# Symptoms to Disease Mapping and Doctor Recommendation System

**Tapodhir Acharjee, Saurav Chanda, Suman Nunia, Abdul Mazid Choudhury, Sanjeev kumar**

***To find an appropriate doctor who is specialized to treat a certain disease while only symptoms are known is not easy job for the patients. In this paper, we describe a recommended framework to find the best doctors in accordance with patients' requirements. In the proposed system, first it considers only those doctors whose profile match with patients' requirements. Second, the best doctors will be recommended out of previously obtained doctors based on the parameter patients' feedback i.e., patients' review. Our proposal will suggest a doctor recommendation system that uses review mining technique, which can be used in those countries that have huge uneven distribution of medical resources. In our model we have used the decision tree for symptoms to disease mapping and Naive Bayes classifier for sentiment analysis which are connected to each other using a bridge of python logic and the required output is top doctors based on the users input***

***Keywords: Recommendation system, Review mining, Machine learnin, Decision tree, Naïve Bayes Classifier***

## I. INTRODUCTION

With the exponential growth in the quantity and complexity of information sources on the internet, information retrieval systems have evolved from a simple concern with the storage and distribution of artifacts, to encompass a broader concern with the transfer of meaningful information. Over the last twenty years, much effort has gone into the development of approaches to deal effectively with this complexity.

Medical information search refers to methodologies and technologies that seek to improve access to medical information archives via a process of information retrieval. Such information is now potentially accessible from many sources including the general web, social media, journal articles, and hospital records. Health-related content is one of the most searched-for topics on the internet, and as such this is an important domain for Information Retrieval research. the journal, rectification is not possible.

Health information needs are also changing the information seeking behavior and can be observed around the globe. Challenges faced by many people are looking online for health information regarding diseases, diagnoses and different treatments.

**Revised Manuscript Received on October 15, 2019**

**Tapodhir Acharjee*,** Department of Computer Science & Engineering, Assam University, Silchar, Assam , India

**Saurav Chanda,** Department of Computer Science & Engineering, Assam University, Silchar, Assam , India

**Suman Nunia,** Department of Computer Science & Engineering, Assam University, Silchar, Assam , India

**Abdul Mazid Choudhury,** Department of Computer Science & Engineering, Assam University, Silchar, Assam , India

**Sanjeev kumar** Department of Computer Science & Engineering, Assam University, Silchar, Assam , India

If a recommendation system can be made for doctors while using review mining will save a lot of time. So, our recommendation system is designed to help the people who need medical assistance without much difficulty.

1. The main objective of our system is to provide an interface to the patients who wants to find a doctor who is specialized in the concerned disease and has a good reputation.
2. The system has to provide an interface where patients can input their symptoms according to which a doctor can be recommended to them.
3. The patients who is clear about what type of doctor they want to go to can simply insert the speciality, according to which doctors will be recommended to them.
4. The system has to provide such interface where they can give their review for the doctors they visited, which in turn improves the recommendation system.

Various works similar to this proposal were proposed for different domains including medical related works[1-6].

## II. DATASET PREPARATION

### A. Dataset Collection

We collected the reviews from various sources of the internet, the sources were the websites where patients provided their reviews of how they felt about their experience. After we collected all the reviews we tried different models unsupervised classification models but the result wasn't good there was mixed result where the reviews were clustered in an unsatisfying manner. Our dataset contains more than 3000 positive and negative reviews. These reviews have been labeled whether they are positive or negative. The source websites for reviews are[7-9]:

- https://www.healthsoul.com/doctors/
- https://www.healthgrades.com/physician/
- https://www.practo.com/bangalore/doctor/

For the disease-symptom dataset we collected the data of patients from New York Presbyterian Hospital admitted during 2004.This dataset is a knowledge database of disease-symptom associations generated by an automated method based on information. The source of the dataset is [10].

### B. Dataset Processing

To make the dataset ready we used the rating on the reviews to classify if the reviews were positive or negative so for that we made two different files for positive reviews and negative reviews. Some of the source websites didn't have a rating system so for those reviews we had to manually classify them into positive and negative.

### C. Dataset Format

The dataset for doctor ranking contains two columns:

• Reviews: This column contains the reviews that we collected from various websites. The reviews are then processed and should be divided into tokens to be provided as an input in our classifier.

• Liked: This column denotes if the review is positive or negative. 0 denotes that a review is negative and 1 denotes that the review is positive. This column will be most essential to train our supervised classifier as it denotes what type of review is there against it.

