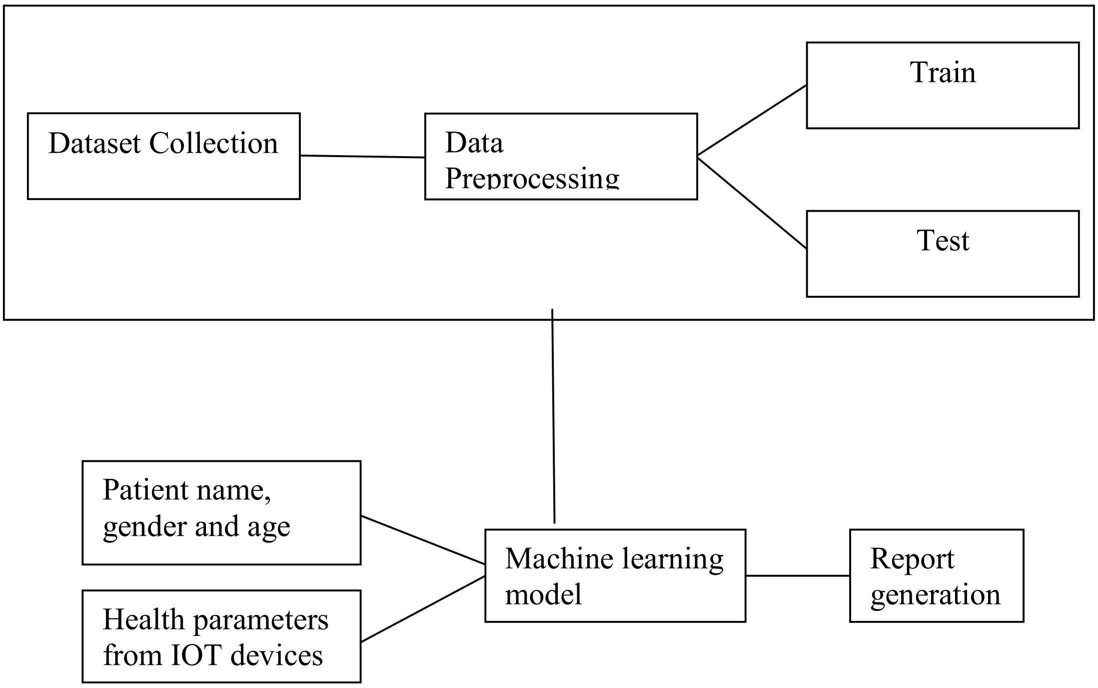


Figure-2

3.2 Hardware/ Software Designing

The system consists of some hardware as well as some software part. The IoT sensors which collect the user health parameters are the hardware devices. In other hand, the user interface to enter user data and the user interface to visualize the health report are webpages.

To design the whole system, we have used the web based resources of IBM. 'Internet of Things Platform' is used to simulate the IoT sensors. These sensors generate data and send to the cloudant database.

A web based health monitoring system is designed. We have used the node-RED service of IBM to design the user interface. The user interface is compatible with computer systems as well as mobile devices which uses android operating system.

The Machine Learning service is used in Watson Studio of IBM is used to implement the machine learning model. The model is deployed in the cloud space. The model is generated with a hybrid combination of algorithms. The pipeline with higher accuracy is saved as the model, for instance Decision tree classification provides highest optimum accuracy in the pipeline. The model has to be promoted to space for instant use of the user. A deployment space is needed to promote the model in the space. We have created a deployment space namely 'Health Monitor' to deploy the model.

4. Experimental Investigations

The dataset contains 1015 number of records with five attributes, out of which 90% is used for training and 10% of the data is used for testing. The dataset consists of five attributes; each attribute has a lower limit and higher limit of values. The values are presented in the following table.

Attribute	Range
Age	0 to 80
Temperature	0 to 41
BP (Cistole)	0 to 200
BP (Distole)	0 to 140
Pulse	0 to 200

Table-1

We have used Multiclass classification algorithm for the experiment. Here we have performed two types of experiment such as 'cross validation' and 'Houldout'.

