# Personal Health Assistant

Gulshankumar Bakle[#1], Gayatri Asodekar[#2], Vijay Chaudhari[#3] , Prof. Chandrashekhar Raut [#4]

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

*Department of Computer Engineering, Mumbai University*

[1]gulshanbakle@gmail.com

[2]asodekarg@gmail.com

[3]vijay.c4dec1997@gmail.com

[4]cmr.cm.dmce@gmail.com

*Abstract*— **While initiating treatment for a particular disease, the main problem that arises, is mapping of symptoms with many diseases. Many symptoms may show presence of multiple diseases in the body of a patient. There has been a significant rise in data generated from healthcare sector, which could be efficiently exploited to give meaningful insights using data analysis and machine learning techniques. This paper aims to propose a system which would use disease-symptoms data and predict chances of various diseases a patient could be suffering with on taking inputs as symptoms. The system will use data analysis tools and machine learning algorithm- Random Forest in order to classify diseases. The system further enables patient to locate nearby doctors around him on maps and use a price comparator functionality that would fetch prices of generic medicines under given prescription.**

***Keywords*— *data analysis, machine learning, random forest, classification, price comparator***

## I. INTRODUCTION

Data has been scaling at an exponential rate in recent years. The scenario in past decade was as such; producer would produce huge amounts of data and on other leg, consumer would consume it. But the recent years has witnessed a swift change in scenario; both producer and consumer are producing and consuming data, which has led to immense rise in data being generated, especially in healthcare and financial sectors. Generally in healthcare sector, diagnosis of a disease is done on the basis of symptoms that are visible in patient. With an increase in usage of internet and presence of information in the forms of blogs, forums, questions and answer portals, people have started spending more time over search engines to conduct self diagnosis for the symptoms visible to them. Sometimes search based self diagnosis may outcome with limited and incomplete information pertaining to prediction of diseases. Often self-diagnosis can lead to low quality results and unsubstantiated information.

Over the years, the symptoms with respect to different diseases have also seen tremendous variations. Many symptoms can be related to occurrence of more than one disease. Such instances, while diagnosis makes the treatment process to be extremely tedious and ambiguous. In order to facilitate with proper self diagnosis and reasonable accuracy, concept of symptom checker has been proposed. A symptom checker comprises of three parts as, selection of affected body part, selection of symptoms being visible and prediction of diseases. Out of these three parts, prediction of diseases is an important task, since the predicted diseases should hold efficient accuracies. The other crucial task in building a symptom checker would be to provide a smooth user experience in sharing their symptoms in symptom checker.

The system will make use of database consisting of all symptoms pertaining to various diseases. Since every industrial database is present in a raw format, it needs to be converted into usable and clean format using data pre-processing techniques. Now this clean and processed data would be used for training our machine learning model using random forest algorithm. The result of our system will give out probability percentages of various diseases the patient might be suffering with. Along with this, we have added two more extra functionalities in our system that will able it to be used as a personal health assistant. First, using Geo-Positioning System (GPS), the patient would be able to locate the nearby physicians, doctors and hospitals present around him or her, for treatment of those predicted diseases. Second being a generic medicine comparator that would fetch out the list of various low priced medicines of the same chemical components as that of branded medicines; prescribed by doctors.

The section II of this article will explore about the related work being done in this field. The section III will throw light on random forest algorithm which would be used in classifying the chances of various diseases based on symptoms. The section IV will focus on proposed system for personal health assistant. Section V will refer to the experiment and result about each module which is used in this system. Section VI will conclude the paper.

## II. LITERATURE REVIEW

There are many proposed systems that predict diseases based on the symptoms. Md. Tahmid Rahman Laskar, Md. Tahmid Hossain, Abu Raihan Mostofa Kamal, Nafiul Rashid proposed a system in a paper[1] that works on Relevant Attribute (RA) Data Structure which takes five relevant parameters S = Symptom name T = Time I = Intensity O = Organ name D = Duration from the user as input. The user will give symptoms in the form of text input. The input is then scanned and tagging of each word is done according to the RA data structure. Synonym Parent Tree, Symptom Reference Tag and Decision Tree Relevant Attribute Array techniques are used for tagging. Then a Data matrix will be formed using RA, from which the symptoms will be retrieved and mapped with the symptoms that are already in the databases. After that asymmetric binary similarity factor is calculated.

