Abstract

To find the specialty-counterpart, diagnosis-accurate, skill-superb and cost-effective doctors is not easy job for the patients. In this report, we describe a recommender frameword 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 report 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.

Chapter 1

Introduction

1.1 Information Retrieval

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.

Information retrieval, as the name implies, concerns the retrieving of relevant information from databases. It is basically concerned with facilitating the user's access to large amounts of (predominantly textual) information. The process of information retrieval involves the following stages:

- Representing Collections of Documents - how to represent, identify and process the collection of documents.
- User-initiated querying - understanding and processing of the queries.
- Retrieval of the appropriate documents - the searching mechanism used to obtain and retrieve the relevant documents

1.2 Information Retrieval in medical field

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. Medical information is of interest to a wide variety of users, including patients and their families, researchers, general practitioners and clinicians, and practitioners with specific expertise such as radiologists. There are several dedicated services that seek to make this information more easily accessible, such as the 'Health on the Net' system for the general public and medical practitioners. However, despite the popularity of the medical domain for users of search engines, and current interest in this topic within the IR research community, development of search and access technologies remains particularly challenging.

A central issue in medical IR is the diversity of the users of these services. In particular, they will have varying categories of information needs, varying levels of medical knowledge, and varying language skills.

These challenges can be summarized as follows:

- Varying information needs: While a patient with a recently diagnosed condition will generally benefit most from simple or introductory information on the disease and its treatment, a patient living with or managing a condition over a longer term will generally be looking for more advanced information, or perhaps support groups and forums. Similarly, a general practitioner might require basic information quickly while advising a patient, but more detailed information if deciding a course of treatment, and a specialist clinician might look for an exhaustive list of similar cases or research papers relating to the condition of a patient that they are currently seeking to advise. Understanding of various types of users and their information needs is one of the cornerstones of medical Information Retrieval development of effective, potentially personalized systems that addresses these needs, is one of the greatest challenges.
- Varying medical knowledge: The different categories of users of medical IR systems have different levels of medical knowledge, and indeed the medical knowledge of different individuals within a category can also vary greatly. This affects the way in which individuals pose search queries to systems and also the level of complexity of information which should be returned to them or the type of support in understanding of retrieved material which should be provided.
- Varying language skills: Given that much of medical content is written in the English language, research to date in medical information search has predominantly focused on monolingual English retrieval. However, given the large number of non-English speakers on the Internet and the lack of content in their native language, effective support for them to search English sources is highly desirable.

Our model is made for those users seeking doctor information according to the symptoms they are facing. Even users with little or no medical knowledge a user can successfully can get recommended doctors easily from our doctor recommendation system. Although our system faces these above challenges but we were able to solve some of them by using various machine learning techniques. Our sentiment analysis model can even classify some reviews written in other language into positive and negative with enough data. And our model is targeted specifically towards patients who have very little medical knowledge.

Chapter 2

Machine Learning Techniques And Sentiment Analysis

Machine learning is a data analytic technique that teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to learn information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases.

2.1 How Machine Learning Works?

Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data.

2.2 Supervised Learning

Supervised machine learning builds a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Use supervised learning if you have known data for the output you are trying to predict. Supervised learning uses classification and regression techniques to develop predictive models. Classification techniques predict discrete responses for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring. Use classification if your data can be tagged, categorized, or separated into specific groups or classes. For example, applications for hand-writing recognition use classification to recognize letters and numbers. In image processing and computer vision, unsupervised pattern recognition techniques are used for object detection and image segmentation. Common algorithms for performing classification include support vector machine (SVM), boosted and bagged decision trees, k-nearest neighbor, Nave Bayes, discriminant analysis, logistic regression, and neural networks. Regression techniques predict continuous responses for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading. Use regression techniques if you are working with a data range or if the nature of your response is a real number, such as temperature or the time until failure for a piece of equipment. Common regression algorithms include linear model, nonlinear model, regularization, step-wise regression, boosted and bagged decision trees, neural networks, and adaptive neuro-fuzzy learning.