**Table 1: Doctor Ranking Table**

| Reviews | Liked |
|---|---|
| Dr. Rose is wonderful. I had rotator cuff surgery in 2015 and was amazed how great I felt day one. I heard so many horror stories about the surgery, but not with Dr. Rose, he is the best. I recently injured my knee and went to see Dr. Rose. Appt was easy to get with a pleasant staff member. Went back today for a follow up from my tests, that were just taken yesterday. Again the appt was easy to get and there was no wait, I never even sat, went right in. Talia was great as were all the staff. | 1 |
| outstanding surgeon ,great pt.care and follow up. | 1 |
| I would absolutely recommend Dr. Levine to anyone that needs an orthopedic surgeon. I am so happy I decided to get surgery myself and the recovery has brought me back to 100% if not even a little better. | 1 |
| Dr. El-Gazzar is the finest surgeon I've ever had the pleasure of meeting. He took care of my son as a charity patient when he shattered his elbow. We were so lucky to have had him as Jesse's surgeon. He has a reputation for arrogance, but he has earned that right, because he's just that good. I unconditionally recommend him to your attention. I don't even care about the extended wait times to see him at the hospital... we were, after all, a charity case | 1 |
| Dr Valenti is phenomenal. She took the time to assess multiple possible etiologies of my headaches, not only checking my vision. Its obvious she truly cares about her patients! | 1 |
| Horrible human and doctor. I had an appointment. Waited over an hour then was put in a room. After another 45 minutes spoke with a PS who took my information. He came back 5 minutes later and said, sorry, he can't prescribe the medication you are on. I asked if I could see the doctor. After waiting again, more time passed and was told to go to someone else and gave me the name of a clinic that couldn't help. | 0 |

The dataset for symptoms to disease mapping contains n columns:

• 0 to *n*-1*th* column: All the columns except the last column contains all the symptoms and the value of these can be either 0 or 1. A 0 denotes that the symptom doesn't cause the disease in the *nth* column and 1 denotes the opposite. There are a total of 132 symptoms in our dataset.

• *nth* column: This column denotes the disease that can be caused by the given symptoms. There are a total of 40 disease in our dataset.

**Table 2: Symptoms to Disease Mapping Table**

| Itching | skin rash | nodal skin eruptions | continuous sneezing | ... | Prognosis |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | ... | Fungal infection |
| 0 | 1 | 1 | 0 | ... | Fungal infection |
| 1 | 0 | 1 | 0 | ... | Fungal infection |
| 1 | 1 | 0 | 0 | ... | Fungal infection |
| 0 | 0 | 0 | 1 | ... | Allergy |
| 1 | 0 | 0 | 0 | ... | Chronic cholestasis |
| 1 | 0 | 0 | 0 | ... | Chronic cholestasis |
| 0 | 0 | 0 | 0 | ... | Allergy |

## III. PROPOSED MODEL

The proposed doctor recommendation system consists of mainly three parts, namely clinical data as input, the PHRS and the doctor recommendation as output. The main components of the proposed system are shown in the figure below:

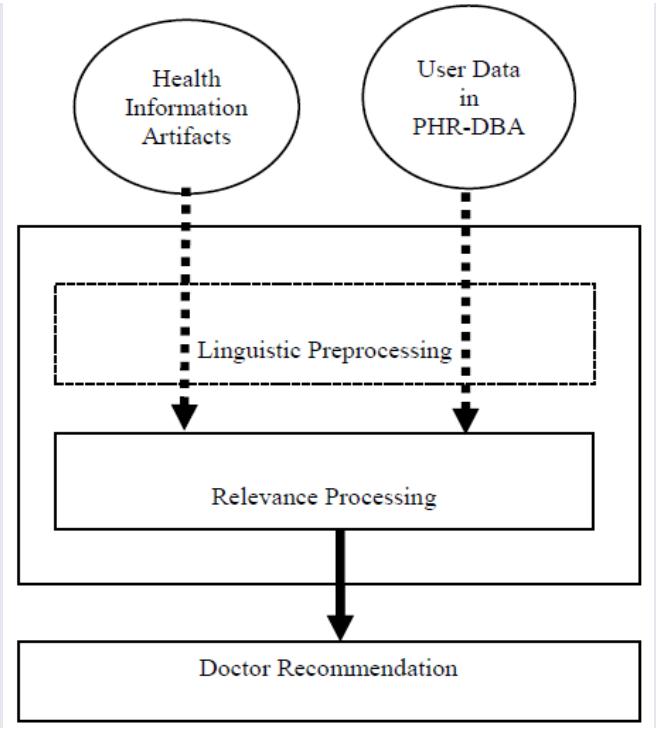


**Figure1: Abstract Doctor Recommendation System**

And using the abstract model we have proposed our own model for the system. This model also has three main modules namely the:

• Symptoms to disease mapping
• Disease to specialty logic
• Doctor rank

The inputs of the PHRS system are:

• Clinical data
• Review data

And finally, the output of the system is Doctor recommendation.

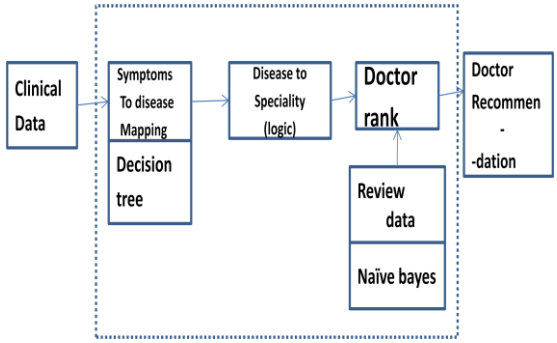


**Figure 2: Modules of Doctor Recommendation System**

The modules of the proposed model are described below:

**A.Clinical Data**

The clinical data is the input to PHRS system. It includes the symptoms of the users using the system. The symptoms can be anything starting from fever, cold, cough to all the other problems they are facing. The symptoms given here decides the disease the person is suffering from.

