

SMART HEALTH PREDICTION

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

by

Nikitha Michael 38110371

Riyasingh 38110463



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119.**

MARCH 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, Chennai -
600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Nikitha Michael (38110371) and Riyasingh (38110463)** who carried out the project entitled "**SMART HEALTH PREDICTION**" under our supervision from November 2021 to March 2022.

Internal Guide

Dr. S. Vigneshwari, M.E., Ph.D.,

Head of the Department

Dr. S. Vigneshwari, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **Nikitha Michael and Riya Singh** hereby declare that the Project Report entitled "**SMART HEALTH PREDICTION**" done by me under the guidance of Dr. S. Vigneshwari, M.E., Ph.D., Department of Computer Science and Engineering at Sathyabama Institute of Science and technology is submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department,** Department of Computer Science and Engineering for providing me the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs.D. Usha Nandini, M.E., (Ph.D.,)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

As we all know that there is a larger population, patients are in line at the hospital's front for care. Smart health prediction is a framework in which this problem can be resolved by using machine learning algorithm. This is an informed health prediction system. Once the standard of medical information is incomplete, the accuracy of the study will be reduced. Moreover, completely different regions have a distinctive appearance of bound regional diseases, which can lead to a weakening of the prediction of unwellness outbreaks. The patient provides symptoms in this model after that decision tree algorithm is used to determine the possible illness. This framework is implemented by using flask for interface and decision tree algorithm. The analysis accuracy is increased by using Machine Learning algorithm. This framework also displays the specialist name after entering specialty and area. This framework help to patients to identify the disease as well as locate the doctors near-by.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1 DOMAIN INTRODUCTION	1
	1.2 MACHINE LEARNING METHODS	2
	1.2.1 Supervised Learning	2
	1.2.2 Unsupervised Learning	2
2	LITERATURE SURVEY	10
3	METHODOLOGY	13
	3.1 OBJECTIVE	13
	3.2 EXISTING SYSTEM	13
	3.3 PROPOSED SYSTEM	14
	3.3.1 System Specification	14
	3.3.1.2 Hardware Requirements	14
	3.3.1.3 Software Requirements	14
	3.3.2 User Module	15
	3.3.2.1 Decision Tree Classifier	15
	3.3.3 Find Doctor Module	24

4	RESULTS AND DISCUSSION	27
	4.1 OUTPUT DISCUSSION	27
5	CONCLUSION AND FUTURE WORK	29
	5.1 CONCLUSION	29
	5.2 FUTURE SCOPE	29
	REFERENCES	30
	APPENDIX	32
	A. SOURCE CODE	32
	B. SCREENSHOTS	40
	C. PUBLICATION	42

LIST OF FIGURES

FIGURE NO.	NAME OF THE FIGURE	PAGE NO.
1.1	MACHINE LEARNING ARCHITECTURE	3
3.1	PROPOSED ARCHITECTURE	15
3.2	EXAMPLE DIAGRAM	16
3.3	EXAMPLE DATA	18
3.4	OUTLOOK ATTRIBUTE	21
3.5	FINAL DECISION TREE	22
3.6	SAMPLE PROGRAM	24
3.7	JINJA IS BEAUTIFUL	25
3.8	FIND DOCTOR TAB	26
4.1	DISPLAYING DISEASE NAME	27
4.2	SUGGESTING NEARBY SPECIALIST	28

LIST OF TABLES

TABLE NO.	NAME OF TABLE	PAGE NO.
1.1	HTTP METHODS	6
3.1	VALUE OF ATTRIBUTES	20
3.2	ATTRIBUTES	22

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
OCR	-	Optical Character Recognition
ML	-	Machine Learning
API	-	Application Programming Interface
WSGI	-	Web Server Gateway Interface
GPL	-	General Public License
GUI	-	Graphical User Interface
KNN	-	K-nearest neighbor
SVM	-	Support Vector Machine
ANN	-	Artificial Neural Network
ID3	-	Iterative Dichotomiser 3
CSV	-	Comma-Separated Values

CHAPTER 1

INTRODUCTION

1.1 DOMAIN INTRODUCTION

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

The analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks. However, those existing work mostly considered structured data. There is no proper methods to handle semi structured and unstructured. The proposed system will consider both structured and unstructured data. The analysis accuracy is increased by using Machine Learning algorithm.

Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Optical character recognition (OCR) technology converts images of text into movable type. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learning to navigate may soon be available to consumers. The analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks. However, those existing work mostly considered structured data. There is no proper methods to handle semi structured and unstructured. The proposed system will consider both

structured and unstructured data. The analysis accuracy is increased by using Machine Learning algorithm.

1.2 MACHINE LEARNING METHODS

In machine learning, tasks are generally classified into broad categories. Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data in order to allow it to find structure within its input data. The following are the detailed explanation of the methods.

1.2.1 Supervised Learning

In supervised learning, the computer is provided with example inputs that are labeled with their desired outputs. The purpose of this method is for the algorithm to be able to “learn” by comparing its actual output with the “taught” outputs to find errors, and modify the model accordingly. Supervised learning therefore uses patterns to predict label values on additional unlabeled data.

1.2.2 Unsupervised Learning

In unsupervised learning, data is unlabeled, so the learning algorithm is left to find commonalities among its input data. As unlabeled data are more abundant than labeled data, machine learning methods that facilitate unsupervised learning are particularly valuable. So, the goal of unsupervised learning may be as straightforward as discovering hidden patterns within a dataset, but it may also have a goal of feature learning, which allows the computational machine to automatically discover the representations that are needed to classify raw data.

1.3 MACHINE LEARNING ARCHITECTURE

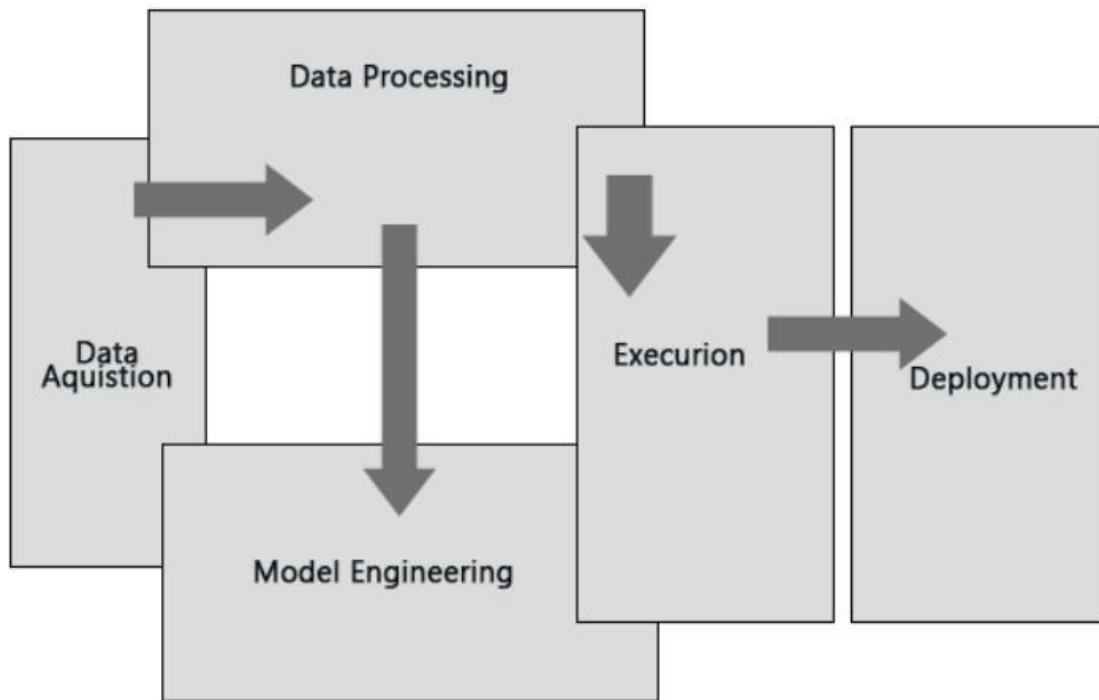


Fig.1.1: Machine learning Architecture

Architecture of Machine Learning involves a series of steps before predicting the accurate outcome for the inputs given. The following is the brief explanation about the steps involved for the prediction process.

1.3.1 Data Acquisition

As machine learning is based on available data for the system to make a decision hence the first step defined in the architecture is data acquisition. This involves data collection, preparing and segregating the case scenarios based on certain features involved with the decision making cycle and forwarding the data to the processing unit for carrying out further categorization. This stage is sometimes called the data preprocessing stage. The data model expects reliable, fast and elastic data which may be discrete or continuous in nature. The data is then passed into stream processing systems (for continuous data) and stored in batch data warehouses (for discrete data) before being passed on to data modeling or processing stages.

1.3.2 Data Processing

The received data in the data acquisition layer is then sent forward to the data processing layer where it is subjected to advanced integration and processing and involves normalization of the data, data cleaning, transformation, and encoding. The data processing is also dependent on the type of learning being used. For e.g., if supervised learning is being used the data shall be needed to be segregated into multiple steps of sample data required for training of the system and the data thus created is called training sample data or simply training data. Also, the data processing is dependent upon the kind of processing required and may involve choices ranging from action upon continuous data which will involve the use of specific function based architecture, for example, lambda architecture, Also it might involve action upon discrete data which may require memory bound processing. The data processing layer defines if the memory processing shall be done to data in transit or in rest.

1.3.3 Data Modeling

This layer of the architecture involves the selection of different algorithms that might adapt the system to address the problem for which the learning is being devised, these algorithms are being evolved or being inherited from a set of libraries. The algorithms are used to model the data accordingly, this makes the system ready for execution step.

1.3.4 Execution

This stage in machine learning is where the experimentation is done, testing is involved and tunings are performed. The general goal behind being to optimize the algorithm in order to extract the required machine outcome and maximize the system performance, The output of the step is a refined solution capable of providing the required data for the machine to make decisions.

1.3.5 Deployment

Like any other software output, ML outputs need to be operationalized or be forwarded for further exploratory processing. The output can be considered as a non-deterministic query which needs to be further deployed into the decision-making system.

1.4 FLASK

Flask is an API of Python that allows to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it have less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine. This framework uses flask for web Interface.

Flask is a lightweight web application framework of python and baseband on the WSGI toolkit and jinja2 template engine. Flask takes the versatile python programming language and offers a simple web creation framework. When loaded into python, Flask can be used to save time for web applications that build up. It leaves the essence basic and stretchable. It has no abstraction layer of the database, type validation, or any other elements. Flask allows prolongations. Extensions exist for object-related mappers, type validation, upload management, different techniques for flexible authentication.