Prediction System for Diseases and Suggestion of Appropriate Medicines in this paper[2] author Disha Mahajan, Mrudula Phalak, Shubhangi Pankore, Saniya Pathan, Deepa Abin proposed a system that not only predicts the disease but also suggests medicines for it. This system uses data mining for predicting diseases. They used the dataset of MedlinePlus. This system takes symptoms as input from the user then checks the symptoms in the database after that it maps the symptoms with the disease and checks the condition for supplement medicine and display the predicted disease with suggested medicines.

In the paper [3] Disease Prediction and Doctor Recommendation System published by Dhanashri Gujar, Rashmi Biyani, TejaswiniBramhane, Snehal Bhosale, Tejaswita P. Vaidya proposed a system that predicts the disease and also recommends a doctor near the user location. The prediction of disease is done by data mining. The system uses Naive Bayes that is implemented using WEKA libraries which contain a collection of tools for visualization and also contain algorithms for data analysis and predictive modeling. Doctors are recommended based on the location as well as the reviews of the doctors that are done by fetching the various doctors based on reviews. Core NLP is used for processing the data that are fetched.

Miss Swati Y. Dugane, Prof. Karuna G. Bagde proposed a system [4] that predicts the disease as well as gives detail information of disease in question like form. The disease prediction is done using deep learning method. Deep learning is part of machine learning. The user can search the symptoms or can ask the questions to the system will respond in the form of the associated data set.

Harini D K, Natesh M in a paper[5] proposed a system that predicts the probability of disease-based on symptoms using machine learning algorithm. The system combines the structure and unstructured data and then to reconstruct missing data system uses the latent factor model. After that to it uses statistical knowledge to determine the major chronic disease. Then, it uses data that are been consulted with hospital expert to extract useful features so it can be used as structured data. Using CNN algorithm Features of unstructured data can be selected. The system uses three algorithms: KNN algorithm, Naïve Bayesian and Decision tree for prediction of disease.

## III. RANDOM FOREST ALGORITHM

Random Forest algorithm is one of supervised algorithm in machine learning, which can be effectively used for binary as well as multi label classification. Random forest or random decision forests are an ensemble learning method, which is used for classification as well as regression problems. It operates by preparing multiple numbers of decision trees on the same dataset, involving different sequence and range of parameters or features. Once multiple decision trees are constructed, all the decisions of individual trees are merged together to obtain more accurate and stable result. Often decision trees are bound to suffer from over-fitting to their training dataset, random forest tends to solve this issue since it operates by calculating the mean of obtained classes of decision tree. Random forest algorithm finds its application in many fields, since it is easy to use and can be effectively used for both classification and regression purposes.

Random forests can be termed as upgraded variant of decision trees. The major difference between them being, decision trees are built over entire dataset, while random forests are used to create multi-decision trees on subset of dataset.

Any decision tree works according to general principle, for predicting any sample or class label, it creates a set of general rules on the features present in dataset. Among all the features, one of the feature is selected as root node, and other features as internal nodes. We continue to split out our internal nodes until we reach at particular classified or predicted class. Every decision tree consists of following components:

1) *Root node*: This is the top most node present in decision tree. It splits the entire dataset into two halves. Root node is the node which has lowest value of impurity.
2) *Internal node*: Root node splits into internal nodes. All the results made out on branches are internal nodes. These internal nodes further branch out into leaf nodes.
3) *Branch*: Every branch indicates an outcome or possible action to be taken.
4) *Leaf nodes*: These are the end nodes of our decision tree.

Every decision tree works in a flow chart like structure, where each internal node denotes a test on attribute, branch indicating outcome of a test and each leaf node representing class labels. Both decision tree and random forest construct trees

using certain metrics. These metrics help in measuring best way to make splits in tree. The metrics are Gini impurity, information gain and variance reduction.