2.3 Unsupervised Learning

Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses. Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for cluster analysis include gene sequence analysis, market research, and object recognition. For example, if a cell phone company wants optimize the locations where they build cell phone towers, they can use machine learning to estimate the number of clusters of people relying on their towers. A phone can only talk to one tower at a time, so the team uses clustering algorithms to design the best placement of cell towers to optimize signal reception for groups, or clusters, of their customers. Common algorithms for performing clustering include k-means and k-medoids, hierarchical clustering, Gaussian mixture models, hidden Markov models, selforganizing maps, fuzzy c-means clustering, and subtractive clustering.

2.4 How to Decide Which Machine Learning Algorithm to Use?

Choosing the right algorithm can seem over whelming there are dozens of supervised and unsupervised machine learning algorithms, and each takes a different approach to learning. There is no best method or one size fits all. Finding the right algorithm is partly just trial and error even highly experienced data scientists cant tell whether an algorithm will work without trying it out. But algorithm selection also depends on the size and type of data you're working with, the insights you want to get from the data, and how those insights will be used. Choose supervised learning if you need to train a model to make a prediction– for example, the future value of a continuous variable, such as temperature or a stock price, or a classification for example, identify makes of cars from web-cam video footage. Choose unsupervised learning if you need to explore your data and want to train a model to find a good internal representation, such as splitting data up into clusters.

2.5 Sentiment analysis

Sentiment Analysis is one of the most important applications of Natural Language Processing. It refers to the study of extraction of opinions from text[1].

There are two approaches to Sentiment Analysis :

- Classier-based approach : which treats Sentiment Analysis as a special case of text classication and uses standard Machine Learning techniques to solve the problem. Various classifiers such as SVM and Naive Bayes are used to successfully get a impressive result but suffers from the domain transfer problem.
- lexicon-based approach : which uses sentiment lexicons – dictionaries of words with labels specifying their sentiments to identify the sentiment of text[1].Though these approach doesn't suffer from domain transfer problem but they have less accuracy. The lexicon based approach often uses a dictionary of pre-weighted words but the problem comes when there is a language for which data is not available.

2.5.1 Limitations of both methods Sentiment Analysis

The classifier-based approach is the dominant approach towards sentiment analysis in literature, its performance is not up to the mark. The reason for this change in performance is that sentiment analysis is fundamentally different from text classification and is therefore not as suited for machine-learning techniques. The following are some fundamental challenges due to which classifier methods are not optimally suitable for sentiment analysis:

- **Domain-specificity** Classifier-based methods work well when trained on a corpus of a particular domain, which is why text classification performs so well using classifier methods. However, this is primarily because the classifier learns several features that are domain specific and may not hold in other domains or even cause sentiment drift.
- **Lack of Context** During the feature extraction stage, a document is vectorized into a bit representation. This may preserve some information from the document at the expense of leaving out other, possibly vital information, mainly context. For instance, using unigram features, information about the order of words is entirely lost, and it is not feasible to use higher order n-grams to capture long-distance dependencies As a result, the following phrases all look very similar to a classifier even though the polarities are vastly different – "good" , "not good" , "not very good" , ". . . do not think that this is any good", etc. In addition, the document-level granularity further exacerbates the problem.

The problem with Lexicon based approach is the immense time investment required. Taking a very conservative estimate of 90 seconds required for actually annotating the sentiment of the word and 30 seconds for post-processing per word to enter it in the database, this requires over 1,200 days working for 8 hours a day without any breaks. Due to this, the sizes of manual sentiment lexicons have been restricted to a few thousand words at most, adversely affecting coverage.

Chapter 3

Literature Survey

Number of papers have been published on several data mining techniques for Sentiment analysis with different advantages and disadvantages depending on the domain and type of dataset.

- Ahire, S.G. (2015)[1]. A Survey of Sentiment Lexicons.

  This is a survey paper that introduces sentiment lexicons and explains the state of the art in the field of sentiment lexicons. Different kinds of lexicons are covered, varying in aspects such as coverage, methods of creation, lexical unit and granularity. It aims at giving a representative sampling of the field of sentiment lexicons.