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr. No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

Table 1.1: Http Methods

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

1.5 SOFTWARE ENVIRONMENT

1.5.1 Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

1.5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

1.5.3 Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

1.5.4 Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

CHAPTER 2

LITERATURE SURVEY

Sivaranjani et al [1] explains that healthcare system needs to be modernized. It means that the healthcare system data should be properly analyzed. To categorize it into group of diseases, symptoms, treatments. In this paper Big data is used to predict the disease and avoid preventable deaths. Big data provides security and privacy. This approach is little bit expensive.

Ankita et al [2] discussed about the thyroid disease. This paper is only for prediction of thyroid disease only by using KNN, SVM, decision Tree algorithms. This framework is used to estimated risk on patient's chance of getting thyroid disease. This framework requires less number of elements of a person to diagnose thyroid disease.

Aakash Khatavkar et al [3] paper say that decisions related to health are crucial as it may cost a person his/her life. The way the doctor makes the decision is similar to the doctor's decision-making power that can be given to the computer in which the machine can recognize and interpret the symptoms and then assign them-based possible diseases just as the doctor does. This also uses patient 's history to make their results better with the help of data mining algorithm. It also suggests homemade remedies for instant relief. But this paper restricted to viral and regular disease only.

G.pooja reddy et al [4] discusses about techniques and applications of data mining in health care industry. Their aim is predict disease by using already existing information. By using that information, it analyses and extracts new patterns. They used machine learning and database management to predict the disease.

Priyanka vijay pawar et al [5] paper does the same as our paper but they used apriori algorithm for identifying probability of diseases. They have created a prototype in which user needs to give his/her symptoms. This framework also provides some health tips that to stay healthy. This will work on real time data also.

Talha Mahboob Alam et al [6] used data mining techniques for prediction of diabetes. It is mainly predicted by using various attributes via principal component analysis method and they used k-means, KNN, ANN for prediction at early stage. This paper is concerned only with diabetes that to its accuracy is 75.7%.

K. Gomathi et al [7] paper discussed about various data mining methods that are used for predicting probable disease. They used naïve bayes's algorithm and decision tree algorithm for predicting the disease. This paper mainly focussed on predicting Diabetes, heart disease and Breast cancer. They also compared these two different classifiers on various disease and performance for accuracy in prediction.

M. Nikhil Kumar et al [8] paper applied many machine learning algorithms to predict heart diseases. They also compared classifiers and selects four among them for predictive analysis. But still it needs some accuracy for prediction and it is not suitable for predicting multiple diseases.

Bibi Amina begum et al [9] papers discussed about predicting thyroid disease at early stages. For this data mining techniques plays an prominent role for making decision, disease-prediction. They used KNN, SVM, Naïve Bayes and ID3 classifiers. This paper also finds the correlation towards hyperthyroidism and hypothyroidism.

Manish Kumar said that healthcare system is producing huge amounts of data which need to undergo for mining to discover hidden info for prediction and diagnosis. This paper implemented by using six machine learning algorithms for predicting kidney disease and also compared the performance of six classifiers.

Diagnosis of Alzheimer's disease is often difficult, especially early in the disease process at the stage of mild cognitive impairment. Yet, it is at this stage that treatment is most likely to be effective, so there would be great advantages in improving the diagnosis process. We describe and test a machine learning approach for personalized and cost-effective diagnosis of AD. It uses locally weighted learning to tailor a classifier model to each patient and computes the sequence of biomarkers most informative or cost-effective to diagnose patients. Using ADNI data, we classified AD versus controls and MCI patients who progressed to AD within a year, against those who did not. The approach performed similarly to considering all data at once, while significantly reducing the number (and cost) of the biomarkers needed to achieve a confident diagnosis for each patient. Thus, it may contribute to a personalized and effective detection of AD, and may prove useful in clinical settings in Machine learning-based method for personalized and cost-effective detection of Alzheimer's disease

Spectral data have been widely used to estimate the disease severity levels of different plants. However, such data have not been evaluated to estimate the disease

stages of the plant. This study aimed at developing a spectral disease index that is able to identify the stages of wheat leaf rust disease at various DS levels. To meet the aim of the study, the reflectance spectra of infected leaves with different symptom fractions and DS levels were measured with a spectroradiometer. Then, pure spectra of the different disease symptoms at the leaf scale were analyzed, and a new function was developed to find the wavelengths most sensitive to disease symptom fraction. The reflectance spectra with highest sensitivity were found at 675 and 775 nm. Finally, the normalized difference of DS and the ratio p_{675}/p_{775} was used as a new SDI to discriminate three different levels of the disease stage at the canopy level. The suggested SDI showed a promising performance to improve the detection disease stages in precision plant protection in Developing an Index for Detection and Identification of Disease Stages

The article emphasizes clinical and prognostic significance of non-linear measures of the heart rate variability, applied on the group of patients with coronary heart disease and age-matched healthy control group. Three different methods were applied: Hurst exponent, Detrended Fluctuation Analysis and approximate entropy. Hurst exponent of the R-R series was determined by the range rescaled analysis technique. DFA was used to quantify fractal long-range-correlation properties of heart rate variability. Approximate entropy measures the unpredictability of fluctuations in a time series. It was found that the short-term fractal scaling exponent. The patients with CHD had lower Hurst exponent in each program of exercise test separately, as well as approximate entropy than healthy control group in Non-Linear Analysis of Heart Rate Variability in Patients with Coronary Heart Disease.

CHAPTER 3

METHODOLOGY

3.1 OBJECTIVE

To design and develop a software to predict the diseases based on the user defined symptoms and also suggests the specialist in the user specified areas. It is useful for the people who do not have idea about whom to concern for the disease.

3.2 EXISTING SYSTEM

Machine will predict unwellness however unable to predict the sub kinds of the diseases reasoned by incidence of one disease. It's not able to predict all doable requirements of the folks. In current past, uncounted unwellness estimate classifications are advanced and in procedure. The standing organizations coordinate a mix of machine learning algorithms in which diseases are considered to be judiciously accurate. However, the restraint with the prevailing systems area unit specked. First, the prevailing systems area unit dearer solely wealthy folks may get to such calculation systems. And also, once it involves of us, it becomes even higher. Second, the guessing mechanisms have so far been non-specific and infinite. In such a way, a computer can envisage a positive disease but cannot anticipate subtypes of diseases and diseases caused by the presence of a single bug. For occurrence, if a group of people are foreseen with Diabetes, doubtless some of them might have complex risk for Heart viruses due to the actuality of Diabetes. The remaining schemes fail to foretell all possible surroundings of the tolerant. In existing system maximum, they are predicting for only specific diseases only.

3.3 PROPOSED SYSTEM

In proposed system, prediction of the disease is done using Decision Tree Machine Learning Algorithm. It is implemented to increase the efficiency and the accuracy of the disease prediction operations. It reduces the time required to retrieve Request and to predict the possible disease. By using Machine Learning algorithm, the accuracy of the prediction will be improved. The proposed system begins with the idea that the ancestors didn't execute. This analyses the diseases easily and it is a

simple light weight framework of python. For the better look of the application flask framework is been used because of its flexible properties and fast response for the actions taken. In this paper it contains two modules. This system is used to predict 41 diseases as the dataset's were limited.

3.3.1 System Specification

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process. It lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

3.3.1.2 Hardware Requirements

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15" LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB.

3.3.1.3 Software Requirements

- Operating system : Windows 7/10.
- Programming Language : Python 3.6.1
- Tool : Flask, pandas
- Database : MYSQL

3.3.2 User Module

In this module user needs to select his/her symptoms from drop-down list. This list contains of nearly 132 symptoms that are collected from various resources. After entering the symptoms user needs to click on symptom checker button. It checks what are the disease are associated with user selected symptoms. It analyses and predicts probable disease using decision tree algorithm.

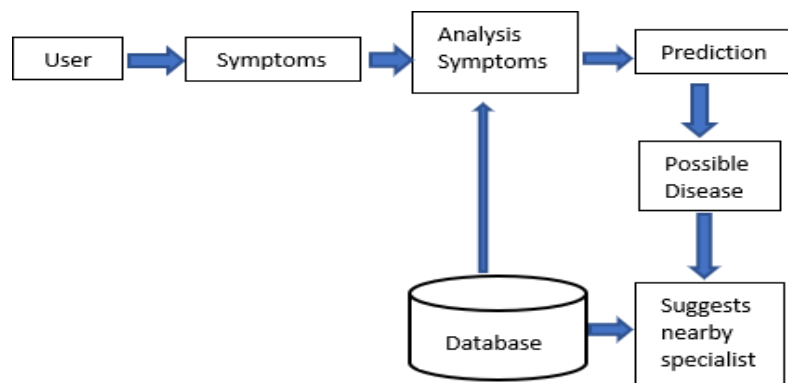


Fig.3.1: Proposed Architecture

3.3.2.1 Decision Tree Classifier

Decision tree is the most popular and best-known method of classification and prediction method. A Decision tree is a graph-like flowchart in which each internal node represents a test on an attribute, each branch is a test result and each leaf node (terminal node) carries a class identifier. Decision tree is a kind of supervised learning algorithm (with a pre-defined target variable) mostly used in classification challenges. It serves for categorical as well as continuous variables of input and output.

- A.** The pre-processing of the database contains NaN (null) values. NaN (null) values cannot be processed by programming, therefore these values must be converted to numerical values. In this approach, the mean of the column is calculated and the mean values of the NaN are replaced by the mean values.
- B.** Splitting The entire database is divided into a training and testing database. The 80 percent data is used for training, while the remaining 20 percent data is used for testing.

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

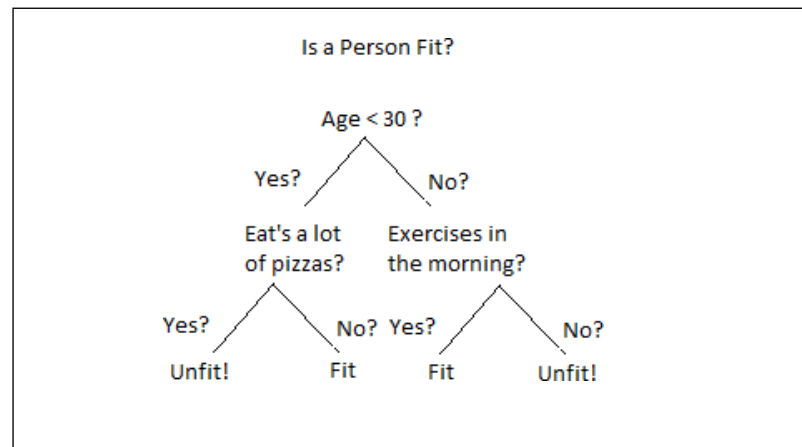


Fig.3.2: Example diagram

An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem).