**B. Symptoms to Disease Mapping**

The input provided to PHRS system are the symptoms like headache, high temperature, and body pain and these symptoms will be used to map the disease user is suffering from. Decision Tree is a type of Supervised Machine Learning technique where data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves.

**Decision Tree**

A Decision tree[10] is a type of Supervised Machine Learning where 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 of the final outcomes. The decision nodes are where the data is split. An example of a decision tree can be explained using the following binary tree.

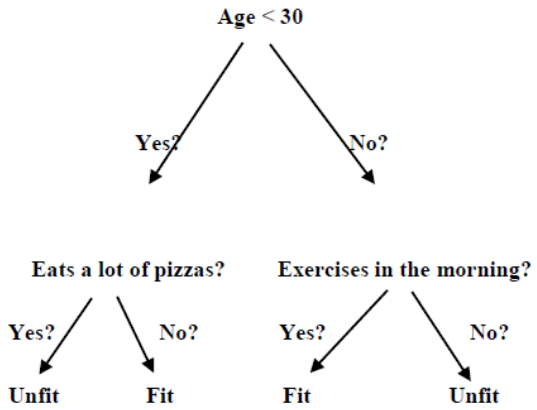


**Figure 3: An Example of Decision Tree**

Let's say we want to predict whether a person is fit given their information like age, eating habit, and physical activity. The decision nodes here are questions like 'What's the age?' 'Does he exercise?' 'Does he eat a lot of pizzas?' And leaves are outcome whereas the result is fit or unfit.

**Working**

Now that we know what a Decision Tree is, we'll see how it works internally. Before discussing the working, we'll go through a few definitions.

• **Entropy[11]**

Entropy, 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 the probability of tails is 0.5. Alternatively, consider a coin that has heads on both sides, the entropy of such an event can be predicted perfectly since we know beforehand that it'll always be headed.

• **Information Gain[12]**

Information gain 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.

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

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.

**Example**

Let's understand this with the help of an example Let us consider data collected for two diseases where the disease gets decided on the basis of their symptoms.

| Data | Itching | Skin Rash | Shivering | Joint Pain | Prognosis |
|---|---|---|---|---|---|
| D1 | 1 | 1 | 0 | 0 | Allergy |
| D2 | 0 | 0 | 1 | 0 | Fever |
| D3 | 0 | 0 | 1 | 1 | Fever |
| D4 | 1 | 0 | 0 | 0 | Allergy |
| D5 | 0 | 1 | 0 | 0 | Allergy |
| D6 | 0 | 0 | 0 | 1 | Fever |
| D7 | 0 | 1 | 0 | 1 | Allergy |

**Table 3: Symptoms to disease data**

Now, our job is to build a predictive model that takes in above 4 parameters and predicts whether the disease is allergy or fever.

Now we shall go ahead and grow the decision tree. The initial step is to calculate H(S), the Entropy of the current state.

We know,

$$Entropy(s) = \sum_{x \in X} P(x) \log 2 \frac{1}{P(x)}$$

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

as we can see there are total 4 examples of allergy and 3 examples of fever, therefore,

Entropy(S)= - (4/7)$\log_2$ 4/7 - (3/7)$\log_2$ 3/7= 0.985

Next, step is to find the highest possible Information gain which we will choose as root node. Let's start with **Itching**

We have to find out information gain,

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

For that we have to find

1. H(S1)
2. H(S0)
3. P(S1)
4. P(S0)
5. H(S) = 0:985 which we calculated above.

Among all 7 examples we can see we have 2 places where **Itching** is 1 and 5 places where **Itching** is 0, so we have to calculate the probability and entropy as follows:

P(S*1*) = Itching1/Total

P(S*1*) = 2/7 = 0.285

P(S*0*) = Itching0/Total

P(S*0*) = 5/7 = 0.71

Now we have 2 examples where **Itching** = 1 and both of the cases we have **allergy** So, we have,

Since, we have no randomness,

Entropy(S1) = 0

but, for itching=0 we have 2 cases of **allergy** and 3 cases of **fever**,

Entropy(S0) = - (2/5)$\log_2$ 2/5 - (3/5)$\log_2$ 3/5= 0.97

Since, we have all the required piece now we can calculate the information gain,

IG (S; Itching) = H(S) - P(S0) H(S0) - P(S1) H(S1)

IG (S; Itching) = 0.985 – 0.71 * 0.97 - P(S1) * 0 = 0.29

Similarly, for **Skin Rash**

Among all 7 examples we can see we have 3 places where Skin rash is 1 and 4 places where Skin rash is 0,

P(S*1*) = Skinrash1/Total

P(S*1*) = 3/7 = 0.42

P(S*0*) = Skinrash0/Total

P(S*0*) = 4/7 = 0.57

Now we have 2 examples where **Skin rash** = 1 and all of the cases we have allergy So, we have,

Since, we have no randomness,

Entropy(S1) = 0

but, for **itching**=0 we have 1 cases of allergy and 3 cases of fever,

Entropy(S)= - (1/4)$\log_2$ 1/4 - (3/4)$\log_2$ 3/4= 0.81

Since, we have all the required piece now we can calculate the information gain,

IG (S; Skinrash) = H(S) - P(S0) H(S0) - P(S1) H(S1)

IG (S; Skinrash) = 0.985 – 0.57 * 0.81 - P(S1) * 0 = 0.52

Similarly, we can find

3. IG (S; Shivering) = 0:47
4. IG (S; Joint pain) = 0:14

So, final results are follows:

1. IG (S; Itching) = 0:29
2. IG (S; Skin rash) = 0:52
3. IG (S; Shivering) = 0:47
4. IG (S; Joint pain) = 0:14

Since **Skin rash** has the highest Information Gain, we will use **Skin rash** as the root node.