  Sentiment Analysis is one of the most important applications of Natural Language Processing. There are two approaches to Sentiment Analysis – the classifier-based approach and the lexicon-based approach. The classifier based approach treats Sentiment Analysis as a special case of text classification and uses standard Machine Learning techniques to solve the problem. While the lexicon-based approach uses sentiment lexicons – dictionaries of words with labels specifying their sentiments – to identify the sentiment of the text. But both the approaches have their own limitations.

  This report presents the lexicon-based approach to sentiment analysis. To that end, it describes the current state-of-the-art in sentiment lexicons. As the classifier based approach has two main problems namely the Domain-specificity and the Lack of Context. To summarize, this paper covered three sentiment lexicons – SO-CAL, SentiWordnet, and Sentiment Treebank. SO-CAL was created manually and associates an integer between 5 and +5 to a word. SentiWordnet was created automatically and associates three values representing positivity, negativity, and objectivity of a Wordnet synset. As it was created automatically, it has high coverage. Sentiment Treebank was created using crowdsourcing and associates a label between 'very positive' and 'very negative' to a phrase. As a result, one can see that the landscape of sentiment lexicons is highly varied.

- Pak, Alexander and Paroubek, Patrick. (2010)[2]. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Proceedings of LREC. 10.

  This paper used Twitter as a source of reviews for Sentiment analysis. The contributions of their paper are as follows:

1. A method to collect a corpus with positive and negative sentiments, and a corpus of objective texts. This method allows for collecting negative and positive sentiments such that no human effort is needed for classifying the documents. Objective texts are also collected automatically. The size of the collected dataset can be arbitrarily large.
2. Perform statistical linguistic analysis of the collected corpus.
3. We use the collected corpora to build a sentiment classication system for microblogging.
4. And conduct experimental evaluations on a set of real microblogging posts to prove that the presented technique is efcient and performs better than previously proposed methods.

They trained a Naive Bayes classifier by using n-grams as features. The authors were able to get up to 81% of accuracy on their test data but the result was bad for 3 categories ("negative", "positive" and "neutral").

This report has illustrated that an effective sentiment analysis can be performed on a television program by collecting a sample audience opinion from Twitter. Throughout the duration of this project many different data analysis tools were employed to collect, clean and mine sentiment from the dataset. Such an analysis could provide valuable feedback to producers and help them to spot a negative turn in viewer's perception of their show. Discovering negative trends early on can allow them to make educated decisions on how to target specific aspects of their show in order to increase its audience's satisfaction. It is apparent from this study that the machine learning classifier used has a major effect on the overall accuracy of the analysis. Commonly used algorithms for text classification were examined such as Naïve Bayes, Decision Tree, Support Vector Machine, and Random Forests.

- Narayanan, Vivek et al. "Fast and accurate sentiment classification using an enhanced Naive Bayes model." IDEAL (2013)[3].
  The authors have tried different methods to improve the accuracy of a Naive Bayes classifier for sentiment analysis. They achieved an accuracy of 88.80% on the popular IMDB movie reviews dataset, by using a combination of methods like effective negation handling, word n-grams and feature selection by mutual information. Sentiment analysis is a complicated problem but experiments have been done using Naive Bayes, maximum entropy classifiers and support vector machines. Pang et al. found the SVM to be the most accurate classifier. This paper suggested using the Naive Bayes classifier as it is a simple probabilistic model based on the Bayes rule along with a strong independence assumption. We implemented the classifier in Python using hash tables to store the counts of words in their respective classes.

  They have obtained an overall accuracy of 88.80% over a test set of 25000 movie reviews. The running time of our algorithm is O(n + V log V) for training and O(n) for testing, where n is the number of words in the documents (linear) and V the size of the reduced vocabulary. It is much faster than other machine learning algorithms like Maxent classification or Support Vector Machines which take a long time to converge to the optimal set of weights.