Random forest algorithm works on the principle of bootstrap aggregation. The bootstrap is a powerful statistical method which is used for estimation of a quantity from a data sample. Given a training set as A = $a_1$, $a_2$, $a_3$ ..., $a_n$ with responses, let's say as $Y = y_1, ..., y_n$, bagging repeatedly (*B* times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b$ = 1,2,3,..., *B*:

*1)* Sample, with replacement, *n* training examples from *A*, *Y*; call these $A_b$, $Y_b$.

*2)* Train a classification or regression tree $f_b$ on $A_b$, $Y_b$.

After training our model, it can be tested for unseen samples *a'*, final decision can be made by averaging the predictions from all regression trees drawn on sample *a'*.

$$f' = \frac{1}{B}\sum_{b=1}^{B} f(x')$$

Or by taking the majority vote in the case of classification trees.

This bootstrapping procedure leads to better model performance since it reduces the variance of the model, without increasing the bias. Since the training and prediction of single tree is much prone to noise, the average of large number of trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the exactly the same tree many times, if the training algorithm used in problem is deterministic); bootstrap sampling is a way which helps in de-correlating the trees using different training sets for separate trees. The number of samples per trees, *B*, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set of our problem.

The procedure described above is original bagging algorithm for trees. Random forests differ in only one way from this methodology: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a randomised subset of the features from the training dataset. This process of section randomly is called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the *B* trees, causing them to become correlated. Thus bootstrapping in random forest helps in making the trees de-correlated and improving the accuracy by taking mean of all tree results.

## IV. PROPOSED SYSTEM

At beginning, our system will mimic like a real doctor, by asking which body part is being affected. The patient will need to select the body part from the displayed list. The second step as a doctor asks about is symptoms, similarly our personal health assistant will display a set of symptoms pertaining to those selected body parts. The patient would be able to select multiple symptoms that he or she is suffering with. On the basis of symptoms given by patient, the system will make predictions of the probable diseases in the form percentages. The predictions are made by our random forest classifier object which is trained beforehand on dataset of symptoms and disease [6]. The patient would be able to see a list of predicted diseases given by our random forest classifier.

Since the dataset present online is raw data, it must be pre-processed using various data pre-processing technique. Our dataset consisted of three columns- symptoms, diseases and weight. Here weight parameter signifies the number of instances of such diseases. The dataset is processed such that all the symptoms are turned into columns and last column would be of disease column. The symptoms pertaining to particular diseases are marked as 1, while rest 0. This processed data is now trained using random forest classification algorithm. The reason behind choosing random forest classifier, as machine learning algorithm, is its ability construct multiple decision trees on features and glue them together to get a more accurate and stable prediction. The forest that it builds is an ensemble of many decision trees which are trained using bagging method.

The second functionality added in our personal health assistant being, locating the nearby doctors with the help of current Geo Positioning System (GPS) location of the patient using our web application. The patient would be able to see a list of doctors, physicians and hospitals around him or her on map.

The third functionality that our personal health assistant would provide is a price comparator for medicines. As the prices of branded medicines are high, not everyone is able to afford them. A substitute for them is generic medicines, which contains the same chemical component as that of other medicines but at lower prices. The patient would need to enter the prescribed medicine name in the system, which in return will fetch out a list of all generic medicines with same chemical components as that of prescribed medicines. This will allow the patient to make a proper choice for purchasing the medicines. A figurative representation of entire system is shown in figure 1.