There are two main types of Decision Trees:

1. **Classification trees** (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

2. **Regression trees** (Continuous data types)

Here the decision or the outcome variable is Continuous, e.g. a number like 123.

Working

Now that we know what a Decision Tree is, we'll see how it works

internally. There are many algorithms out there which construct Decision Trees, but one of the best is called as ID3 Algorithm. ID3 Stands for Iterative Dichotomiser

Before discussing the ID3 algorithm, we'll go through few definitions.

Entropy

Entropy, also called as Shannon Entropy is denoted by $H(S)$ for a finite set S , is the measure of the amount of uncertainty or randomness in data.

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

Intuitively, it tells us about the predictability of a certain event. Example, consider a coin toss whose probability of heads is 0.5 and probability of tails is 0.5. Here the entropy is the highest possible, since there's no way of determining what the outcome might be. Alternatively, consider a coin which has heads on both the sides, the entropy of such an event can be predicted perfectly since we know beforehand that it'll always be heads. In other words, this event has no randomness hence it's entropy is zero. In particular, lower values imply less uncertainty while higher values imply high uncertainty.

Information Gain

Information gain is also called as Kullback-Leibler divergence denoted by $IG(S,A)$ for a set S is the effective change in entropy after deciding on a particular attribute A . It measures the relative change in entropy with respect to the independent variables.

$$IG(S, A) = H(S) - H(S, A)$$

Alternatively,

$$IG(S, A) = H(S) - \sum_{i=0}^n P(x) * H(x)$$

where $IG(S, A)$ is the information gain by applying feature A . $H(S)$ is the Entropy of the entire set, while the second term calculates the Entropy after applying the feature A , where $P(x)$ is the probability of event x . Let's understand this with the help of an example. Consider a piece of data collected over the course of 14 days where the features are Outlook, Temperature, Humidity, Wind and the outcome variable is whether Golf was played on the day. Now, our job is to build a predictive model which takes in above 4 parameters and predicts whether Golf will be played on the day. We'll build a decision tree to do that using ID3 algorithm.

Day	Outlook	Temperature	Humidity	Wind
D1	Sunny	Hot	High	Weak
D2	Sunny	Hot	High	Strong
D3	Overcast	Hot	High	Weak
D4	Rain	Mild	High	Weak
D5	Rain	Cool	Normal	Weak
D6	Rain	Cool	Normal	Strong
D7	Overcast	Cool	Normal	Strong
D8	Sunny	Mild	High	Weak
D9	Sunny	Cool	Normal	Weak
D10	Rain	Mild	Normal	Weak
D11	Sunny	Mild	Normal	Strong
D12	Overcast	Mild	High	Strong
D13	Overcast	Hot	Normal	Weak
D14	Rain	Mild	High	Strong

Fig.3.3: Example Data

ID3 Algorithm will perform following tasks recursively

1. Create root node for the tree
2. If all examples are positive, return leaf node 'positive'

3. Else if all examples are negative, return leaf node 'negative'
4. Calculate the entropy of current state $H(S)$
5. For each attribute, calculate the entropy with respect to the attribute 'x' denoted by $H(S, x)$
6. Select the attribute which has maximum value of $IG(S, x)$
7. Remove the attribute that offers highest IG from the set of attributes
8. Repeat until we run out of all attributes, or the decision tree has all leaf nodes.

Now we'll go ahead and grow the decision tree. The initial step is to calculate $H(S)$, the Entropy of the current state. In the above example, we can see in total there are 5 No's and 9 Yes's.

$$Entropy(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

$$Entropy(S) = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right)$$

$$= 0.940$$

Remember that the Entropy is 0 if all members belong to the same class, and 1 when half of them belong to one class and other half belong to other class that is perfect randomness. Here it's 0.94 which means the distribution is fairly random. Now the next step is to choose the attribute that gives us highest possible Information Gain which we'll choose as the root node.

Let's start with 'Wind'

$$IG(S, Wind) = H(S) - \sum_{i=0}^n P(x) * H(x)$$

where 'x' are the possible values for an attribute. Here, attribute 'Wind' takes two possible values in the sample data, hence $x = \{\text{Weak, Strong}\}$

We'll have to calculate:

1. $H(S_{weak})$
2. $H(S_{strong})$
3. $P(S_{weak})$
4. $P(S_{strong})$
5. $H(S) = 0.94$ which we had already calculated in the previous example

Amongst all the 14 examples we have 8 places where the wind is weak and 6 where the wind is Strong

Table 3.1: Value of attributes

Wind = Weak	Wind = Strong	Total
8	6	14

$$\begin{aligned}
 P(S_{weak}) &= \frac{\text{Number of Weak}}{\text{Total}} \\
 &= \frac{8}{14} \\
 P(S_{strong}) &= \frac{\text{Number of Strong}}{\text{Total}} \\
 &= \frac{6}{14}
 \end{aligned}$$

Now out of the 8 Weak examples, 6 of them were 'Yes' for Play Golf and 2 of them were 'No' for 'Play Golf'. So, we have,

$$\begin{aligned}
 Entropy(S_{weak}) &= -\left(\frac{6}{8}\right) \log_2 \left(\frac{6}{8}\right) - \left(\frac{2}{8}\right) \log_2 \left(\frac{2}{8}\right) \\
 &= 0.811
 \end{aligned}$$

Similarly, out of 6 Strong examples, we have 3 examples where the outcome was 'Yes' for Play Golf and 3 where we had 'No' for Play Golf.

$$\begin{aligned}
 Entropy(S_{strong}) &= -\left(\frac{3}{6}\right) \log_2 \left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) \log_2 \left(\frac{3}{6}\right) \\
 &= 1.000
 \end{aligned}$$

Remember, here half items belong to one class while other half belong to other. Hence we have perfect randomness.

Now we have all the pieces required to calculate the Information Gain,

$$\begin{aligned}
 IG(S, Wind) &= H(S) - \sum_{i=0}^n P(x) * H(x) \\
 IG(S, Wind) &= H(S) - P(S_{weak}) * H(S_{weak}) - P(S_{strong}) * H(S_{strong}) \\
 &= 0.940 - \left(\frac{8}{14}\right)(0.811) - \left(\frac{6}{14}\right)(1.00) \\
 &= 0.048
 \end{aligned}$$

Which tells us the Information Gain by considering 'Wind' as the feature and give us information gain of **0.048**. Now we must similarly calculate the Information Gain for all the features.

$$\begin{aligned}
 IG(S, Outlook) &= 0.246 \\
 IG(S, Temperature) &= 0.029 \\
 IG(S, Humidity) &= 0.151 \\
 IG(S, Wind) &= 0.048 \text{ (Previous example)}
 \end{aligned}$$

We can clearly see that $IG(S, Outlook)$ has the highest information gain of 0.246, hence we chose Outlook attribute as the root node. At this point, the decision tree looks like.

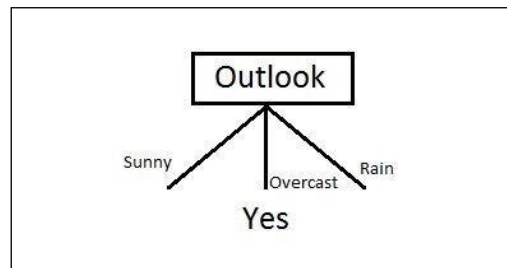


Fig.3.4: Outlook attribute

Here we observe that whenever the outlook is Overcast, Play Golf is always 'Yes', it's no coincidence by any chance, the simple tree resulted because of the highest information gain is given by the attribute Outlook. Now how do we proceed from this point? We can simply apply recursion, you might want to look at the algorithm steps described earlier.

Now that we've used Outlook, we've got three of them remaining Humidity, Temperature, and Wind. And, we had three possible values of Outlook: Sunny, Overcast, Rain. Where the Overcast node already ended up having leaf node 'Yes', so we're left with two subtrees to compute: Sunny and Rain.

Table where the value of Outlook is Sunny looks like:

Table 3.2: Attributes

Temperature	Humidity	Wind	Play Golf
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
Cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

$$H(S_{\text{sunny}}) = \left(\frac{3}{5}\right) \log_2 \left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) \log_2 \left(\frac{2}{5}\right) = 0.96$$

In the similar fashion, we compute the following values

$$IG(S_{\text{sunny}}, \text{Humidity}) = 0.96$$

$$IG(S_{\text{sunny}}, \text{Temperature}) = 0.57$$

$$IG(S_{\text{sunny}}, \text{Wind}) = 0.019$$

As we can see the highest Information Gain is given by Humidity. Proceeding in the same way with S_{rain} will give us Wind as the one with highest information gain. The final Decision Tree looks something like this.

The final Decision Tree looks something like this.

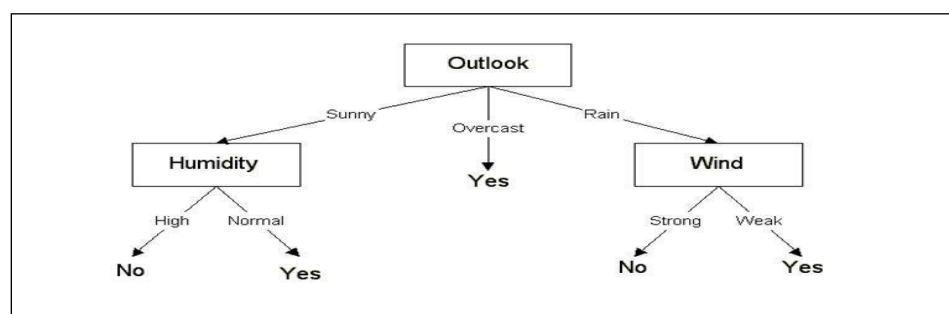


Fig.3.5: Final Decision Tree

Python Packages

1. NumPy

- a. NumPy is a Numeric Python module. Provides fast mathematical operations.
- b. Numpy provides robust data structures for efficient computing of multi-dimensional arrays & matrices.
- c. We used numpy to read data files to numpy arrays and to manipulate data.

2. Pandas

- a. Provides reading & writing data b/w different files.
- b. DataFrames can hold a variety of multidimensional array data types.