- Tan S., Cheng X., Wang Y., Xu H. (2009) [4] Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In: Boughanem M., Berrut C., Mothe J., Soule-Dupuy C. (eds) Advances in Information Retrieval. ECIR 2009. Lecture Notes in Computer Science, vol 5478. Springer, Berlin, Heidelberg.
  In the community of sentiment analysis, supervised learning techniques have been shown to perform very well. When transferred to another domain, however, a supervised sentiment classifier often performs extremely bad. This is a so-called domain-transfer problem. In this work, we attempt to attack this problem by making the maximum use of both the old-domain data and the unlabeled new-domain data. To leverage knowledge from the old-domain data, they proposed an effective measure, i.e., Frequently Co-occurring Entropy (FCE), to pick out generalizable features that occur frequently in both domains and have a similar occurring probability. To gain knowledge from the new domain data, they proposed Adapted Naïve Bayes (ANB), a weighted transfer version of Naive Bayes Classifier. The experimental results indicate that the proposed approach could improve the performance of the base classifier dramatically, and even provide much better performance than the transfer-learning baseline, i.e. the Naïve Bayes Transfer Classifier (NTBC).

  To investigate the effectiveness and robustness of proposed approach, they conducted an extensive experiment on three Chinese domain-specific tasks, including education reviews, stock reviews and computer reviews. The experimental results indicate that proposed approach could improve the performance of base classifier dramatically, and even provide much better performance than the transfer-learning baseline, i.e. the Naïve Bayes Transfer Classifier (NTBC)

  To gain knowledge from the new-domain data, Adapted Naive Bayes (ANB), a weighted transfer version of Naive Bayes Classifier was proposed. ANB employs a weighted Expectation-Maximization (EM) algorithm to train a transfer model using the old-domain data as well as the new domain data. The basic difference from the traditional EM-based classifier is that, with the iteration, ANB gradually expands the weight for the new-domain data while decreases the weight for the old-domain data, and simultaneously utilizes all features for the new-domain data while only uses generalizable features for the old-domain data.

Authors	Name of the Problem	Pros	Cons
Tan S., Cheng X., Wang Y., Xu H.	Advances in Information Retrieval	Supervised Learning technique works very well for a single domain and gives excellent result.	When transferred to another domain, however, a supervised sentiment classifier often performs extremely bad.
Pak, Alexander and Paroubek, Patrick	Twitter as a Corpus for Sentiment Analysis and Opinion Mining	According to this paper, we can use Twitter to get reviews about a particular topic. This boosts the making of the dataset to a certain extent.	But this paper does not focus on how the reviews will be classifier and what type of sentiment analysis is best for what type of reviews.
Narayanan, Vivek	Fast and accurate sentiment classification using an enhanced Naive Bayes model.	Using the Naive Bayes classifier as it is a simple probabilistic model based on the Bayes rule along with a strong independence assumption.	Naive Bayes classifier makes a very strong assumption on the shape of the data distribution.
Ahire, S.G.	A Survey of Sentiment Lexicons.	• Lexicon-based approach uses sentiment lexicons – dictionaries of words with labels specifying their sentiments – to identify the sentiment of the text. • It doesn't faces domain transfer problem.	The limitation of the lexicon-based approach is that it doesn't provide accuracy as the classifier based approach provides.

Table 3.1: Literature Survey Table

Chapter 4

Dataset Preparation

4.1 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. We collected those reviews using web scraping tools as Beautiful Soup (A python library for pulling data out of HTML and XML files). So while we were creating the dataset we used Twitter as a source for reviews and tested different classifiers, but we didn't yet know about the Domain Transfer Problem. 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. So we labeled the reviews we collected as we were going to try some Supervised learning model. 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:

- 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:

- http://people.dbmi.columbia.edu/ friedma/Projects/DiseaseSymptomKB/index.html

4.2 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.

4.3 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.