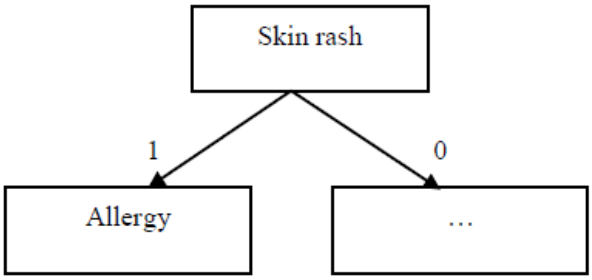


**Figure 4: Decision Tree Phase One**

**Table 4: Symptoms to disease data after phase 1**

| Data | Itching | Shivering | Joint | Prognosis |
|---|---|---|---|---|
| D2 | 0 | 1 | 0 | Fever |
| D3 | 0 | 1 | 1 | Fever |
| D4 | 1 | 0 | 0 | Allergy |
| D6 | 0 | 0 | 1 | Fever |

Now, since skin rash has been put into the tree, we are left with three more features so we will repeat the same procedure with the next three features,

For the following table **Skin rash**=1, we have,

Entropy(S)= - (1/4)$\log_2$ 1/4 - (3/4)$\log_2$ 3/4 = 0.81

We have to find the Information Gain for **Itching**, **Shivering**, **Joint pain**. for **Itching** For calculating IG (S, Itching) we have to find

1. H(S1)
2. H(S0)
3. P(S1)
4. P(S0)
5. H(S)=0.81 which we calculated above

Among all 4 examples we can see we have 1 place where **Itching** is 1 and 3 places where **Itching** is 0,

P(S*1*) = Itching1/Total

P(S*1*) = 1/4 = *0.25*

P(S*0*) = Itching0/Total

P(S*0*) = *3/4 = 0.75*

Now we have 1 example where **Shivering** = 1, we have result fever

Since, we have no randomness,

Entropy(S1) = 0

but, for **Shivering**=0 we have 1 case of allergy and 1 case of fever,

Entropy(S0) = - (1/2)$\log_2$ 1/2 - (1/2)$\log_2$ 1/2 = 0.50

Since, we have all the required piece now we can calculate the information gain,

IG (S; Shivering) = H(S) - P(S0) H(S0) - P(S1) H(S1)

IG (S; Shivering) = 0.81 – 0.5 * 0.5 - 0 = 0.56

Similarly, for **Joint pain**

Among all 4 examples we can see we have 2 places where **Joint pain** is 1 and 2 places where **joint pain** is 0,

P(S*1*) = jointpain1/Total

P(S*1*) = 2/4 = *0.50*

P(S*0*) = jointpain0/Total

P(S*0*) = 2/4 = 0.50

Now we have 1 example where **joint pain** = 1, we have result fever.

Since, we have no randomness,

Entropy(S1) = 0

but, for **joint pain**=0 we have 1 case of allergy and 1 case of fever,
Entropy(S0) = - (1/2)$\log_2$ 1/2 - (1/2)$\log_2$ 1/2 = 0.50
Since, we have all the required piece now we can calculate the information gain,
IG (S; Joint pain) = H(S) - P(S0) H(S0) - P(S1) H(S1)
IG (S; joint pain) = 0.81 – 0.5 ∗ 0:5 - 0 = 0.56
So, we can see
IG (S; Itching) = 0
IG (S; Shivering) = 0.56
IG (S; Joint Pain) = 0.56
Since **Shivering** and **Joint pain** have same Information Gain, we can take any one of them, in our case let's make **Shivering** as next node.

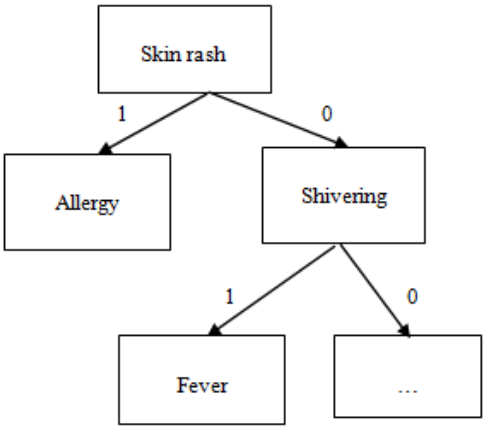


**Figure 5: Decision Tree Phase Two**

Now both **skin rash** and **shivering** has been placed in the graph hence the dataset for the leftover process is:

**Table 5: Symptoms to disease data after phase two**

| Data | Itching | Joint Pain | Prognosis |
|---|---|---|---|
| D4 | 1 | 0 | Allergy |
| D6 | 0 | 1 | Fever |

From the above table we can see clearly that both the features have similar information gain so we can choose any one of them and place it in the graph.
Hence, the final decision tree is given below,

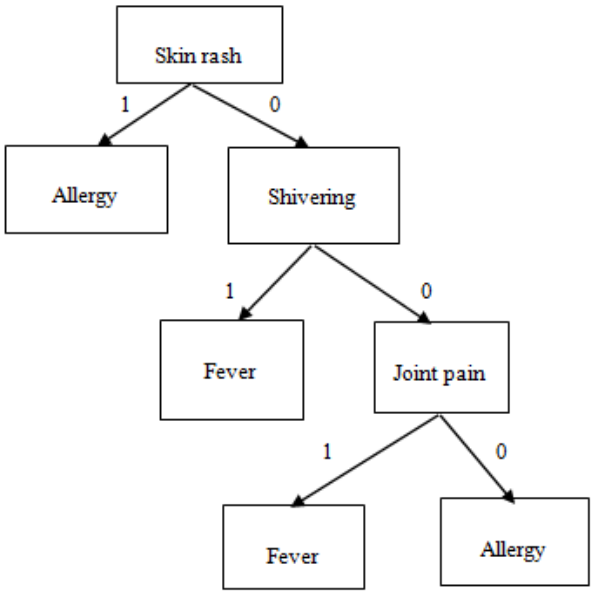


**Figure 6: Final Decision Tree**

### C. Disease to speciality logic

As the disease has been mapped with the help of a decision tree the very next step is to suggest the speciality for the specified disease provided by the user. For this purpose, we have made a MySQL table with two columns: Disease and Speciality
Disease column: The disease column contains all the possible diseases that can be detected by our decision tree. These diseases are to be mapped to a particular specialty. Speciality column: This column contains the specialty that would be mapped to a particular disease. Though there can be multiple diseases mapped to a specialty but for a single disease, there can be only one specialty.

So, when the previous module (Symptoms to Disease Mapping) gives its output we map the output disease to a specialty according to the table and pass it to the next module (Doctor
ranking).

### D. Doctor Ranking

#### Review Data

We have collected reviews of doctors from various websites (both positive and negative reviews) but the review has to be processed before using it as a input to the classifier. The steps for pre-processing are as follows: The Health Information Artifacts was the collection of reviews from various sources and used Naive Bayes classifier to sort it into positive reviews and negative reviews by using labeled data. Using the classified data, we will make a rank table. The rank table is made according to how many positive reviews a person has. Before using the review data, the review is vectorized i.e. divided into tokens. Then the data is cleaned and forwarded into the classifier for removal of stopwards and classification.
**Tokenization:** After the input is taken the next step is Tokenization. Here the review is divided into tokens i.e., words. `For example, the doctor was great!`:

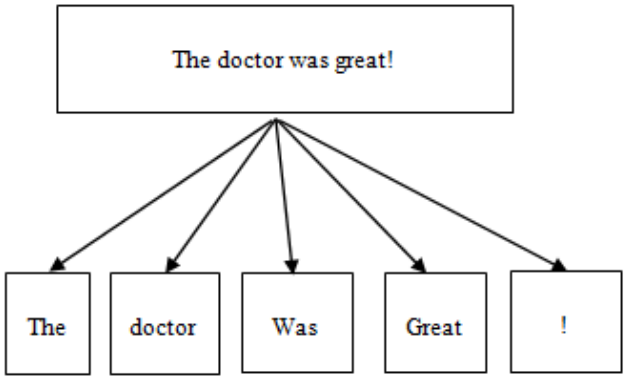


**Figure 7: Tokenization**

**Cleaning the data:** Tokenization is followed by Cleaning of Data, where the data is cleaned i.e., special characters are removed from the review.

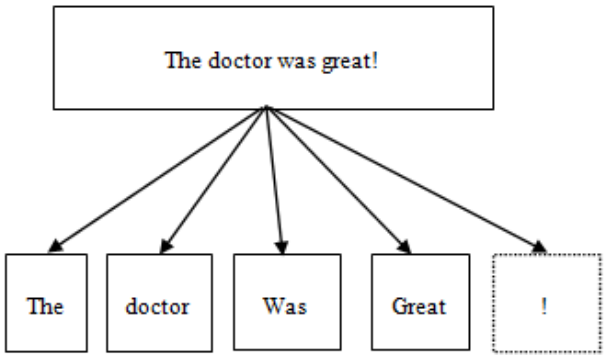


**Figure 8: Cleaning the Data**

**Removing stopwords:** The next step is Removing the stopwords i.e., unwanted words such as 'the', 'of',' in' etc. In computing, Stopwords are those words that are filtered out before and after the processing of natural data. The stopwords are then removed and the processed data is ready for classification. Example:

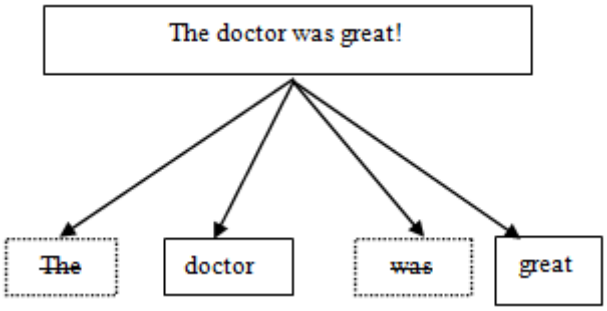


**Figure 9: Removing Stop Words**

**Naïve Bayes Classifier[13]**

The classifier we are using is the Naive Bayes Classifier to divide the reviews into positive and negative. The simplest solutions are usually the most powerful ones, and Naive Bayes is good proof of that. It has been successfully used for many purposes, but it works particularly well with natural language processing (NLP) problems. Naive Bayes Naive Bayes are a family of powerful and easy-to-train classifiers, which determine the probability of an outcome, given a set of conditions using the Bayes theorem. Naive Bayes is multi-purpose classifiers and it's easy to find their application in many different contexts. A good example is given by natural language processing, where a text can be considered as a particular instance of a dictionary and the relative frequencies of all terms provide enough information to infer a belonging class. Our examples may be generic, so to let you understand the application of naive Bayes in various contexts.

$$P(y|x_1, x_2, x_3 \ldots xm) = \alpha P(y) \prod P(x_i|y)$$

Now our model distinguishes the reviews whether it is a positive or a negative review. From those reviews, we can rank the doctors' based on their positive reviews. To understand this let us take an example.

**Example of Sentiment analysis using Naive Bayes**

Let's see how this works in practice with a simple example. Suppose we are building a classifier that says whether a text is about sports or not. Our training data has 5 sentences:

**Table 6: Review containing stopwords**

| Text | Tag |
|---|---|
| "A great doctor" | Positive |
| "The treatment was the worst" | Negative |
| "Best in town" | Positive |
| "A very friendly doctor" | Positive |
| "Too much meds" | Negative |

After removing stopwords,

**Table 7: Removing stopwords**

| Text | Tag |
|---|---|
| "~~A~~ great doctor" | Positive |
| "~~The~~ treatment ~~was~~ the worst" | Negative |
| "Best ~~in~~ town" | Positive |
| "~~A~~ very friendly doctor" | Positive |
| "~~Too~~ much meds" | Negative |

Now, which tag does the sentence "A great doctor and friendly one" belongs to? Since Naïve Bayes is a probabilistic classifier, we want to calculate the probability that the sentence is Positive or Negative. To calculate the probabilities we have to note this two very important points :

• **Bayes' Theorem[14]**

Now we need to transform the probability we want to calculate into something that can be calculated using word frequencies. For this, we will use some basic properties of probabilities, and Bayes' Theorem. Bayes' Theorem is useful when working with conditional probabilities, because it provides us with a way to reverse them:

$$P(A/B) = \frac{P(B/A) \times P(A)}{P(B)}$$

In our case, we have P(Positive |*A great doctor and friendly one*), so using this theorem we can find the conditional probability:

P(positive/a great doctor and friendly one)
= P(a great doctor and friendly one / positive) × P(positive) / P(a great doctor and friendly one)

Since for our classifier we're just trying to find out which tag has a bigger probability, we can discard the divisor which is the same for both tags and just compare

P(a great doctor and friendly one/Positive) × P(Positive)

With P(a great doctor and friendly one/Negative) × P(Negative)

• **Being Naive**

So here comes the Naive part: we assume that every word in a sentence is independent of the other ones. This means that we're no longer looking at entire sentences, but rather at individual words. So for our purposes, "this was a fun party" is the same as "this party was fun" and "party fun was this". We write this as: P(A great doctor and friendly to everyone)

= P(a) × P(great) × P(doctor) × P(and) × P(friendly) × P(one)

P(A great doctor and friendly to everyone/positive) = P(a/positive) × P(great/positive) × P(doctor/positive) × P(and/positive) ×

P(friendly/positive) × P(one/positive)

**Calculating probabilities**, The final step is just to calculate every probability and see which one turns out to be larger. Calculating a probability is just counting in our training data. First, we calculate the probability of each tag: for a given sentence in our training data, therefore the probability is

P(positive) is 3/5

*P*(*negative*) *is* 2/5

**CALCULATING FINAL PROBABILITY**

**Table 8: Calculating Probability for first sentence**

| Words | P(Words/Positive) | P(Words/Negative) |
|---|---|---|
| Great | (1+1) / (6+10) = 0.12 | (0+1) / (4+10) = 0.07 |
| Doctor | (2+1) / (6+10) = 0.18 | (1+1) / (4+10) = 0.14 |
| Friendly | (1+1) / (6+10) = 0.12 | (0+1) / (4+10) = 0.07 |
| One | (0+1) / (6+10) = 0.06 | (0+1) / (4+10) = 0.07 |

P(great/positive) × P(doctor/positive) × P(friendly/positive) × P(one/positive) = 0:000155

P(great/negative) × P(doctor/negative) × P(friendly/negative) × P(one=negative) = 0:000048

Hence, our classifier gives a result of Positive. Similarly, for the review "tons of meds but the worst treatment"

**Table 9: Calculating Probability for second sentence**

| Words | P(Words/Positive) | P(Words/Negative) |
|---|---|---|
| Tones | (0+1) / (6+10) = 0.06 | (0+1) / (4+10) = 0.07 |
| Meds | (0+1) / (6+10) = 0.06 | (1+1) / (4+10) = 0.14 |
| Worst | (0+1) / (6+10) = 0.06 | (1+1) / (4+10) = 0.14 |
| Treatment | (0+1) / (6+10) = 0.06 | (1+1) / (4+10) = 0.14 |

P(tons/positive) × P(meds/positive) × P(worst/positive) × P(treatment/positive) = 0:0000129

P(tons/negative)×P(meds/negative)×P(worst/negative)×P(treatment/negative) = 0:00019

Hence, our classifier gives a result of Negative.