3. Scikit-Learn

- a. It's a machine learning library.
- b. We are using its for train_test split, Decision Tree Classifier

Introduction to Flask

This framework uses flask for web Interface. Flask is a lightweight web application framework of python and baseband on the WSGI toolkit and jinja2 template engine. Flask takes the versatile python programming language and offers a simple web creation framework. When loaded into python, Flask can be used to save time for web applications that build up. It leaves the essence basic and stretchable. It has no abstraction layer of the database, type validation, or any other elements. Flask allows prolongations. Extensions exist for object-related mappers, type validation, upload management, different techniques for flexible authentication. This is followed by a simple program that is implemented in flask. Flask is a simple light weight python web framework that used in this paper to design the interface of project. It is very fast and simple.

```

from flask import Flask

app = Flask(__name__)

@app.route('/')
def demo():

    return 'Demo'

```

Fig.3.6: Sample program

This framework uses CSV file where it contains all the disease names and symptoms associated with it. This helps decision tree algorithm to predict probable disease. From Flask this framework used `render_template`, `request`, `redirect` and `url_for`.

- `render_template` is a function in flask used to render the html pages. For this a templates directory to be created in a folder. This is used to return the html pages which user wants.
- `request` means that the data from the client's web page is transferred to the server as a global request object.

Form – is a dictionary entity that consists of the key and value pairs of the type parameters and its values.

- `redirect()` function is used to redirect the user to specified URL.
- `url_for()` function enables hard coding of URLs to be avoided.

A web developer has option to choose from a wide range of web frameworks while using Python as server-side programming languages. He can take advantage of the full-stack Python web frameworks to accelerate development of large and complex web applications by availing a number of robust features and tools. Likewise, he can also opt for micro and lightweight Python web frameworks to build simple web applications without putting extra time and effort. Both Django and Flask are hugely popular among Python programmers. Django is a full-stack web framework for Python, whereas Flask is a lightweight and extensible Python web framework. Django is developed based on batteries-included approach. It enables programmers to accomplish common web development tasks without using third-party tools and libraries. Django lacks some of the robust features provided by Python. But it is developed based on two robust POCO projects – Web Server Gateway Interface

(WSGI) toolkit and Jinja2 template engine. The developers can take advantage of WSGI to communicate web application and web server through a universal interface. Likewise, Jinja2 template engine makes it easier for programmers to build dynamic web applications. Hence, it becomes essential for web developers to understand the major differences between Flask vs Django.

Jinja

Jinja2 is a full-featured template engine for Python. It has full unicode support, an optional integrated sandboxed execution environment, widely used and BSD licensed.

```
{% extends "layout.html" %}

{% block body %}

<ul>

{% for user in users %}

<li><a href="{{ user.url }}">{{ user.username }}</a></li>

{% endfor %}

</ul>

{% endblock %}
```

Fig.3.7: Jinja is beautiful

Jinja2 is one of the most used template engines for Python. It is inspired by Django's templating system but extends it with an expressive language that gives template authors a more powerful set of tools. On top of that it adds sandboxed execution and optional automatic escaping for applications where security is important.

Features

- Sandboxed execution mode. Every aspect of the template execution is monitored and explicitly whitelisted or blacklisted, whatever is preferred. This makes it possible to execute untrusted templates.
- Template inheritance makes it possible to use the same or a similar layout for all templates.

- High performance with just in time compilation to Python bytecode. Jinja2 will translate your template sources on first load into Python bytecode for best runtime performance.
- Easy to debug with a debug system that integrates template compile and runtime errors into the standard Python traceback system.
- Configurable syntax. For instance, you can reconfigure Jinja2 to better fit output formats such as LaTeX or JavaScript.
- Template designer helpers. Jinja2 ships with a wide range of useful little helpers that help solving common tasks in templates such as breaking up sequences of items into multiple columns and more.

3.3.3 Find Doctor Module

There are two labels in this module, such as speciality and location. Where the user needs to enter the speciality that is needed for that specific disease and the location where his/her present. This helps the user to find the location of the specialist nearby. After entering those details, it shows the details of doctor such as name, speciality and location. This is achieved by the python xlrd library which help to read the Excel data.

We have collected the real-time data of Chennai city where it includes maximum of areas. We have nearly 29 specialists all over Chennai for different health issues. These details are stored in Excel sheet. To access this data python uses xlrd module. This compare entered data with stored data and gives details as output.

```
def get_location():
    specialitySearchTerm = request.form['speciality']
    print("The search term in doctor field is: ", specialitySearchTerm)
    df = pd.read_excel('Doctor Locations.xlsx', sheet_name = 'Sheet1')
    df = df.iloc[:,0:3]
    print(df.shape)
    df = df.dropna()
    print(df.shape)
    df = df[df['Speciality'].str.contains(specialitySearchTerm)]
    locationSearchTerm = request.form['location']
    print("The search term in location field is: ", locationSearchTerm)
    df = df[df['Area'].str.contains(locationSearchTerm)]
    print(df)
```

Fig.3.8: Find doctor tab

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Output Discussion

First the user needs to open the application and then it is important for him to know what symptoms he has and then he has to choose the symptoms he has from the dropdown list. The user can choose up to five different symptoms to get more accurate results for the inputs he mentioned.

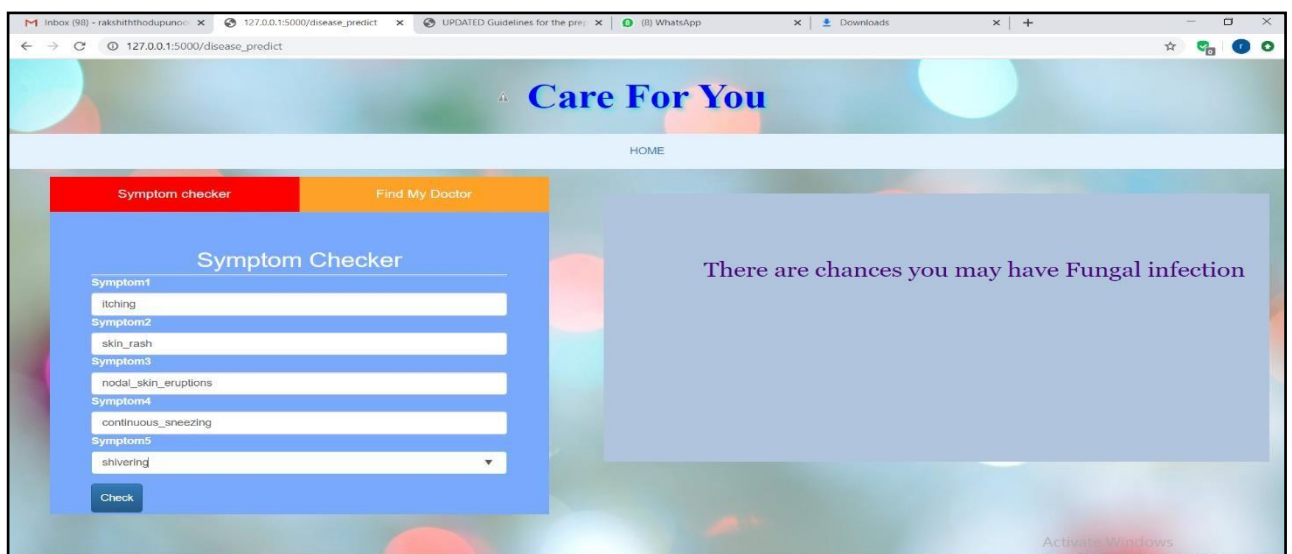


Fig.4.1: Displaying disease name

After choosing the symptoms from the drop-down list, the disease name will be displayed and presented on the right side of the home page. After getting the name of the disease we need to click on the Doctor tab. User need to enter speciality and location after submitting it displays doctor name along his/her speciality and location.

This framework contains of two tabs namely symptom checker and doctor tab. Symptom checker tab provides user to select symptoms from custom-made drop-down list. After selecting user need to submit then predicted disease is show at right side of home page. This project interface is done by help of web languages as well as flask a python framework. This comes with decision tree algorithm for predicting diseases. It additionally suggests doctor nearby location by using Chennai real time dataset. It suggests doctor name after user entering speciality and location.

This framework is used to predict disease as well as to suggest nearby specialists. This contains of two major sections where user needs to enter his/her symptoms in symptom checker tab. This tab takes users data and analyze the symptoms provided by user with respective to data. Where it shows value which disease name is entered by user in symptom checker. After analyzing the symptoms application will display the predicted disease on right side of home page.

The above screenshot says that disease name is displayed when user enter his/her symptoms in symptom checker tab. In the second snapshot it will ask user to enter speciality and location required for treatment. After entering these details in find my doctor tab it will display speciality, location and doctor's name. This process is done by decision tree algorithms it processes the data and will display the result

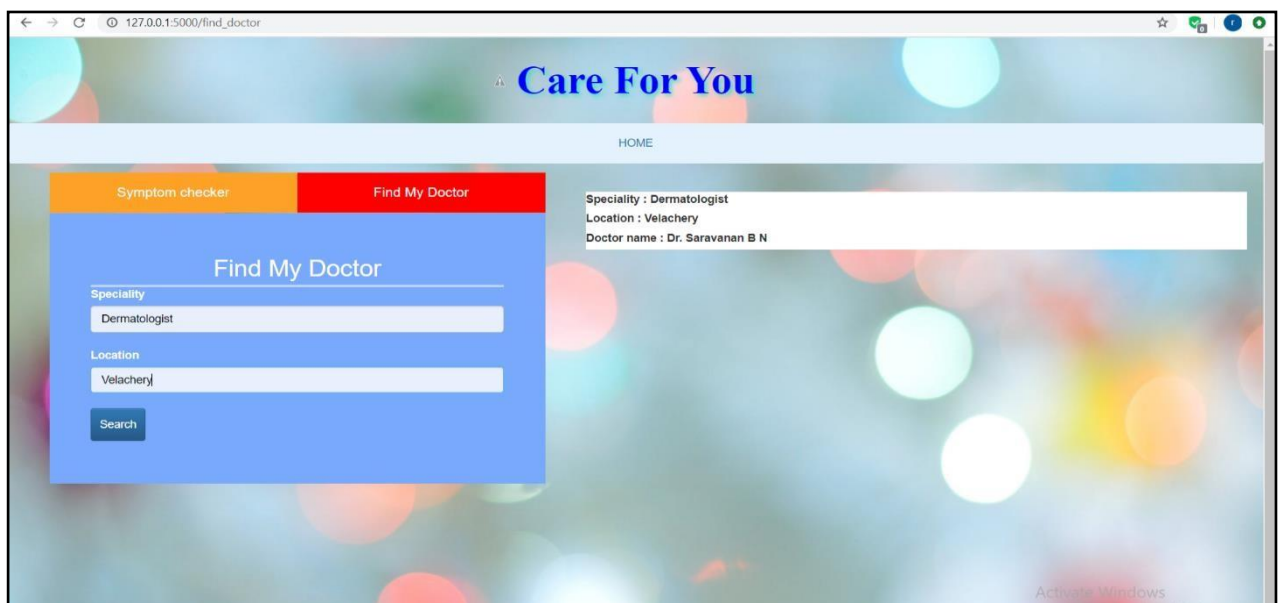


Fig.4.2: Suggesting nearby specialist

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Smart health prediction is used for predicting the probable disease and suggests doctor. This framework is simple and small where user needs to enter his symptoms then Decision Tree algorithm classifier analyses and predicts the probable disease. This framework consists of find doctor module where it helps user or patient to find nearby doctors. This project is implemented using real-time data. Basically, this model predicts probable disease and suggests nearby specialists. This paper uses flask framework to view the pages which user requests. It shows speciality, name of doctor and location in find doctor tab.