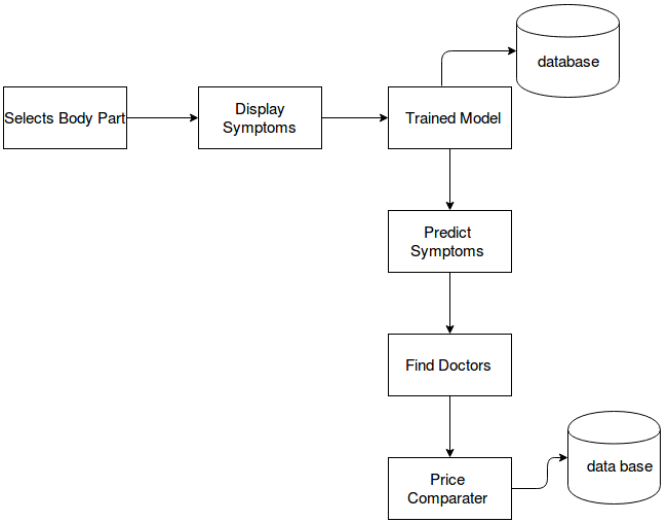


*Fig. 1 Proposed system of personal health assistant*

These all functionalities are represented in a form of a web application. For front end purposes, we have used HTML, CSS and JavaScript in order to bring out responsiveness in real time. The back-end part of this web app is made with django framework. Django provides an easy way to integrate database with website, and also indulges in easy implementation of machine learning in a form of web app. For machine learning part we have used python language, since it contains wide range of inbuilt data analysis libraries like numpy, pandas, matpotlib, seaborn, etc.

## V. EXPERIMENTS AND RESULTS

The personal health assistant consists of three important functionalities- symptom checker, doctor locater and medicine price comparator. Working of each module is explained below in detail:

### *A. Symptom Checker:*

The symptom checker uses symptoms diseases database, which is pre-processed and trained using machine learning algorithm- random forest.The information within this database is transformed into dataset. A dataset is the collection of related sets of information that is composed of separate elements but can be manipulated as single unit. As our database consisted of three columns- symptoms, disease and weight of instances of symptoms noticed for that disease. We used python libraries like numpy and pandas for processing these three columns into encoded form. The raw dataset is converted into clean dataset, thus removing any missing, repetitive and faulty data. This clean dataset is now turned into a form that can be used for training our model. All the unique symptoms are turned into columns and a separate label column for unique disease names. The cells of symptoms pertaining to each disease are filled with 1 and rest cells as 0. There are in total 148 unique diseases in our dataset and 404 unique symptoms.

Following libraries were used while building our symptom checker:

*1)* Scikit learn library: Scikit-learn is a free machine learning library especially built for python language. It features many algorithms like decision tree, random forests, and Multinomial Naive Bayes, and it also supports python numerical and scientific libraries like NumPy and SciPy.

*2)* Numpy: NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Such operations are executed more efficiently and in less lines of code as compared to other python operations.

*3)* Pandas: Pandas is popular package used widely in machine learning, commonly much used for data manipulation and analysis. Much of data pre-processing task is done using Numpy and Pandas.

We have used scikit learn library, from which we will import our random forest classifier.

Before applying random forest classifier on our dataset, we split the dataset into two parts- training set and testing set. The split is done in the ratio of 80:20. The total length of our dataset is 148, that is, 148 unique diseases which are split in the mentioned ratio. Training set consists of 118 diseases while testing set with 30 diseases. The training set was made up of two components, x train and y train; x train consisting of all the symptoms and y train list of diseases of those symptoms. When random forest classifier was applied on our entire dataset, we obtained an accuracy score of 90.54 %. When this classifier was run on our training dataset, it gave us an accuracy of 88.98%, while on testing dataset, accuracy rose to 96.66%. This result has been formulated in table 1. There were 13 such instances, when our classifier gave false predicted diseases.

Table 1

| Dataset | Accuracy achieved in percentage |
|---|---|
| Complete dataset with 148 diseases | 90.54 |
| Training dataset with 118 diseases | 88.98 |
| Testing dataset with 30diseases | 96.66 |

There are in total 404 unique symptoms which act like features in our classifier. Some of the features are important in constructing multiple decision trees, based on how efficiently the nodes reach out to predict class labels. Feature importance is generally calculated as the decrease in impurity of node, which is weighted by the probability of reaching till that node in tree. The probability of node can be calculated by dividing the number of samples that reach the node, by the total number of samples. Higher the value of probability, more important the feature would be.

Table 2 shows list of top features, which showed higher node probability, thus come under important features:

Table 2

| Feature index | Feature name |
|---|---|
| 255 | Pain abdominal |
| 70 | Cough |
| 122 | Fever |
| 80 | Diarrhoea |
| 45 | Breathiness sound decreased |
| 87 | Dizziness |

We have used matplot library and seaborn library in order to plot most important features among 404 symptoms. As we can observe from table 2, pain abdominal forms most important feature in the entire set of features which is followed by cough, fever, diarrhoea and breathiness sound decreased. The feature importance values for each one of above features (symptoms) is shown in figure 2.