### E. Doctor Recommendation

The final output of the whole system is Doctor Recommendation. This phase finally pops out the top ranked doctors using the rank-table produced by the Naive Bayes' classifier.

## IV. RESULTS AND DISCUSSIONS

### A. Implementations

Our PHRS (Personal Health Recommendation System) has three modules namely Symptoms to Disease Mapping, Disease to Speciality Logical and Doctor Ranking. We have implemented the Symptoms to Disease mapping using a Decision tree so that the mapping would take as little time as possible.

The decision tree was successful in that attempt, and our mapping requirement was to map simple symptoms to a simple disease. The result does not have to be spot on since according to the disease a doctor will be recommended and further diagnosis would be carried out by the recommended doctor.

On the other hand, the Health Information Artifacts (Reviews provided by the patients) is the input to the linguistic processing module. But we found that doctors don't have many negative reviews since doctors are respected highly in society. Our result of the Naive Bayes classifier is satisfactory but it can be better hence we are working on a classifier that uses lexicon based along with classifier based analysis which may provide better results.

### B. Results

The Naive Bayes classifier used in this model gives us the best result compared to the Support Vector Machine. The accuracy of Naive Bayes is 93% and the data fitting time is 0.00501ms. The Decision Tree used for mapping disease to speciality gives an accuracy of 95.7%.

In the doctor rank module the review dataset acts as a input in our classifier which gives us the number of positive reviews for a particular doctor. The goal of making a rank table is that we don't have to run the classifier over and over every time we need to recommend a doctor, the system just searches a doctor from the rank table which is way faster.

**Table 10: Top doctors with their respective scores from our rank table**

| ID | Name | Number of positive review | Score |
|---|---|---|---|
| 156 | Dr. Justin Greisberg | 79 | 100 |
| 297 | Dr. William Levine | 60 | 75. 94936709 |
| 9 | Dr. Adeeb Ahmed | 52 | 65.82278481 |
| 114 | Dr. Geoffrey Bland | 50 | 63.29113924 |
| 5 | Dr Misty Phillips | 46 | 58.2278481 |
| 276 | Dr. Stephen Silver | 42 | 53.16455696 |
| 47 | Dr. Charles Jobin | 39 | 49.36708861 |
| 106 | Dr. Filamer Kabigting | 39 | 49.36708861 |

| | | | |
|---|---|---|---|
| 174 | Dr. Lindsey Bordone | 38 | 48.10126582 |
| 232 | Dr. Philip Garcia | 36 | 45.56962025 |
| 19 | Dr. Amy Hall | 34 | 43.03797468 |
| 62 | Dr. Daniel Seigerman | 31 | 39.24050633 |
| 277 | Dr. Steven Beldner | 31 | 39.24050633 |
| 154 | Dr. Joshua Renken | 30 | 37.97468354 |

The result of the module symptom to disease mapping is a Decision tree in which the leaves are the diseases and in each node the data splits according to the query of the symptoms. We used a decision tree since it only needs 0.3948 ms to fit such a large dataset and only needs an average of 0.001002 ms to predict a query.

### C. Analysis

The Classifier based approach for Sentiment Analysis for a single domain gives a very excellent result but it works really bad if the domain is switched. This is a desirable property as sentiment analysis using a correct resource is bound to perform better, and there are times when correctness requires innate human judgment while classifiers may get misled.
Other than the Naive Bayes classifier we have used Support Vector Machine initially but the result was not satisfactory. Though both of them have comparable performance and are the most used classifiers in sentiment analysis we used Naive Bayes because of the slight advantage in time, for our dataset. As our domain is specific to doctors review so we have used the classifier-based approach as it gives a very accurate result for a specific result.