5.2 FUTURE SCOPE

The future scope of this project is to increase accuracy and to recommend natural remedies for his/her illness. Now this is just recommending the doctor name and location but in future it should also direct the user to best hospitals by using maps and machine learning algorithms for best hospitals. As it should include rating system for hospitals based on that user will choose the hospital.

REFERENCES

- [1] Aakash Khatavkar, Piyush Potpose and Pankajkumar Pandey, "Smart Health Prediction System", IJSRD- International Journal for Scientific Research & Development, Vol. 5, Issue 02, 2017, ISSN: 2321-0613.
- [2] Ankita Tyagi, Rithika Mehra and Aditya Saxena, "Interactive Thyroid Disease Prediction System Using Machine Learning Technique", 5th IEEE International Conference on Parallel, Distributed and grid computing (PDGC-2018).
- [3] Bibi Amina Begum and Dr.Parkavi A, "Prediction of thyroid Disease Using Data Mining Techniques", 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS).
- [4] Gomathi.K and Dr. D. Shanmuga Priya, "Multi Disease Prediction using Data Mining Techniques", International Journal of System and Software Engineering.
- [5] Jancy.S, C.JayaKumar "Sequence Statistical Code Based Data Compression Algorithm for Wireless Sensor Network", Wireless Personal Communication(2019), 106:971-985. Springer.
- [6] Jithina Jose, Suja Cherukullapurath Mana, B Keerthi Samhitha "An Efficient System to Predict and Analyze Stock Data using Hadoop Techniques" International Journal of Recent Techniques and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-2, July 2019.
- [7] Manish Kumar, "Prediction of Chronic Kidney Disease Using Random Forest Machine Learning Algorithm", International Journal of Computer Science and Mobile Computing IJCSMC, Vol. 5, Issue. 2, February 2016, pg, 24-33.
- [8] Nikhil Kumar.M, K.V.S. Koushik and K.Deepak, "Prediction of Heart Diseases Using Data Mining and Machine Learning Algorithms and Tools, International Journal of Scientific Research in Computer Science and Information Technology IJSRCSEIT, Volume 3, Issue 3, ISSN: 2456-3307.
- [9] Pooja reddy.G, M. Trinath basu, K. Vasanthi, k. Bala Sita Ramireddy and Ravi Kumar Tenali, "Smart E-Health prediction System Using Data Mining", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-6, April 2019.
- [10] Priyanka Vijay Pawar, Megha Sakharam Walunj and Pallavi Chitte, "Estimation based on Data Mining Approach for Health Analysis", International Journal on Recent

and Innovation Trends in Computing and Communication, ISSN: 2321-8169, Volume: 4 Issue: 4 743-746.

[11] Sivaranjani. A, S. Priyadharshini, A. porkodi, A. Vijayalakshmi and S. Suseela, "Smart Health prediction Using Hadoop", International Journal of Engineering research & technology (IJERT) ISSN: 2278-0181.

[12] Talha Mahboob Alam, Mushammad Atif Iqbal, Yasir Ali and Zunish Abbas, "A model for early prediction of diabetes", Informatics in Medicine unlocked 16 (2019) 100204.

[13] Usha Nandini.D, Ezil Sam Leni, "Efficient shadow detection by using PSO segmentation and Region-based boundary detection technique", The Journal of Supercomputing 75,no.7(2019):3522-3533.

APPENDIX

A. SOURCE CODE:

home.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-
BVYiISIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4
u" crossorigin="anonymous">

    <!-- Optional theme -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp"
crossorigin="anonymous">

    <!-- Latest compiled and minified JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-
Tc5lQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNlpG9mGCD8wGNlcPD7Tx
a" crossorigin="anonymous"></script>

    <style type="text/css">
      body, html {
        height: 100%;
      }
      .bg {
        /* The image used */
```

```
background-image: url("https://images.pexels.com/photos/255379/pexels-photo-255379.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260");
```

```
/* Full height */
```

```
height: 100%;
```

```
/* Center and scale the image nicely */
```

```
background-position: center;
```

```
background-repeat: no-repeat;
```

```
background-size: cover;
```

```
}
```

```
.head {
```

```
color: white;
```

```
font-family: sans-serif;
```

```
font-size: 30px;
```

```
border-bottom: double;
```

```
text-align: center;
```

```
}
```

```
.nav.navbar-nav {
```

```
display: flex;
```

```
justify-content: center;
```

```
flex-wrap: wrap;
```

```
}
```

```
@media (min-width: 768px) {
```

```
  .navbar-nav {
```

```
    float: none;
```

```
  }
```

```
}
```

```
.tablink {
```

```
background-color: #FEA127;
```

```
color: white;
```

```

float: left;
border: none;
outline: none;
cursor: pointer;
padding: 14px 16px;
font-size: 17px;
width: 25%;
}

```

```

.tablink:hover {
    background-color: green;
}

```

/* Style the tab content */

```

.tabcontent {
    color: white;
    display: none;
    padding: 50px;
    width: 855px;
    position: relative;
    height: 350px;
}

```

```

#footer{background-color: black;
text-decoration: underline;
text-align: center;
font-family: sans-serif;
text-decoration-color: white;
color: white;
align-self: bottom;}

```

```

#Symptom {background-color:#78A9FD; width: 600px; top:45px ;left:
50px;height:450px;}

```

```

#Doctor {background-color: #78A9FD; width: 600px;top:45px; left: 50px}

```

```

#Drug {background-color: #78A9FD; width: 600px;top:45px ;left: 50px}

```

```

</style>

</head>
<body>
  {% block body%}
  <div class="bg">
    <br>
    <div>
      <center>
      <span style="font-size:50px; text-shadow: 3px 3px 6px #40e0d0;font-weight:
bolder;font-family:'Garamound';color:blue">Disease Predictor</span></center>
    </div>
    <br>
    <nav class="navbar navbar-light" style="background-color: #e3f2fd;">
      <div class="container-fluid">
        <ul class="nav navbar-nav navbar-center">
          <li class=""><a href="/">HOME</a></li>
          <li class=""><a href="#">ABOUT US</a></li>
        </ul>
      </div>
    </nav>
    <div>
      <button class="tablink" onclick="openTab('Symptom', this, 'red')"
id="defaultOpen" style="width: 300px; position: absolute; top: 175px; left:50px;
">Symptom checker</button>
      <button class="tablink" onclick="openTab('Doctor', this, 'red')" id="doctortab"
style="width: 300px; position: absolute; top: 175px; left:350px; ">Find My
Doctor</button>
      <!-- <button class="tablink" onclick="openTab('Drug', this, 'red')" style="width:
200px; position: absolute; top: 175px; left:450px; ">Know My Drug</button> -->

```

</div>

<div id="Symptom" class="tabcontent">

<form name="SymptomForm" id="SymForm" method="POST"

action=/disease_predict>

<div class="head">

Symptom Checker

</div>

<div class="form-group">

<!-- <select multiple data-role="tagsinput" class="form-control" size=1>

{% for symp in symptoms %}

<option value= "{{symp}}">{{symp}}</option>

{% endfor %}

</select> -->

<label for="Symptom1" align="left">Symptom1</label>

<input type="text" list="symp" multiple="multiple" class="form-control"
name="Symptom1" placeholder="Enter Symptom">

<datalist id="symp">

{% for symp in symptoms %}

<option value= "{{symp}}">{{symp}}</option>

{% endfor %}

</datalist>

<label for="Symptom2" align="left">Symptom2</label>

<input type="text" list="symp" multiple="multiple" class="form-control"
name="Symptom2" placeholder="Enter Symptom">

<datalist id="symp">

{% for symp in symptoms %}

<option value= "{{symp}}">{{symp}}</option>

{% endfor %}

</datalist>

<label for="Symptom3" align="left">Symptom3</label>

```

<input type="text" list="symp" multiple="multiple" class="form-control"
name="Symptom3" placeholder="Enter Symptom">
  <datalist id="symp">
    {% for symp in symptoms %}
      <option value= "{{symp}}">{{symp}}</option>
    {% endfor %}
  </datalist>

<label for="Symptom4" align="left">Symptom4</label>
<input type="text" list="symp" multiple="multiple" class="form-control"
name="Symptom4" placeholder="Enter Symptom">
  <datalist id="symp">
    {% for symp in symptoms %}
      <option value= "{{symp}}">{{symp}}</option>
    {% endfor %}
  </datalist>

<label for="Symptom5" align="left">Symptom5</label>
<input type="text" list="symp" multiple="multiple" class="form-control"
name="Symptom5" placeholder="Enter Symptom">
  <datalist id="symp">
    {% for symp in symptoms %}
      <option value= "{{symp}}">{{symp}}</option>
    {% endfor %}
  </datalist>

</div>

<button type="submit" class="btn btn-primary" style="padding: 10px;"
name=form>Check</button>
</form>
</div>

<div id="Doctor" class="tabcontent">

```

```

<div class="head">
    Find My Doctor<br>
</div>

<form name="Doctor_search" id="DoctorForm" method="POST"
action=/find_doctor>
<input type="text" class="form-control" name="doctor"><br>
<button type="submit" class="btn btn-primary" style="padding: 10px;"
name="search_doctor">Search</button>
</form>
</div>

