**PROJECT REPORT**

**1. HEALTH MONITORING SYSTEM**

**1.1 Introduction**

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.

Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think. AI is a study of how human brain think, learn, decide and work, when it tries to solve problems. And finally this study outputs intelligent software systems. The aim of AI is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving.

The intelligence is intangible. It is composed of

* Reasoning
* Learning
* Problem Solving
* Perception
* Linguistic Intelligence

**1.2 Objective Of Research**

The main objective of health monitoring system is to measure patient body temperature and heartbeat rate and these also mainly involves in monitoring the condition of the patient. The objective considers the patient health conditions and so on, based on these criteria's prediction is evolved. It also includes the survival of the patient whether they can spend more on theirnecessities or not. The other objective of prediction is to know the status of the patient and to know the ability of a person. The output of this prediction tells us that whether the patient is healthy or less healthy based on the available parameters.

**1.3 Problem Statement**

Thehealth monitoring can provide useful physiological information in the home. This monitoring is useful for elderly or chronically ill patients who would like to avoid a long hospital stay. It is used to collect the data automatically and transmit the signals to the other telecommunicational devices. In this project we are choosing appropriate sensors according to what you would like to detect and design algorithms to realize your detection.

**Example**: The detection of a fall of a person, monitoring cardiac signals, brain signal.

Health monitoring system was designed that extends healthcare from the traditional clinic or hospital setting to the patient's home. The system was to collect a heartbeat detection system data and a fall detection system data.

During design the following characteristics of the future medical applications were adhered to:

a) Integration with current trends in medical practices and technology.

b) Assistance to the elderly and chronic patients. The device should be easy to use with minimal buttons.

**1.4 Industry Profile**

Industry profile are in-depth documents that give insight into an where it came from, and where it appears to be going. A typical report looks at the industry leaders, forces affecting the industry and financial data for the industry.

Health monitoring is repeatedly mentioned as one of the main application areas for pervasive computing. It is the application of mobile computing technologies for improving communication among patients, physicians, and other health care workers we use the health monitoring system. These nodes capable of sensing, processing, and communicating one or more vital signs, can be seamlessly integrated into wireless personal or body area networks for health monitoring system. The present invention is directed to a health monitoring system and monitoring the supervision of outpatient vital signs using videoconferencing techniques.

Here we use Logistic Regression algorithm for our prediction. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

**2. REVIEW LITERATURE**

The health monitoring system have been proposed by (vidmar in the year 1990), where patients are monitored by means of a clinical protocol, based on periodical measurements through biomedical transducers of some physiological parameters related to the disease. Typically, the acquired data of some physiological parameters are sent via the internet to a remote centre, where clinicians analyze the data and make decisions. Patients data are sent through e-mail and physiological signals over the internet.

The health monitoring task is achieved by telemedicine (enabling medical information exchange as the support to distant decision making) and telemonitoring (enabling simultaneous distant monitoring of patient and his vital function) that monitors the changes in ECG signals and provide feedback. On the personal server module, visualization and analysis software are implemented. It displays the patient’s parameters received from the sensors and sends the commands and medical decisions.

There are many advantages of using this type of sensor. It can be used for broader range of patients and medical professionals and those people living in rural or isolated regions. It reduces cost and time. It provides the best health care and flexible capable of measuring, pre-processing and transmission ECG info to personal server. This type of sensor can be used for those patients’ that has a longer time span than normal span and the proposed system could also be used as a warning system for monitoring during normal activity or physical exercise. It specifically targets assisted-living residents and others who may benefit from continuous health monitoring system. It presents best practices in wireless sensor network design for health care applications. Based on the most important aspects like power efficiency and security which guide the development of a wireless sensor network based applications. The sensor boards handle acquisition of physiological signals and pre-processing. For example, the ISPM samples three independent accelerometer axes each at a rate of 200 Hz. The raw accelerometer data is filtered and pre-processed. The filtering includes moving an average filter to eliminate high frequency movement artifacts, and separation of low and high frequency components of the acceleration signal. Sensor orientation can be calculated as the angle between low frequency accelerometer components.