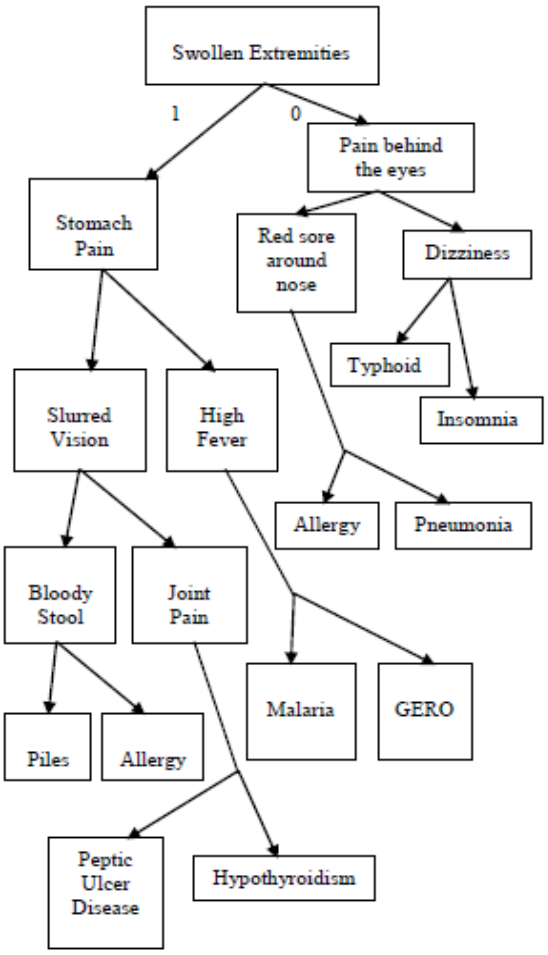


**Figure 10: A snippet of the result Decision Tree**

Since the prediction from the decision tree is not always correct, it is corrected in the disease to speciality module. The main objective of these two module is finally to suggest what type of doctor is necessary for the given symptoms.

**Table 11: Accuracy and Fitting time in Naive Bayes and SVM**

| | Naive Bayes | Support Vector Machine |
|---|---|---|
| Accuracy | 96.875% | 93.721% |
| Fitting Time | 0.00501 ms | 0.14524 ms |

## V. CONCLUSION AND FUTURE SCOPE

We have developed a hybrid model using Decision Tree and Naive Bayes classifier for doctor recommendation. In our model we have used the decision tree for symptoms to disease mapping and Naive Bayes classifier for sentiment analysis to find the required output that is the recommended doctors based on the users input. The system remain to be implemented and tested with more data and using some of the current machine learning/deep learning methods.

## REFERENCES


1. Alexander Pak , Patrick Paroubek, “Twitter based system: Using Twitter for disambiguating sentiment ambiguous adjectives”, Proceedings of the 5th International Workshop on Semantic Evaluation, p.436-439, July 15-16, 2010.
2. S. Tan., X. Cheng., Y.Wang H. Xu, “Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In: Advances in Information Retrieval. ECIR 2009. Lecture Notes in Computer Science, vol 5478. Springer, Berlin, Heidelberg, vol 5478. Springer,Berlin, Heidelberg.(2009).
3. Monica Mandage, Sahil Shah, Pragati Vhatkar, Anita Waghmode, Kirti Wanjale, "Recommendationof Doctors and Medicine Using Review Mining". International Journal of Innovative Research in Science,Engineering and Technology, Vol 8, issue 1, 2019.
4. E. Sezgin and S. Ozkan, "A systematic literature review on Health Recommender Systems,"2013 E-Health and Bioengineering Conference (EHB), Iasi, 2013, pp. 1-4. doi: 10.1109/EHB.2013.6707249
5. F. Ricci, L. Rokach, B. Shapira and P. B. Kantor, "Introduction to Recommender Systems Handbook", Springer, Berlin, 2011.
6. C. Lee, M. Lee, and D.A. Han, Framework for Personalized Healthcare Service Recommendation. Health", Proceedings of 10th International Conference on e-health Networking, Applications and Services, 7-9 July 2008.
7. https://www.healthsoul.com/doctors/
8. https://www.healthgrades.com/physician/
9. https://www.practo.com/bangalore/doctor
10. http://people.dbmi.columbia.edu/friedma/Projects/DiseaseSymptomKB/index.html
11. P. H. Swain, H. Hauska, "The decision tree classifier: design and potential", *IEEE Trans. Geosci. Electron.*, vol. GE-15, pp. 142-147, July 1977
12. C. Shannon, Elwood , "A Mathematical Theory of Communication" (PDF). *Bell System Technical Journal*. **27** (3): 379–423,1948
13. Quinlan, J. Ross,"Induction of Decision Trees". *Machine Learning*. **1** (1): 81–106, 1986.
14. P Domingos, M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss". *Machine Learning*. **29** (2/3): 103–137, 1997.
15. Bayes Thomas and Price LII. “An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S**53** *Phil. Trans. R. Soc*

## AUTHORS PROFILE

**Dr Tapodhir Acharjee** is working as an Assistant Professor in the department of Computer Science and Engineering in Assam University, Silchar. His areas of interests are ad-hoc networks, cryptography and network security, machine learning etc.

**Saurav Chanda** received B. Tech degree in Computer Science & Engineering from Assam University, Silchar, Assam, India in 2019. His interests include machine learning and data mining and software development.

**Suman Nunia** received B. Tech degree in Computer Science & Engineering from Assam University, Silchar, Assam, India in 2019. His interests include machine learning and data mining and web development.

**Abdul Mazid Choudhury was pursuing** B.Tech. degree in the department of Computer Science and Engineering from Assam University, Silchar, India in 2016-2019 batch

**Sanjeev Kumar** received his B.Tech. degree in Computer Science and Engineering from Assam University, Silchar and currently pursuing M.Tech. in the same department. His area of interests are Cryptography, Network Security, machin learning etc.