<!-- <div id="Drug" class="tabcontent">
<form name="MedicineForm" id="medForm" method="POST">
<div class="head">
    Know Your Drug
</div>
<div class="form-group">
    <label for="medicine" align="left">Name</label>
    <input type="text" class="form-control" id="medicine" placeholder="medicine
name" name="medicine">
    <br>
    <button class="btn btn-primary" type="submit" method='POST'>Know about
it</button>
</div>
</form>
<script type="text/javascript">
function showResult(argument) {
    // body...
    document.getElementById(argument).style.color = "white";
}
</script>
</div> -->

```

```

</div>
<div id="footer" style="position: absolute; bottom: 0px; left: 0px; width: 100%; height:
50px;">
  <p>Copyright:Rakshith Guptha-2020</p>
</div>
</div>
<script>

```

```

function openTab(tabName,elmnt,color) {
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablink");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].style.backgroundColor = "";
  }
  document.getElementById(tabName).style.display = "block";
  elmnt.style.backgroundColor = color;

}

```

```

// Get the element with id="defaultOpen" and click on it
document.getElementById("defaultOpen").click();

```

```

</script>

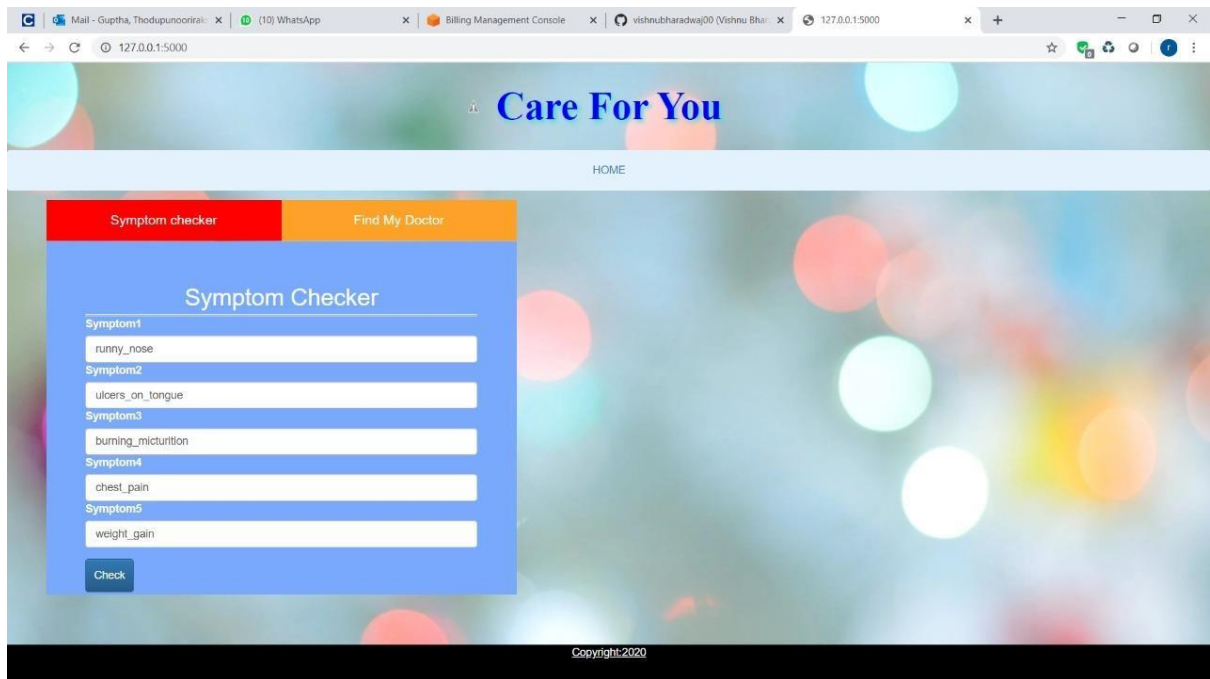
```

```

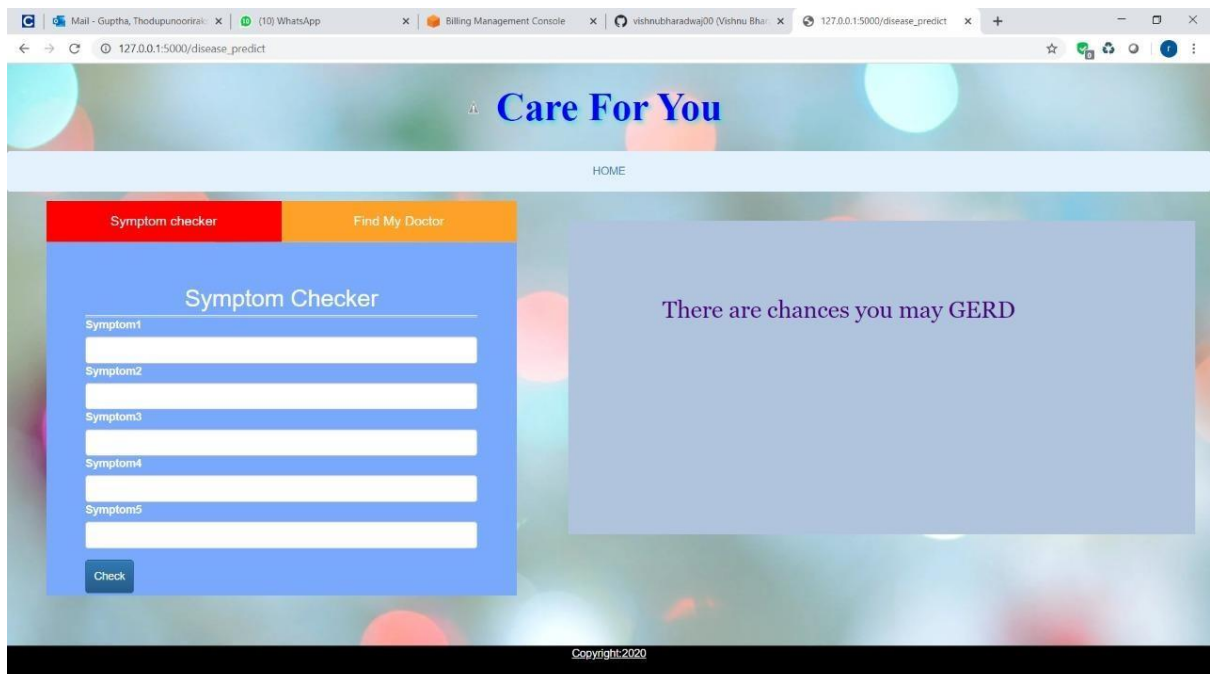
  {% endblock %}
</body>
</html>

```

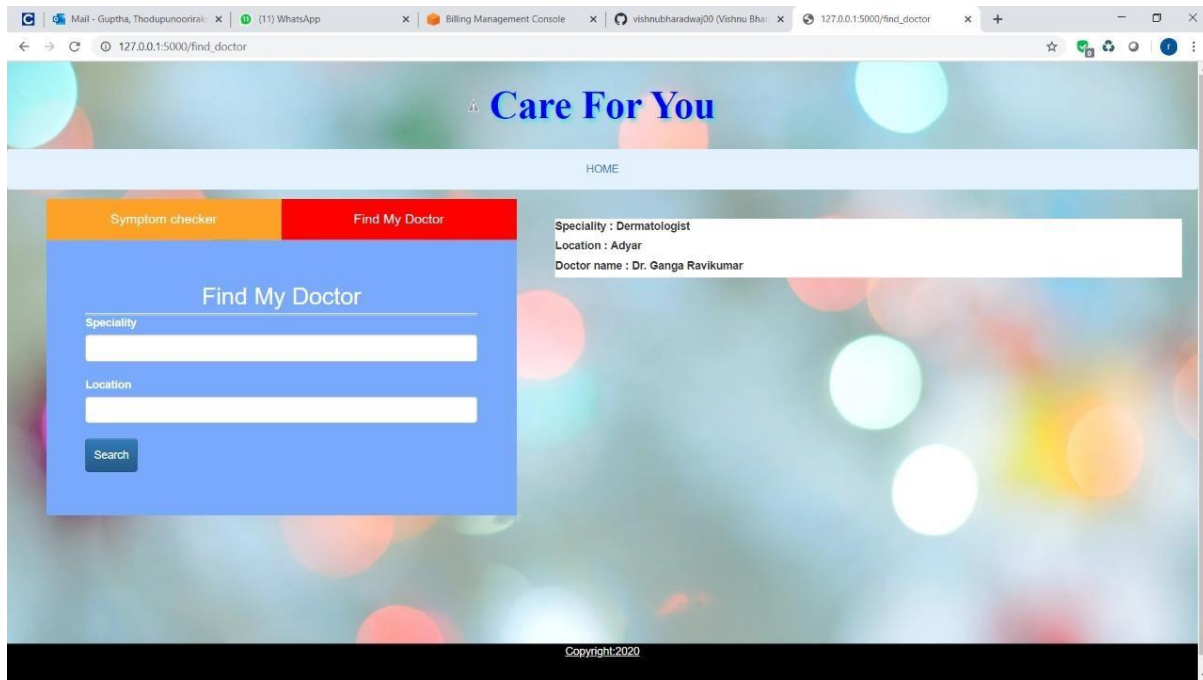
B. SCREENSHOTS:



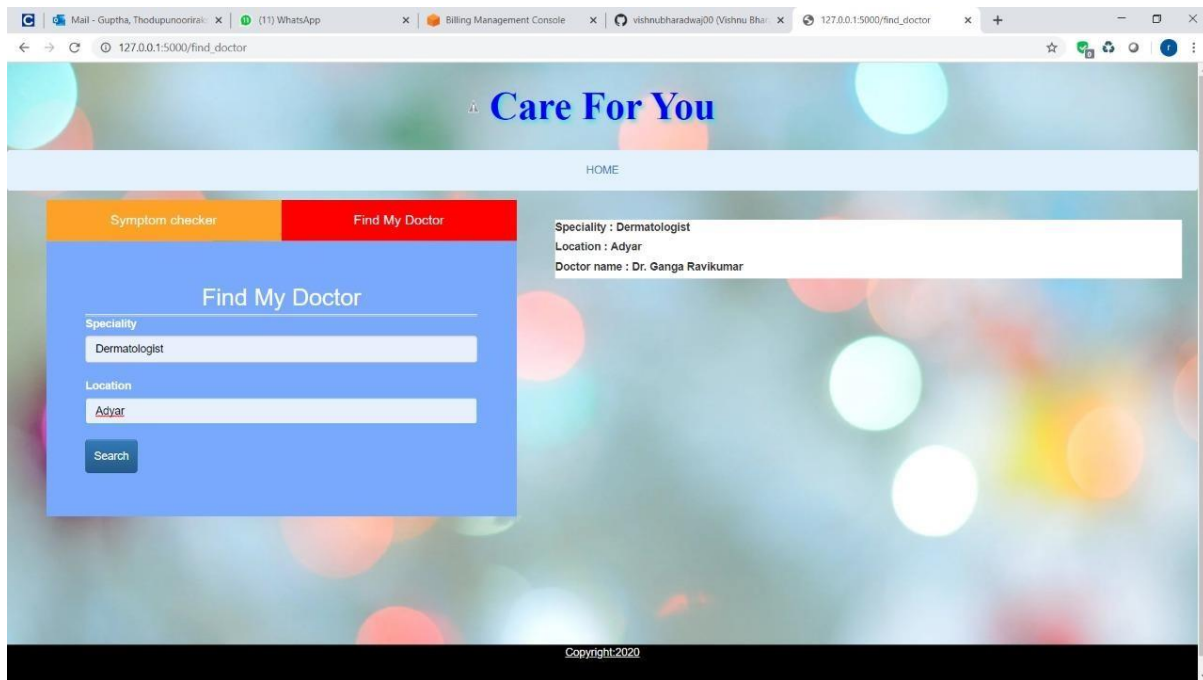
Initial symptoms application to enter the symptoms of the user.



This is the result for the entered symptoms, the result appears on the right side of the screen.



This is the second phase of the application, where the user enters in which specialty and location does the user wants to find a doctor.



This is the result of the “Find My Doctor”, where it gives the name of the specialist name based on the user choices.

C. PUBLICATION

A SYSTEM FOR INFORMED PREDICTION OF HEALTH

Rakshith Gupta Thodupunoori ^{1*}, Praneeth Sai Ummadisetty², Usha Nandini Duraisamy³

[1][2] Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Assistant Professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

Author email id: rakshiththodupunoori@gmail.com, praneethsai8864@gmail.com,
usha.cse@sathyabama.ac.in

Abstract: As we all know that there is a larger population, patients are in line at the hospital's front for care. Smart health prediction is a framework in which this problem can be resolved by using machine learning algorithm. This is an informed health prediction system. Once the standard of medical information is incomplete, the accuracy of the study will be reduced. Moreover, completely different regions have a distinctive appearance of bound regional diseases, which can lead to a weakening of the prediction of unwellness outbreaks. The patient provides symptoms in this model after that decision tree algorithm is used to determine the possible illness. This framework is implemented by using flask for interface and decision tree algorithm.

Keywords— Healthcare, Smart Health, Disease-Prediction, Prediction-Model, Symptom Checker

1. Introduction

Healthcare is a field in which decisions are generally associated with very high risks and high costs. Health-related decisions are critical because they can cost a person his / her life. Doctor analyses the patient's symptoms while diagnosing the disease. The final disease is expected based on the symptoms. In today's computerized environment, given the automated and complex requirements of healthcare system, predicting the disease and delivering effective medicines through user-friendly mobile applications should be more efficient. This research is primarily aimed at the health concerns and those who want to be their own Doctor. It is an interactive tool for people who want to learn about the effects they are going through regarding health problems.

Machine learning is computers that are trained to optimize performance using example data or past data. Machine learning is exploring computer systems that learn from data and experience. Machine learning model has two passes: training, testing. Disease prediction using the symptoms of patients and history machine learning technology has been struggling over the past decades. Machine learning technology gives an excellent medical framework to solve health concerns efficiently.

A well-known method for storing information is the health care system which uses a Database. For standard database systems, it is sometimes not possible to meet the requirements of the customer due to the existence of huge data and provide them with the exact information they need to make a decision. Nevertheless, when the consistency of medical data is insufficient, this precision of the study is reduced. The Healthcare system is far from being perfect and boring. For safer and automatic treatment these huge record sets can be analyzed and studied. To assess the patient, diagnose them and treat them with medication, this can be accomplished with machine learning systems.

2. Related Work

Sivaranjani et al [1] explains that healthcare system needs to be modernized. It means that the healthcare system data should be properly analyzed. To categorize it into group of diseases, symptoms, treatments. In this paper Big data is used to predict the disease and avoid preventable deaths. Big data provides security and privacy. This approach is little bit expensive. Ankita et al [2] discussed about the thyroid disease. This paper is only for prediction of thyroid disease only by using KNN, SVM, decision Tree algorithms. This framework is used to estimated risk on patient's chance of getting thyroid disease. This framework requires less number of elements of a person to diagnose thyroid disease. Aakash Khataavkar et al [3] paper say that decisions related to health are crucial as it may cost a person his/her life. The way the doctor makes the decision is similar to the doctor's decision-making power that can be given to the computer in which the machine can recognize and interpret the symptoms and then assign them-based possible diseases just as the doctor does. This also uses patient 's history to make their results better with the help of data mining algorithm. It also suggests homemade remedies for instant relief. But this paper restricted to viral and regular disease only. G.pooja reddy et al [4] discusses about techniques