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

Table 4.1: Doctor Ranking Table

Reviews	Liked
Great experience was very happy with my results it has been 7 weeks and I am already lifted weight lightly. Was very important to me to go back to my activity.Dr.Silver did what he said put me back together.Was in horrible pain for years if I knew it would be like this I would have had it done years ago.Staff was great . So satisfied I am waiting to do my left .Words can not explain how great he was .Thank you Dr. Silver.	1
Dr Silver operated on my shoulder and my knee. He is patient and caring and takes the time to explain things in plain English. He answered my questions and made me feel comfortable about both procedures. I had rotator cuff surgery on my shoulder and meniscus repair on my knee. My treatment was excellent and I highly recommend him	1
This physician refuses to speak to physicians when they want to refer patients. His staff gives you the run around and lies and says that they will have the doctor call their primary care doctor when he has no interest in doing so. He obviously only cares about the money and not patients.	0
unfriendly and inefficient! His staff are not great, too!	0
DO NOT book an appointment with Dr. Rose. His medical skills are great, but his office staff (his family members) mistreat and denigrate their patients, make frequent mistakes, no customer service, dont return phone calls or emails, lie and may attack you. This happened to both me and my mom after being patients there for four years. This is absolutely the most incompetent and unprofessional practice I have ever experienced. I do not recommend him to anyone, seek competent care elsewhere.	0
Dr Morton is a great primary care doctor. Always makes visits a quick and painless process.	1
He is great to his patients and their family members! He is patient, precise, knowledgeable, and very personable.	1
I would highly recommend Dr. Valenti. She is very personable, on time and never seems rushed. She was patient with me on many occasions to find the perfect fit for contacts.	1
Good care. will recommend. have also seen other doctors at this office.	1
Dr. Zellner is an excellent Cardiologist. He is very genuine and truly cares about his patients that he provides care to. He appreciates the staff helping him for his procedures and is a great source of knowledge for the Springfield community.	1
I would NOT recommend seeing Dr. Rosenwasser. He is obviously well qualified and an intelligent person, however his lack of care, respect, and attention are what make him a horrible doctor. If you want someone who will be attentive to your case and will provide solutions, he is NOT the doctor for you. He treated me like absolute garbage and gave me no time of day (treated me like I was a bother). Faculty is rude and had multiple forms incorrectly process on their end.	0
does it all, diagnostics, intervention, great staff and PA, very friendly and easy to talk to.	1

Table 4.2: Doctor Ranking Table

Reviews	Liked
Arrogant, awful, money-grubbing doctor. If you want to read real reviews–and not the ones his staff makes up–visit Yelp.	0
Horrible to understand, didn't even have my file at hand during my appt. I was there for results and answers about a procedure needed to be done, but he comes and he asked my daughter and me what was wrong? when asked about my file, he said he had so many patients he didn't get a chance to get my file, yet I am sitting there and he never asked one of the assistants to get it. It was like he was trying to brush us off. I DO NOT TRUST HIM or recommend.	0
Dr. Zellner is an excellent Cardiologist. He is very genuine and truly cares about his patients that he provides care to. He appreciates the staff helping him for his procedures and is a great source of knowledge for the Springfield community.	1
Dr. Beldner is an excellent hand surgeon. He explains the injury and treatment plans very clearly. He is techically excellent in the operating room. I am a physician and he treated my injury very effectively and got me back to work at full strength. I have full confidence in him and refer all my colleagues and friends with hand and wrist problems to him.	1
Dr. Beldner is great–he's kind, gentle and explains things in an accessible way. He's clearly and expert and I immediately trusted him to make the best decisions for my care. I'd recommend him without reservations to anyone.	1
This physician refuses to speak to physicians when they want to refer patients. His staff gives you the run around and lies and says that they will have the doctor call their primary care doctor when he has no interest in doing so. He obviously only cares about the money and not patients.	0
Please do yourself a favor never go and see Dr El-Gazzar. I was made to wait 120 minutes before his assistant come and took a stock at my problem. 6 mins that's what he took to document my years old problem. Then comes Dr El-Gazzar, a self proclaimed super busy guy, not in mood to understand patient needs and problem. He spent 2 mins, yeah that's right 2 MINS, and went away. Then comes his assistant back again and gave me prescription and left the room. That's it the appointment was over.	0
DO NOT book an appointment with Dr. Rose. His medical skills are great, but his office staff (his family members) mistreat and denigrate their patients, make frequent mistakes, no customer service, dont return phone calls or emails, lie and may attack you. This happened to both me and my mom after being patients there for four years. This is absolutely the most incompetent and unprofessional practice I have ever experienced. I do not recommend him to anyone, seek competent care elsewhere.	0

Table 4.3: Doctor Ranking Table

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 n^th column and 1 denotes the opposite. There are a total of 132 symptoms in our dataset.
- n^th column: This column denotes the disease that can be caused by the given symptoms.There are a total of 40 disease in our dataset.