In cross validation the whole dataset is divided into 'n' number of parts. Then one part is set as the test set and rest are used as the training set. The process continues till the completion of each unique combination of test set. Holdout is a simplest type of cross validation where the whole dataset is divided into two groups such as train set and test set. Train set is used for training the model, and test set is used for validation.

Here, eight pipelines are generated using both cross validation and holdout method. Since Holdout is the simple form of cross validation, we have considered cross validation as our base method for experiment.

The relationship map for cross validation is depicted in the figure below.

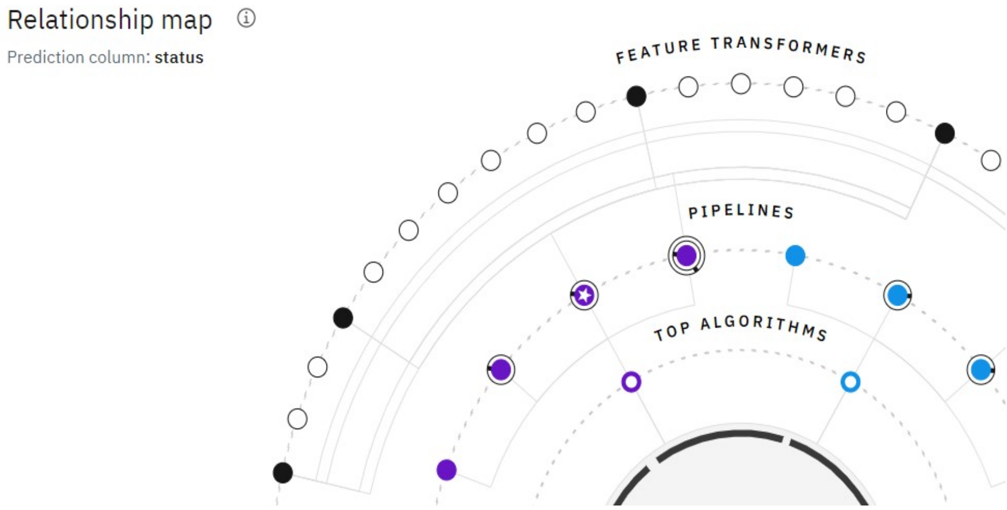


Figure-3 : Relationship map for cross validation

The progress map for cross validation algorithm of the eight pipelines is shown in figure below.

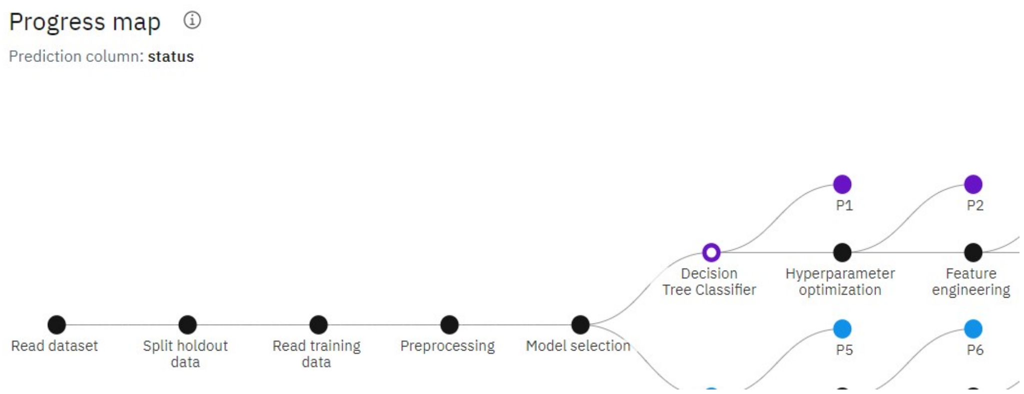


Figure-4: Progress map for cross validation

Out of the eight pipelines generated from the cross validation experiment, we have saved the model with highest accuracy. Here the pipeline-3 with Decision Tree classification gives highest accuracy, so we have saved the pipeline as the model. The eight pipelines are shown in figure-5, where the accuracy and time to build the model is mentioned with the pipeline.