and applications of data mining in health care industry. Their aim is predict disease by using already existing information. By using that information, it analyses and extracts new patterns. They used machine learning and database management to predict the disease. Priyanka vijay pawar et al [5] paper does the same as our paper but they used apriori algorithm for identifying probability of diseases. They have created a prototype in which user needs to give his/her symptoms. This framework also provides some health tips that to stay healthy. This will work on real time data also. Talha Mahboob Alam et al [6] used data mining techniques for prediction of diabetes. It is mainly predicted by using various attributes via principal component analysis method and they used k-means, KNN, ANN for prediction at early stage. This paper is concerned only with diabetes that to its accuracy is 75.7%. K. Gomathi et al [7] paper discussed about various data mining methods that are used for predicting probable disease. They used naïve baye's algorithm and decision tree algorithm for predicting the disease. This paper mainly focussed on predicting Diabetes, heart disease and Breast cancer. They also compared these two different classifiers on various disease and performance for accuracy in prediction. M. Nikhil Kumar et al [8] paper applied many machine learning algorithms to predict heart diseases. They also compared classifiers and selects four among them for predictive analysis. But still it needs some accuracy for prediction and it is not suitable for predicting multiple disease. Bibi Amina begum et al [9] papers discussed about predicting thyroid disease at early stages. For this data mining techniques plays an prominent role for making decision, disease-prediction. They used KNN, SVM, Naïve Bayes and ID3 classifiers. This paper also finds the correlation towards hyperthyroidism and hypothyroidism. Manish Kumar said that healthcare system is producing huge amounts of data which need to undergo for mining to discover hidden info for prediction and diagnosis. This paper implemented by using six machine learning algorithms for predicting kidney disease and also compared the performance of six classifiers.

3. Existing System

Machine will predict unwellnesss however unable to predict the sub kinds of the diseases reasoned by incidence of one disease. It's not able to predict all doable requirements of the folks. In current past, uncounted unwellness estimate classifications are advanced and in procedure. The standing organizations coordinate a mix of machine learning algorithms in which diseases are considered to be judiciously accurate. However, the restraint with the

prevailing systems area unit specked. First, the prevailing systems area unit dearer solely wealthy folks may get to such calculation systems. And also, once it involves of us, it becomes even higher. Second, the guessing mechanisms have so far been non-specific and infinite. In such a way, a computer can envisage a positive disease but cannot anticipate subtypes of diseases and diseases caused by the presence of a single bug. For occurrence, if a group of people are foreseen with Diabetes, doubtless some of them might have complex risk for Heart viruses due to the actuality of Diabetes. The remaining schemes fail to foretell all possible surroundings of the tolerant. In existing system maximum, they are predicting for only specific diseases only.

4. Proposed System

In the proposed system, Decision Tree Machine Learning Algorithm predicts Disease. It is implemented to increase the efficiency of operations. It reduces the time required to retrieve Request. Machine Learning algorithm improves accuracy. The proposed system begins with the idea that the ancestors didn't execute. This analyses the diseases easily and it is simple light weight framework of python. In this paper it contains two modules. This system is used to predict 41 diseases.

4.1 User Module

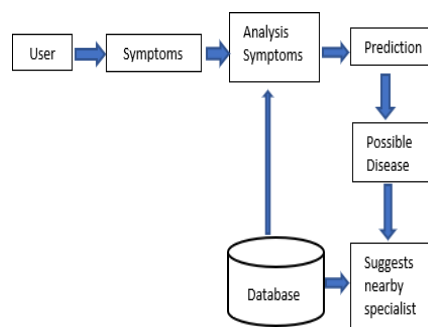


Fig.4.1. Proposed Architecture

4.1.1 Decision Tree Classifier

Decision tree is the most popular and best-known method of classification and prediction method. A Decision tree is a graph-like flowchart in which each internal node represents a test on an attribute, each branch is a test result and each leaf node (terminal node) carries a class identifier. Decision tree is a kind of supervised learning algorithm (with a pre-defined target variable) mostly used in classification challenges. It serves for categorical as well as continuous variables of input and output.

The pre-processing of the database contains NaN values. NaN values cannot be processed by programming, therefore these values must be converted to numerical values. In this approach, the mean of the column is calculated and the mean values of the NaN are replaced by the mean values.

Splitting The entire database is divided into a training and testing database. The 80 percent data is used for training, while the remaining 20 percent data is used for testing.

4.1.2 Decision Tree Algorithm

Place dataset best attribute at the tree root. Divide the training set into subtypes. Subtype should be made in such a way that each subset contains data with the same attribute value. Repeat initial two steps on each subset until the leaf nodes are located in all tree branches.

While building our decision tree classifier, we can increase its accuracy by tuning it to different parameters. But this tuning should be done carefully because, by doing so, our algorithm can overfit our training data and ultimately build a worse model.

4.2 Python Packages

4.2.1 Numpy:

NumPy is a Numeric Python module. Provides fast mathematical operations. Numpy provides robust data structures for efficient computing of multi-dimensional arrays & matrices. We used NumPy to read data files to NumPy arrays and to manipulate data.

4.2.2 Pandas:

It provides reading & writing data b/w different files. DataFrames can hold a variety of multidimensional array data types.

4.2.3 Scikit-learn:

It's a machine learning library. We are using its train_test split for the Decision Tree Classifier. By using the packages and libraries it predicts the probable disease with dosomething() function. This function takes symptoms entered by user as input. It analyses the disease associated with that symptoms and predicts the disease.