Itching	skin$_{r}ash$	nodal$_{s}kin_{e}ruptions$	continuous$_{s}neezing$	...	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
1	0	0	0	...	Drug reaction
1	1	0	0	...	A Drug reaction
1	1	0	0	...	A Drug reaction
0	0	0	0	...	GERD
1	0	0	0	...	Jaundice
0	0	0	0	...	Jaundice
1	1	0	0	...	Chicken pox
0	1	0	0	...	Chicken pox
1	0	0	0	...	Chicken pox
0	1	0	0	...	Dengue
0	0	0	0	...	Dengue
1	0	0	0	...	Hepatitis B
0	0	0	0	...	Hepatitis B
0	0	0	1	...	common cold
0	1	0	0	...	acne
0	0	0	0	...	acne
0	1	0	0	...	Psoriasis
0	0	0	0	...	Psoriasis
0	1	0	0	...	Psoriasis
0	0	0	0	...	Pepticulcer diseae

TABLE 4.4: Symptoms to Disease Mapping Table

Chapter 5

Proposed Model

5.1 Personal Health Recommendation System

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

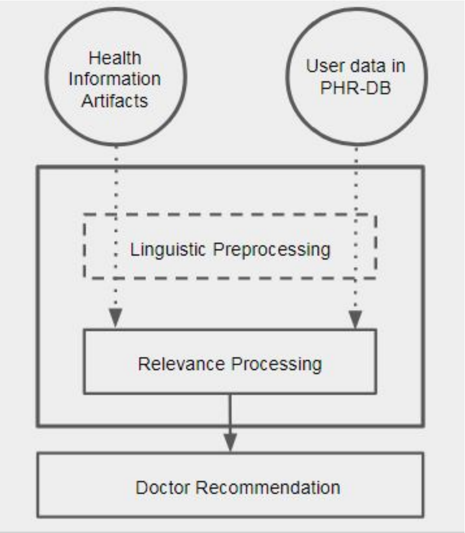


FIGURE 5.1: Personal Health Recommendation System

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

And the inputs are :

- Clinical data
- Review data

And finally, the output of the system is :

- Doctor recommendation

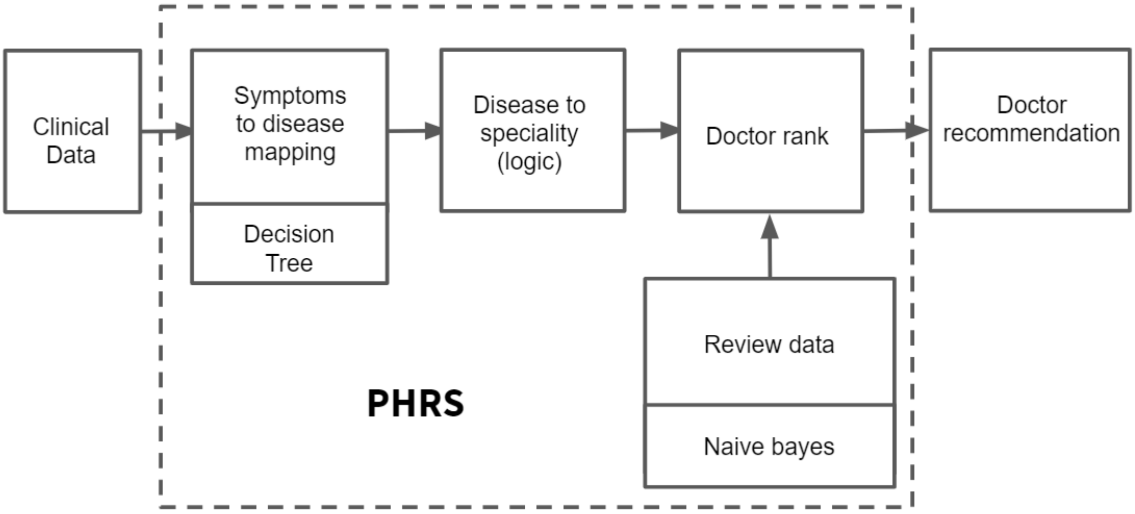


Figure 5.2: Personal Health Recommendation System

The modules of the proposed model are described below :

5.1.1 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.