Pipeline leaderboard

Rank ↑	Name	Algorithm	Accuracy (Opt...	Enhancements
★ 1	Pipeline 3	Decision Tree Classifier	**0.959**	HPO-1 FE
2	Pipeline 4	Decision Tree Classifier	**0.959**	HPO-1 FE HPO-
3	Pipeline 1	Decision Tree Classifier	**0.958**	*None*
4	Pipeline 2	Decision Tree Classifier	**0.958**	HPO-1
5	Pipeline 7	Random Forest Classifier	**0.953**	HPO-1 FE
6	Pipeline 8	Random Forest Classifier	**0.953**	HPO-1 FE HPO-

Figure-5 : Pipeline leaderboard showing accuracy and build time

The accuracy of the classifier is the probability of correctly classifying the records in the test dataset. It is represented mathematically as,

$$Accuracy \ = \frac{True\ Positive + True\ negative}{Total}$$

In multi-class classification, the true positive is the sum of all the true positive case of all the pipelines and similarly false positive, false negative and true negative is calculated.

The detailed accuracy of the eight pipelines of cross validation, which are generated by using AutoAI are depicted in the figure-6. The detailed accuracy is presented on the basis of precision, recall and F-measure of the experiment. The precision, recall and F-measure are basic characteristics of any machine learning algorithm.

The precision of the classifier is the probability of records actually being in a class if they are classified to be in that class. It is represented mathematically as,

$$Precision \ = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

The recall of the classifier is the probability that a record is classified as being in a class if it actually belongs to that class. It is represented mathematically as,

$$Recall \ = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

The F-measure is the harmonic mean of precision and recall. It is computed mathematically as,

$$F - measure \ = \frac{2 \times Recall\ \times Precision}{Recall + Precision}$$

Pipeline leaderboard

Rank	Name	Algorithm	Accuracy (Optimiz...	F1 macro	F1 micro	F1 weighted	Log loss	Precision m...	Precision m...	P
★ 1	Pipeline 3	Decision Tree Classifier	0.959	0.930	0.959	0.959	1.400	0.939	0.959	0
2	Pipeline 4	Decision Tree Classifier	0.959	0.930	0.959	0.959	1.400	0.939	0.959	0
3	Pipeline 1	Decision Tree Classifier	0.958	0.930	0.958	0.958	1.437	0.937	0.958	0
4	Pipeline 2	Decision Tree Classifier	0.958	0.930	0.958	0.958	1.437	0.937	0.958	0
5	Pipeline 7	Random Forest Classifier	0.953	0.923	0.953	0.952	0.315	0.946	0.953	0
6	Pipeline 8	Random Forest Classifier	0.953	0.923	0.953	0.952	0.315	0.946	0.953	0

Figure-6: Pipeline leaderboard showing detailed accuracy

The comparative analysis is obtained from the metric chart using cross validation is presented in figure-7. The comparison is essential to identify the appropriate pipeline, which should be the model. The comparison should be on the basis of detailed accuracy obtained from the experiment.