Thus, also the ongoing miniaturization allows building tiny computers that are able to observe all kinds of physical phenomena. Structural Health Monitoring (SHM) has the potential to dramatically reduce lifecycle costs, safe useful life of a structure, shrink insurance costs, decrease the need for expensive repairs, and circumvent complete failure of the monitored structure. A portable real-time wireless health monitoring system also provides an accurate result which is used for remote monitoring of patients’ pulse rate and oxygen saturation in blood. The system was designed and implemented using ZigBee wireless technologies.

The basic wireless technologies which currently healthcare systems are RFID, Bluetooth, ZigBee and wireless sensor network which gives innovative medium for data transmission in the field of medicine. One of many applications in medical domain is computer assisted physical rehabilitation. Intelligent sensors used by patients transmit vital signs to personal server sequentially, the data is transmitted from personal server to servers of the healthcare system, such as weather forecast, medical database or emergency server over Internet. Algorithms may be executed on the healthcare system servers to give instant and patient-specific recommendations

**3. DATA COLLECTION**

A data set (or dataset) is collection of [data](https://en.wikipedia.org/wiki/Data). Most commonly a data set corresponds to the contents of a single [database table](https://en.wikipedia.org/wiki/Table_(database)), or a single statistical [data matrix](https://en.wikipedia.org/wiki/Data_matrix_(multivariate_statistics)), where every [column](https://en.wikipedia.org/wiki/Column_(database)) of the table represents a particular variable, and each [row](https://en.wikipedia.org/wiki/Row_(database)) corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows.

The Attributes that are used in our dataset are:

* **Systolic Blood Pressure (sbp):** It measures the pressure in your blood vessels when your heart beats.
* **Diastolic Blood Pressure (dbp):** It measures the pressure in your blood vessels when your heart rests between beats.
* **Pulse:** It represents the tactile arterial palpation of the heartbeat by trained fingertips. The pulse may be palpated in anywhere at any time.
* **Temperature:** It refers average temperature of the patient.
* **Level:** It tells us that different levels. Based on that, we determine whether the person is healthy or unhealthy.

**4. METHODOLOGY**

**4.1 Exploratory Data Analysis**

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

The objectives of EDA are to:

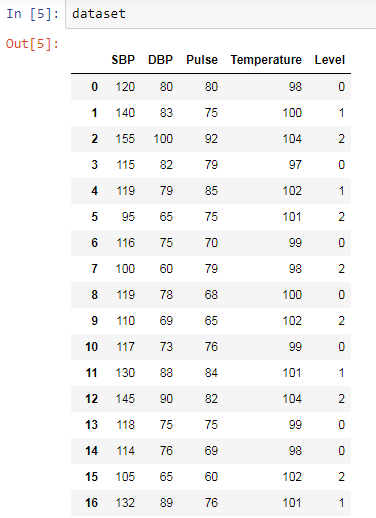
• Suggest hypotheses about the causes of observed phenomena.

• Assess assumptions on which statistical inference will be based.

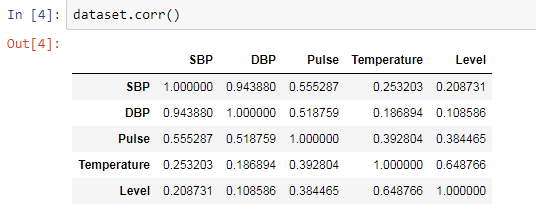
• Support the selection of appropriate statistical tools and techniques.

• Provide a basis for further data collection through surveys or experiments.

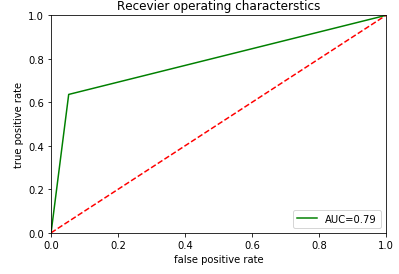
**4.1.1 Figures and Tables**



**Fig 4.1.1.1: Dataset of Health Monitoring System**



**Fig 4.1.1.2: Correlation Analysis**



**Fig: 4.1.1.3 AOC Curve**

**4.2 Statistical Techniques and Visualization**

Statistics is a collection of tools that you can use to get answers to important questions about data. You can use descriptive statistical methods to transform raw observations into information that you can understand and share. You can use inferential statistical methods to reason from small samples of data to whole domains. Statistics is a pillar of machine learning. You cannot develop a deep understanding and application of machine learning without it.