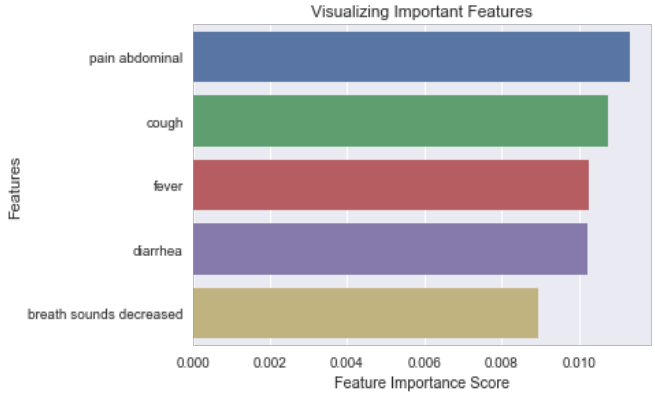


*Fig. 2 Feature importance values of symptoms*

From figure 2 we can interpret that pain abdominal has highest feature importance value as 0.011 and cough as 0.0107. These features would be present in most of the decision trees as internal or leaf nodes while traversing towards classified disease. Higher the values of feature importance, indicates that feature to reduce the impurity across all trees in the forest.

*B. Doctors locator :*

Once our symptom checker has shown the list of probable diseases that the patient might be suffering with, our system will also enable them to locate some of the nearby doctors and physicians treating those diseases. This module of the system would require current location of the patient, in order to locate doctors around him or her. The location can be known with the help of Geo Positioning System (GPS), which will track the current coordinates of patient and use it for locating nearby hospitals. The patient at the end of it would be able to see a list of hospitals near to him on map and in a list format as well.

*C. Price Comparator:*

Since the prices of all branded medicines are a bit higher, but there is a substitute for them in the form generic medicines. These generic medicines have same chemical content as present in branded ones, but are sold at cheaper price as compared to former. Thus, our system will provide a functionality where patient can enter names of prescribed medicines, and will fetch out details of all generic medicines under that prescription.

## IV. CONCLUSION

We have discussed how this application can help person to detect any chances of disease, may be before turn into any serious disease so he or she can consultant doctor. We highlighted how machine learning algorithm can be used to detect the chances of the disease. We have described different tools and technologies that can be used to build a system that will help us to identify the disease. Moreover, depending on the severity of disease our application will also provide the information of the nearby physicians and hospitals using the “GPS” (Geo Positioning System) and Google maps.

## ACKNOWLEDGEMENT

The authors wish to thank the Management and Principal of Datta Meghe College of Engineering, for their support in carrying out this research work. Also would like to thank our guide Prof. Chandrashekhar Raut for guiding us throughout the paper.

## REFERENCES

[1]Md. Tahmid Rahman Laskar, Md. Tahmid Hossain, Abu Raihan MostofaKamal ,Nafiul Rashid ”Automated Disease Prediction System (ADPS): A User Input-based Reliable Architecture for Disease Prediction”International Journal of Computer Applications (0975 – 8887) Volume 133 – No.15, January 2016

[2] Disha Mahajan, MrudulaPhalak, ShubhangiPankore, Saniya Pathan, Deepa Abin “Prediction System for Diseases and Suggestion of Appropriate Medicines” International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 6, Issue 11 November 2017.

[3] Dhanashri Gujar, Rashmi Biyani, TejaswiniBramhane, Snehal Bhosale, Tejaswita P. Vaidya“Disease Prediction and Doctor Recommendation System” International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 05 Issue: 03 Mar-2018.

[4] Miss Swati Y. Dugane (ME, CSIT) Prof. Karuna G. Bagde “Framework for Disease Prediction from Symptoms and Health related data”International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)Volume 6, Issue 9, September 2017.

[5] Harini D K, NateshM ”Prediction of Probability of disease-based o symptoms using machine learning algorithm”International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 05 Issue: 05 May-2018.

[6]http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymtpomKB/index.html