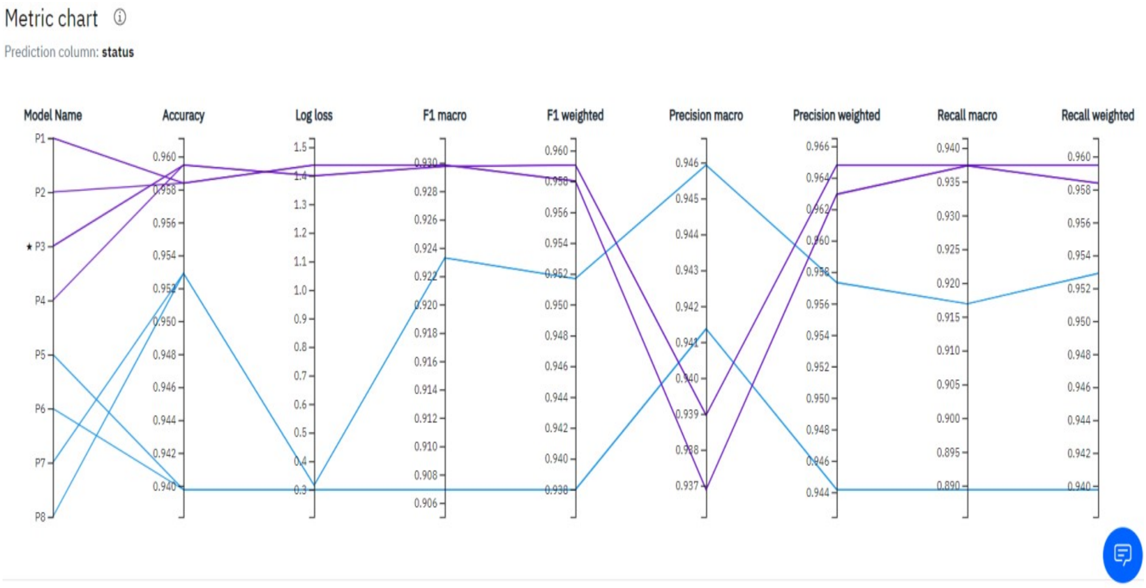


Figure-7: Comparative analysis of different pipelines of cross validation

The comparative analysis obtained from the metric chart using holdout method is presented in figure-8. Sometimes holdout algorithm performs well. So we should also observe the pipelines of holdout algorithm.

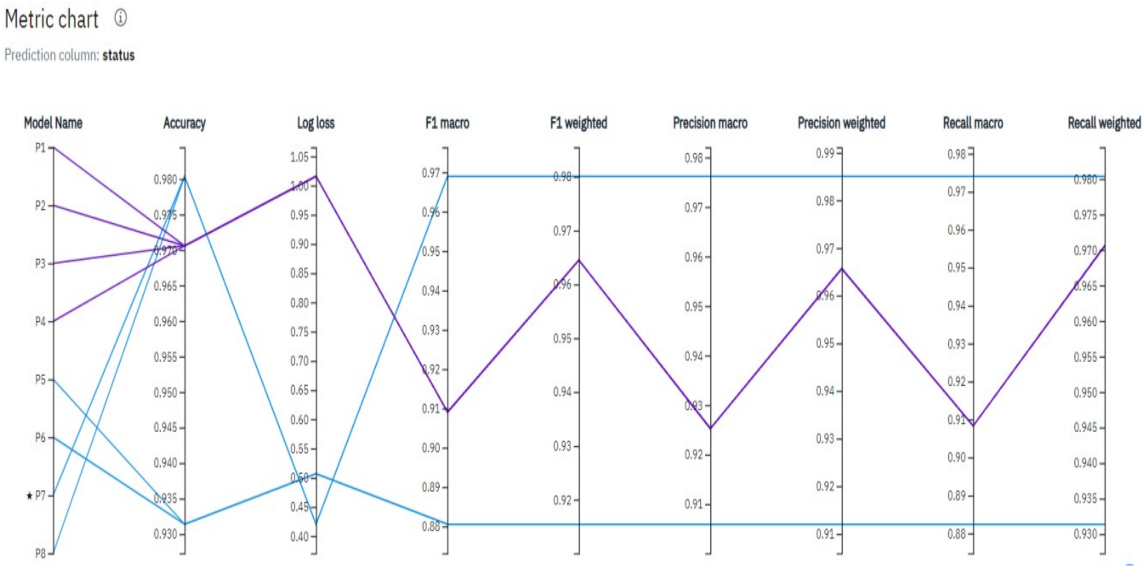


Figure-8: Comparative analysis of different pipelines of Holdout

The experiment requires the iot platform to generate health parameters. The health parameters are saved in the cloudant database. The health parameters will be sent to the machine learning platform which is deployed in the cloud space. Evaluation is being carried out in the backend and the result is generated. We have used nodered service to design a web based platform which could show the health status report.

5. Flowchart

The system consists of three components as mentioned in section 3.1. The user interface collects user name, gender and age from the user. This is a manual entry to the system. The IoT interface is a hardware interface which collect the health parameters automatically from the user. So the complete input to the system will be name, gender, age along with the IoT sensor data. Now the data is sent to the machine learning platform for evaluation. If the data is not accepted by the model, then error report is generated otherwise the system will generate the health report which will be sent to the user by means of the user interface. The complete flowchart of the system is presented in figure-9.