* Problem Framing: Requires the use of exploratory data analysis and data mining.
* Data Understanding: Requires the use of summary statistics and data visualization.
* Data Cleaning. Requires the use of outlier detection, imputation and more.
* Data Selection. Requires the use of data sampling and feature selection methods.
* Data Preparation. Requires the use of data transforms, scaling, encoding and much more. Model Evaluation. Requires experimental design and resampling methods.
* Model Configuration. Requires the use of statistical hypothesis tests and estimation statistics. Model Selection. Requires the use of statistical hypothesis tests and estimation statistics. Model Presentation. Requires the use of estimation statistics such as confidence intervals. Model Predictions. Requires the use of estimation statistics such as prediction intervals.

NumPy is a commonly used Python data analysis package. By using NumPy, you can speed up your workflow, and interface with other packages in the Python ecosystem, like scikit-learn, that use NumPy under the hood. NumPy was originally developed in the mid 2000s, and arose from an even older package called Numeric. This longevity means that almost every data analysis or machine learning package for Python leverages NumPy in some way.

We’ll walk through using NumPy to analyze data on wine quality. The data contains information on various attributes of wines, such as pH and fixed acidity, along with a quality score between 0 and 10 for each wine. The quality score is the average of at least 3 human taste testers. As we learn how to work with NumPy, we’ll try to figure out more about the perceived quality of wine.

Numpy 2-Dimensional Arrays with NumPy, we work with multidimensional arrays. We’ll dive into all of the possible types of multidimensional arrays later on, but for now, we’ll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with.

Creating A NumPy Array: We can create a NumPy array using the numpy.array function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. Because we want all of the elements in the array to be float elements for easy computation, we’ll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats. In the below code, we:

* Import the numpy package.
* Pass the list of lists wines into the array function, which converts it into a NumPy array.
* Exclude the header row with list slicing.
* Specify the keyword argument dtype to make sure each element is converted to a float.

Indexing NumPy Arrays: We now know how to create arrays, but unless we can retrieve results from them, there isn’t a lot we can do with NumPy. We can use array indexing to select individual elements, groups of elements, or entire rows and columns. One important thing to keep in mind is that just like Python lists, NumPy is zero-indexed, meaning that the index of the first row is 0, and the index of the first column is 0. If we want to work with the fourth row, we’d use index 3, if we want to work with the second row, we’d use index 1, and so on.

Pandas is an open source python library that is built on top of NumPy. It allows you do fast analysis as well as data cleaning and preparation. Pandas is hands down one of the best libraries of python. It supports reading and writing excel spreadsheets, CVS's and a whole lot of manipulation. It is more like a mandatory library you need to know if you’re dealing with datasets from excel files and CSV files i.e. for Machine learning and data science. This is part one of Pandas tutorial. I’m not going to cover everything possible with pandas however, I want to give you a taste of what it is and how you can get started with it. This tutorial is going to be super short just introducing you to Series object of pandas.

As other libraries, you’d import pandas and reference it as pd.

import pandas as pd

**pd.Series()**

pd.Series() is a method that creates a series object from data passed. The data must be defined as a parameter. What is a “Series” object in Pandas? It is a data structure defined by Pandas. Basically it looks like a table having rows and columns. Notice that these numbers on the first column were added automatically by pandas. They serve as index. These variables are known as categorical variables and in terms of pandas, these are called ‘object’.

To retrieve information using the categorical variables, we need to convert them into ‘dummy’ variables so that they can be used for modelling. We do that using pandas.get\_dummies feature.

First we create a list of the categorical variables. Then we convert these variables into dummy variables. We have created dummy variables for each categorical variables and printing out the head of the new data-frame. You can understand, how the categorical variables are converted to dummy variables which are ready to be used in the modelling of this data-set. But, we have a slight problem here. The actual categorical variables still exist and they need to be removed to make the data-frame ready for machine learning. We do that by first converting the column headers of the new data-frame to a list using tolist() attribute. Then we create a new list of column headers with no categorical variable and rename the headers.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib. Matplotlib was originally written by John D. Hunter, has an active development community, and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell. As of 23 June 2017, matplotlib 2.0.x supports Python versions 2.7 through 3.6. Matplotlib 1.2 is the first version of matplotlib to support Python 3.x. Matplotlib 1.4 is the last version of Matplotlib to support Python 2.6. Matplotlib has pledged to not support Python 2 past 2020 by signing the Python 3 Statement. Pyplot is a Matplotlib module which provides a MATLAB-like interface.

Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source. Several toolkits are available which extend Matplotlib functionality. Some are separate downloads, others ship with the Matplotlib source code but have external dependencies.

* Basemap: map plotting with various map projections, coastlines, and political boundaries.
* Cartopy: a mapping library featuring object-oriented map projection definitions, and arbitrary point, line, polygon and image transformation capabilities. (Matplotlib v1.2 and above)
* Excel tools: utilities for exchanging data with Microsoft Excel
* GTK tools: interface to the GTK+ library.

Visualization with Matplotlib: One of Matplotlib’s most important features is its ability to play well with many operating systems and graphics backends. Matplotlib supports dozens of backends and output types, which means you can count on it to work regardless of which operating system you are using or which output format you wish. This cross-platform, everything-to-everyone approach has been one of the great strengths of Matplotlib. It has led to a large user base, which in turn has led to an active developer base and Matplotlib’s powerful tools and ubiquity within the scientific Python world.

In recent years, however, the interface and style of Matplotlib have begun to show their age. Newer tools like ggplot and ggvis in the R language, along with web visualization toolkits based on D3js and HTML5 canvas, often make Matplotlib feel clunky and old-fashioned. Still, I'm of the opinion that we cannot ignore Matplotlib's strength as a well-tested, cross-platform graphics engine. Recent Matplotlib versions make it relatively easy to set new global plotting styles (see Customizing Matplotlib: Configurations and Style Sheets), and people have been developing new packages that build on its powerful internals to drive Matplotlib via cleaner, more modern APIs—for example, Seaborn (discussed in Visualization With Seaborn), ggpy, HoloViews, Altair, and even Pandas itself can be used as wrappers around Matplotlib's API. Even with wrappers like these, it is still often useful to dive into Matplotlib's syntax to adjust the final plot output. For this reason, I believe that Matplotlib itself will remain a vital piece of the data visualization stack, even if new tools mean the community gradually moves away from using the Matplotlib API directly.

Importing Matplotlib: Just as we use the np shorthand for NumPy and the pd shorthand for Pandas, we will use some standard shorthands for Matplotlib imports.

Plotting from a script: If you are using Matplotlib from within a script, the function plt.show() is your friend. plt.show() starts an event loop, looks for all currently active figure objects, and opens one or more interactive windows that display your figure or figures.

The plt.show() command does a lot under the hood, as it must interact with your system's interactive graphical backend. The details of this operation can vary greatly from system to system and even installation to installation, but matplotlib does its best to hide all these details from you.

One thing to be aware of: the plt.show() command should be used only onceper Python session, and is most often seen at the very end of the script. Multiple show() commands can lead to unpredictable backend-dependent behavior, and should mostly be avoided.

Setting Styles: We will use the plt.style directive to choose appropriate aesthetic styles for our figures. Here we will set the classic style, which ensures that the plots we create use the classic Matplotlib style.

**4.3 Data Modelling and Visualization**

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click. JavaScript functions can be created within the editor using a rich text editor.

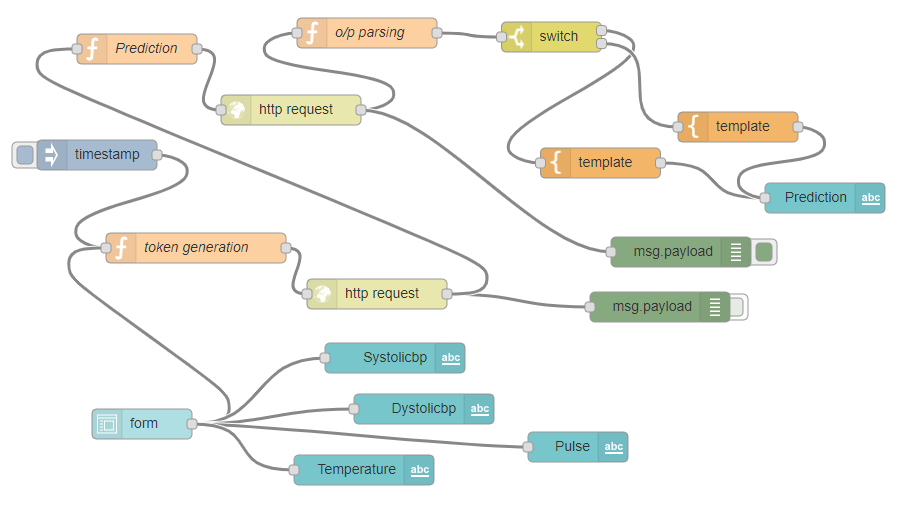
A built-in library allows you to save useful functions, templates or flows for re-use.