```
def dosomething(symptom):
    user_input_symptoms = symptom
    user_input_label = [0 for i in range(132)]
    for i in user_input_symptoms:
        idx = dictionary[i]
        user_input_label[idx] = 1
    user_input_label = np.array(user_input_label)
    user_input_label = user_input_label.reshape((-1,1)).transpose()
    return(dt.predict(user_input_label))
```

Fig.4.2.Symptom checker

4.3 Introduction to Flask

This framework uses flask for web Interface. Flask is a lightweight web application framework of python and baseband on the WSGI toolkit and jinja2 template engine. Flask takes the versatile python programming language and offers a simple web creation framework. When loaded into python, Flask can be used to save time for web applications that build up. It leaves the essence basic and stretchable. It has no abstraction layer of the database,

type validation, or any other elements. Flask allows prolongations. Extensions exist for object-related mappers, type validation, upload management, different techniques for flexible authentication.

This is followed by a simple program that is implemented in flask. Flask is a simple light weight python web framework that used in this paper to design the interface of project. It is very fast and simple.

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def demo():
    return 'Demo'
```

Fig.4.3. Sample program in Flask

This framework uses CSV file where it contains all the disease names and symptoms associated with it. This helps decision tree algorithm to predict probable disease. From Flask this framework used `render_template`, `request`, `redirect` and `url_for`. `render_template` is a function in flask used to render the html pages. For this a templates directory to be created in a folder. This is used to return the html pages which user wants. `request` means that the data from the client's web page is transferred to the server as a global request object. `Form` is a dictionary entity that consists of the key and value pairs of the type parameters and its values. `redirect()` function is used to redirect the user to specified URL. `url_for()` function enables hard coding of URLs to be avoided.

4.3.1 Find Doctor Module

There are two labels in this module, such as specialty and location. Where the user needs to enter the specialty that is needed for that specific disease and the location where his/her present. This helps the user to find the location of the specialist nearby. After entering those details, it shows the details of doctor such as name, specialty and location. This is achieved by the python `xlrd` library which help to read the Excel data.

We have collected the real-time data of Chennai city where it includes maximum of areas. We have nearly 29 specialists all over Chennai for different health issues. These details are stored in Excel sheet. To access this data python uses `xlrd` module. This compares entered data with stored data and gives details as output.

```

def get_location():
    specialitySearchTerm = request.form['speciality']
    print("The search term in doctor field is: ", specialitySearchTerm)
    df = pd.read_excel("Doctor Locations.xlsx", sheet_name = 'Sheet1')
    df = df.iloc[:,0:3]
    print(df.shape)
    df = df.dropna()
    print(df.shape)
    df = df[df['Speciality'].str.contains(specialitySearchTerm)]
    locationSearchTerm = request.form['location']
    print("The search term in location field is: ", locationSearchTerm)
    df = df[df['Area'].str.contains(locationSearchTerm)]
    print(df)

```

Fig.4.4. Find doctor Tab

5. Results and Discussions

After choosing the symptoms from the drop-down list, the disease name will be displayed and presented on the right side of the home page. After getting the name of the disease we need to click on the Doctor tab.

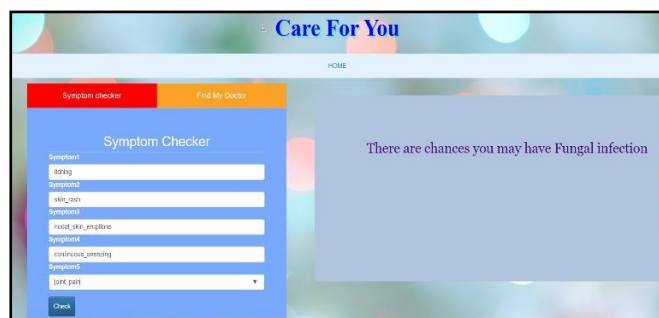


Fig.5.1. Home page screenshot

In find doctor tab, the user needs to enter the speciality and place where he wants to be treated. Upon pressing the search button, you will see the specialty, position and doctor's name.

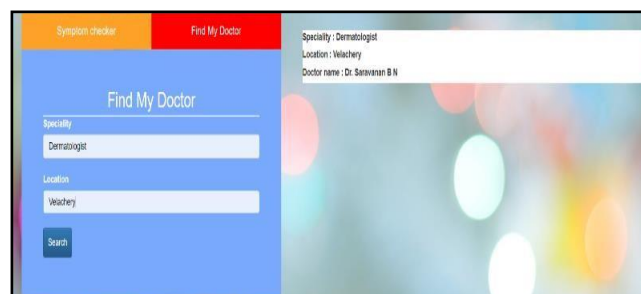


Fig 5.2. Find Doctor tab screenshot

6. Conclusion and Future Enhancement

In the above paper we studied a machine learning model that can be used to predict probable disease. This framework is simple and small where user needs to enter his symptoms. Then Decision Tree algorithm classifier analyses and predicts the probable disease. This framework consists of find doctor module where it helps user or patient to find nearby doctors. This paper is implemented using real-time data. Basically, this model predicts probable disease and suggests nearby specialists. This paper uses flask framework to view the pages which user requests. In future this model will show addresses as well as natural remedies to cure that particular disease. It should also include hospital ratings to help users in a better way.

REFERENCES

- [1] Aakash Khatavkar, Piyush Potpose and Pankajkumar Pandey, "Smart Health Prediction System", IJSRD-International Journal for Scientific Research & Development, Vol. 5, Issue 02, 2017, ISSN: 2321-0613.
- [2] Ankita Tyagi, Rithika Mehra and Aditya Saxena, "Interactive Thyroid Disease Prediction System Using Machine Learning Technique", 5th IEEE International Conference on Parallel, Distributed and grid computing (PDGC-2018).
- [3] Bibi Amina Begum and Dr.Parkavi A, "Prediction of thyroid Disease Using Data Mining Techniques", 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS).
- [4] Gomathi.K and Dr. D. Shanmuga Priya, "Multi Disease Prediction using Data Mining Techniques", International Journal of System and Software Engineering.
- [5] Jancy.S, C.JayaKumar "Sequence Statistical Code Based Data Compression Algorithm for Wireless Sensor Network", Wireless Personal Communication(2019), 106:971-985. Springer.
- [6] Jithina Jose, Suja Cherukullapurath Mana, B Keerthi Samhitha "An Efficient System to Predict and Analyze Stock Data using Hadoop Techniques" International Journal of Recent Techniques and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-2, July 2019.
- [7] Manish Kumar, "Prediction of Chronic Kidney Disease Using Random Forest Machine Learning Algorithm", International Journal of Computer Science and Mobile Computing IJCSMC, Vol. 5, Issue. 2, February 2016, pg, 24-33.
- [8] Nikhil Kumar.M, K.V.S. Koushik and K.Deepak, "Prediction of Heart Diseases Using Data Mining and Machine Learning Algorithms and Tools, International Journal of Scientific Research in Computer Science and Information Technology IJSRCSEIT, Volume 3, Issue 3, ISSN: 2456-3307.
- [9] Pooja reddy.G, M. Trinath basu, K. Vasanthi, k. Bala Sita Ramireddy and Ravi Kumar Tenali, "Smart E-Health prediction System Using Data Mining", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-6, April 2019.
- [10] Priyanka Vijay Pawar, Megha Sakharam Walunj and Pallavi Chitte, "Estimation based on Data Mining Approach for Health Analysis", International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169, Volume: 4 Issue: 4 743-746.
- [11] Sivaranjani. A, S. Priyadharshini, A. porkodi, A. Vijayalakshmi and S. Suseela, "Smart Health prediction Using Hadoop", International Journal of Engineering research & technology (IJERT) ISSN: 2278-0181.
- [12] Talha Mahboob Alam, Mushammad Atif Iqbal, Yasir Ali and Zunish Abbas, "A model for early prediction of diabetes", Informatics in Medicine unlocked 16 (2019) 100204.
- [13] Usha Nandini.D, Ezil Sam Leni, "Efficient shadow detection by using PSO segmentation and Region-based boundary detection technique", The Journal of Supercomputing 75,no.7(2019):3522-3533.

PLAGIARISM REPORT

Turnitin Originality Report

Processed on: 07-Mar-2020 12:03 AM CST ID: 1271042412
Word Count: 2675 Submitted: 1

02 By 02 02

Similarity Index

15%

Similarity by Source

Internet Sources: 8%
Publications: 4%
Student Papers: 8%

5% match (Internet from 09-Feb-2020)

<https://dataaspirant.com/2017/02/01/decision-tree-algorithm-python-with-scikit-learn/>

2% match (student papers from 11-Jun-2018)

[Submitted to Savitribai Phule Pune University on 2018-06-11](#)

2% match (student papers from 09-May-2019)

[Submitted to University of Greenwich on 2019-05-09](#)

1% match (Internet from 05-May-2019)

http://www.shanlaxjournals.in/wp-content/uploads/sijash_feb2018_sivakasinadarspioneercollege.pdf

1% match (publications)

[Ankita Tyagi, Ritika Mehra, Aditya Saxena. "Interactive Thyroid Disease Prediction System Using Machine Learning Technique", 2018 Fifth International Conference on Parallel, Distributed and Grid Computing \(PDGC\), 2018](#)

1% match (Internet from 02-Feb-2020)

<https://www.ijitee.org/wp-content/uploads/papers/v8i12/L24761081219.pdf>

1% match (Internet from 08-Sep-2018)

<https://ijcsmc.com/docs/papers/February2016/V5I2201606.pdf>

1% match (Internet from 23-Jul-2019)

<https://www.ijert.org/smart-health-prediction-using-hadoop>

< 1% match (publications)

[Manuel Amunategui, Mehdi Roopaei. "Chapter 1 Introduction to Serverless Technologies", Springer Science and Business Media LLC, 2018](#)

< 1% match (Internet from 14-May-2016)

<http://www.maxwellsci.com/print/rjaset/v10-293-297.pdf>

< 1% match (student papers from 24-May-2019)

[Submitted to Nottingham Trent University on 2019-05-24](#)

< 1% match (Internet from 31-Oct-2019)

<https://docplayer.net/75654209-Role-of-engineers-in-nation-building.html>

< 1% match (student papers from 02-Jul-2018)

[Submitted to University of Southern California on 2018-07-02](#)

< 1% match (student papers from 14-Jan-2020)

[Submitted to Cork Institute of Technology on 2020-01-14](#)

< 1% match (publications)

[R.Aroul Canessane, R. Dhanalakshmi, B. Muthukumar, Cheedella Venkata Sailaja, Bobba Jyothi, S. Dhamodaran. "Diabetes To Heart Disease: A Survey", 2019 Fifth International Conference on Science Technology Engineering and Mathematics \(ICONSTEM\), 2019](#)

< 1% match (student papers from 16-Apr-2012)

[Submitted to University of East London on 2012-04-16](#)