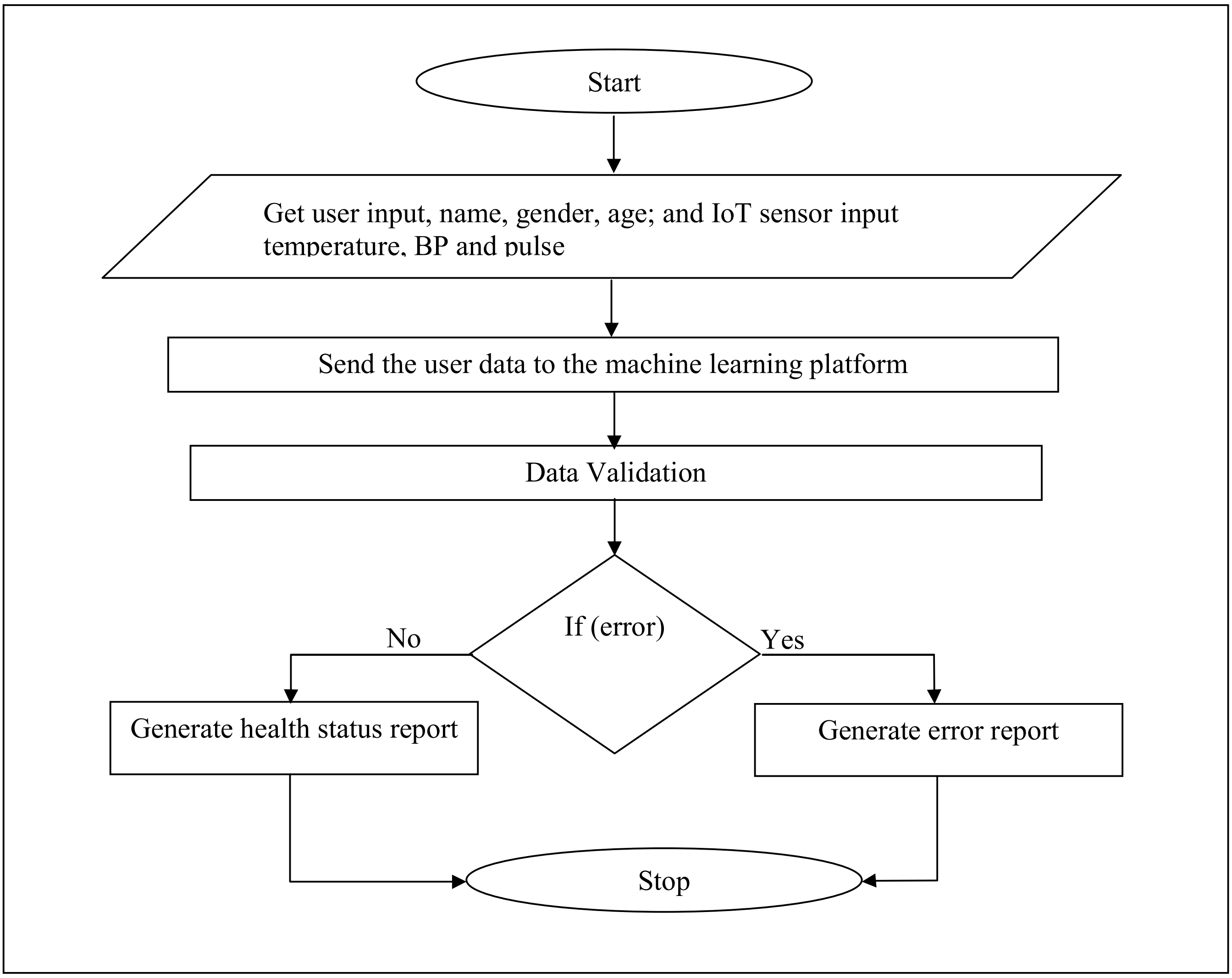