The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non- blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

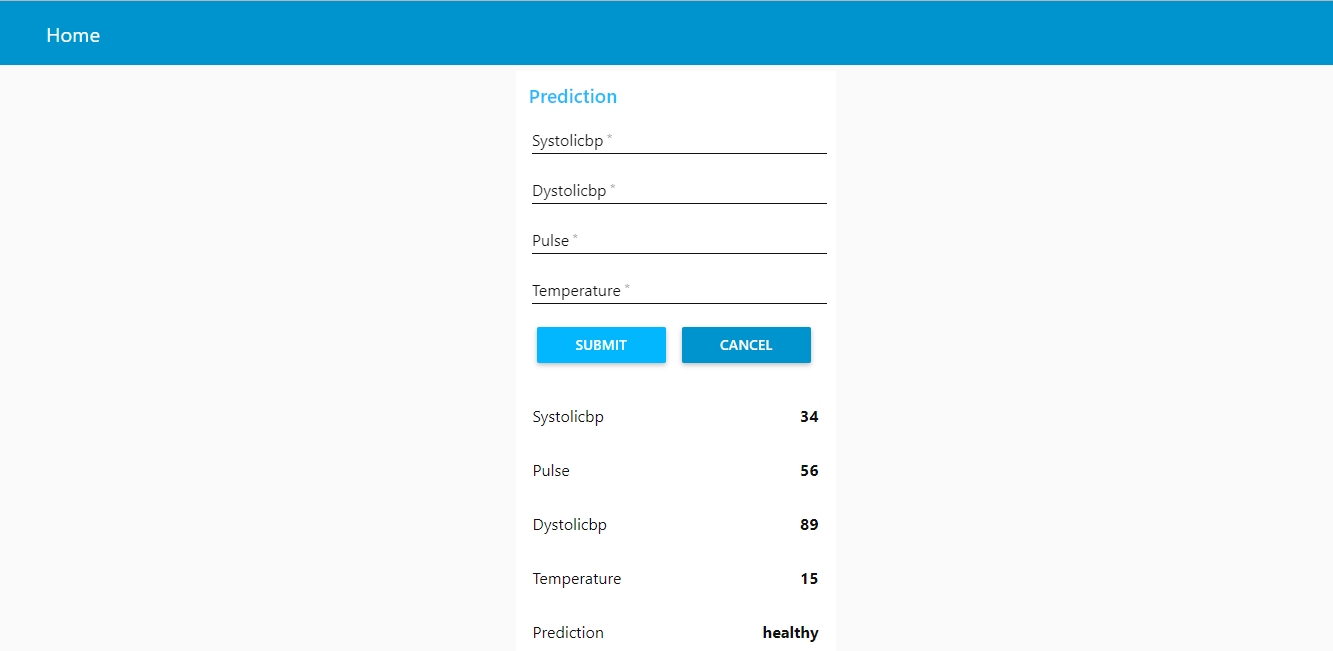
With over 225,000 modules in Nodes package repository, it is easy to extend the range of palette nodes to add new capabilities.

The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others.

An online flow library allows you to share your best flows with the world.



**Fig 4.3.1: Flow Diagram**



**Fig 4.3.2: Output**

**5. FINDINGS AND SUGGESTIONS**

The Health Monitoring System is used to test patient health conditions like checking blood pressure, heart beat rate etc. Here we take blood pressure as one measure, and divide into two categories. Based on that, we predict whether person is healthy or less healthy. I think that in future more than single patient at different places can be monitored using single system.

**6. CONCLUSION**

With the help of our work, we will be predicting whether patient is healthy or less healthy. This prediction will give suggestions based on the result obtained which level of attribute can lead to a failure. We will implement different models and find out which one is of higher accuracy.