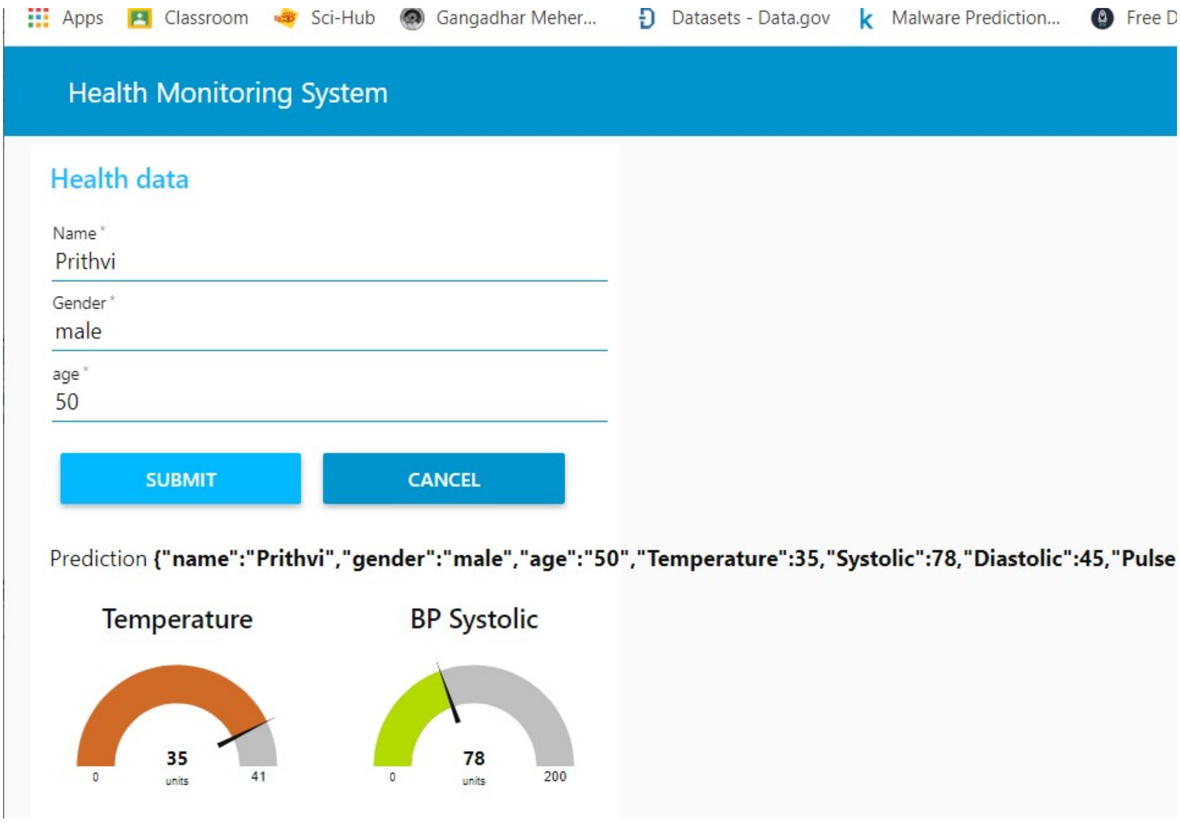


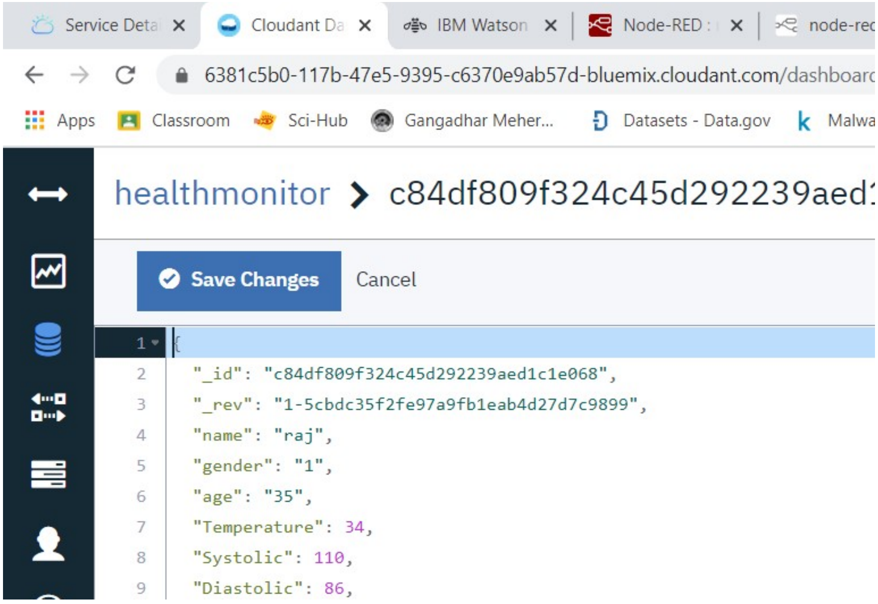
Figure-9 : Flow chart of the system

6. Result

The result of the system is the status report. Here two components are there; one is manual entry, and other is automatic. The manual entry part contains a form where the user enters name, gender and age. There are four sensors used here namely temperature, BP Cistole, BP Distole and pulse. The sensors collect the user health parameters and combines with the form input. Then the combined data is sent to the cloud machine learning platform for prediction.



The data is also saved in the cloudant database as shown in the figure below. This process is useful for observation of critical patients.



7. Advantages and Disadvantages

7.1 Advantages

The following advantages are achieved by using this project.

i. Quick access by means of the mobile devices.
ii. Avoid contagious diseases, since this platform avoids rush.
iii. Daily or hourly status of a patient can be maintained for observation. This facility can be helpful to keep track of the patients who are in critical condition.
iv. Utilization of sensors decreases the possibility of human mistake.
v. Ensures better care and treatment.
vi. Reduces medical expenses by reducing the travelling frequency.
vii. Lessens the involved space of the room and improves overall performance

7.2 Disadvantages

The following are the points which may be considered as the limitation of the project.

i. Some of the diseases require privacy and confidentiality. If the database got exposure to the public, then it violates the right to privacy of a person.
ii. We cannot replicate the healthcare system, no matter how efficient this system is, sometimes the person has to consult the health care professional.

8. Applications

This project can be used in handhold devices and/or the wearable which is much closure to any person. This can also be used in web based monitoring.

9. Conclusion and future scope

The health care system has to be improvised with the evolution and revolution of engineering and technology. This will create a positive environment to use the ICT in health care system. In this system the overall accuracy 95.9%, which holds good in the prediction but the overall accuracy of the system has to be improved in future..

The remote health monitoring system with analytics dashboard surely helps but some of the diseases which require the privacy and confidentiality of the patient has to be taken care. This can hamper the right to privacy in some extent which has to be taken care.

10. Bibliography

[1] Ananda Mohon Ghosh, Debashish Halder, SK Alamgir Hossain, "Remote Health Monitoring System through IoT", 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp 921-926, 2016

[2] Dahlia Sam, S.Srinidhi, V. R. Niveditha, S.Amudha, D. Usha, "Progressed IOT Based Remote Health Monitoring System", International Journal of Control and Automation, Vol. 13, No. 2s, pp. 268-273, 2020

[3] Ngo Manh Khoi, Saguna Saguna, Karan Mitra, Christer Ahlund, "IReHMo: An Efficient IoT-Based Remote Health Monitoring System for Smart Regions", https://www.diva-portal.org/smash/get/diva2:1005647/FULLTEXT01.pdf

[4] Mohd. Hamim, Sumit Paul, Syed Iqramul Hoque, Md. Nafiur Rahman, Ifat-Al-Baqee, "IoT Based Remote Health Monitoring System for Patients and Elderly People", International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 978-1-5386-8014-8/19, IEEE, 2019

11. Source Code

A. **The source code to simulate sensors is as follows.**

Payload

Specify the event payload in the editor window or by uploading a CSV fil

```
{
    "temperature": random(0, 41),
    "pulse": random(0,200),
    "BPsystolic": random(0,200),
    "BPdiastolic": random(0,140)
```

B. **The endpoint for the machine learning model is,**

https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9ca2433b-f932-4703-8a7b-035c603d1593/predictions?version=2020-10-19

C. **The python code for the machine learning model is,**

```
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
token_response = requests.post('https://iam.ng.bluemix.net/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
```

```
# NOTE: manually define and pass the array(s) of values to be scored in
the next line
payload_scoring = {"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}

response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/9ca2433b-f932-4703-8a7b-
035c603d1593/predictions?version=2020-10-19', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
```