5.1.2 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. Here, in this case, the user might be suffering from fever as the symptoms point that. And to perform this task Decision Tree is used. 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. The leaves are the decisions of the final outcomes. The decision nodes are where the data is split.

5.1.2.1 Decision Tree

A Decision tree 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 above binary tree. Let's say we want to predict whether a person is fit given their information like age, eating habit, and physical activity.

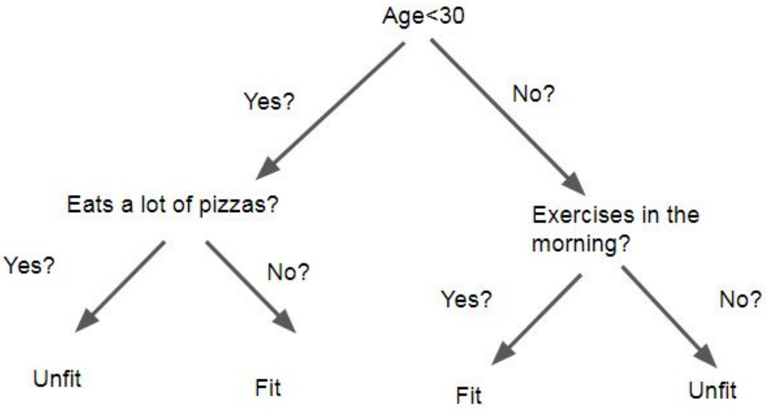


FIGURE 5.3: An Example Of Decision Tree

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. In this case this was a binary classification problem (a yes or no type problem).

5.1.2.2 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**

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

$$H(S) = \sum_{x \in X} P(x) log_2 \frac{1}{P(x)} \tag{5.1}$$

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. Here the entropy is the highest possible since there's no way of determining what the outcome might be. 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. In other words, this event has no randomness hence it's entropy is zero.

- **Infornation Gain**

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.

$$IG(S, A) = H(S) - \sum_{i=0}^{n} P(x) \times H(x) \tag{5.2}$$

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.

5.1.2.3 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 5.1: 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'll go ahead and grow the decision tree. The initial step is to calculate H(S), the Entropy of the current state.
We know,

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

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

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

$Entropy(S) = -(\frac{4}{7}) log_2 \frac{4}{7} - (\frac{3}{7}) log_2 \frac{3}{7} = 0.985$

Next, step is to find the highest possible Information gain which we will choose as root node. Lets start with Itching We have to find out information gain,

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

For that we have to find

1. $H(S_1)$
2. $H(S_0)$
3. $P(S_1)$
4. $P(S_0)$
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:

$P(S_1) = \frac{Itching_1}{Total}$
$P(S_1) = \frac{2}{7} = 0.285$

$P(S_0) = \frac{Itching_0}{Total}$
$P(S_0) = \frac{5}{7} = 0.71$

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

Since,we have no randomness,

$Entropy(S_1) = 0$

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

$Entropy(S_0) = -(\frac{2}{5})log_2\frac{2}{5} - (\frac{3}{5})log_2\frac{3}{5} = 0.97$

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

$IG(S, Itching) = H(S) - P(S_0)H(S_0) - P(S_1)H(S_1)$ $IG(S, Itching) = 0.985 - 0.71 * 0.97 - P(S_1) * 0 = 0.29$

Similarly, for **Skin Rash**

Among all 7 examples we can see we have 3 places where Skinrash is 1 and 4 places where Skinrash is 0,
$P(S_1) = \frac{Skinrash_1}{Total}$
$P(S_1) = \frac{3}{7} = 0.42$
$P(S_0) = \frac{Skinrash_0}{Total}$
$P(S_0) = \frac{4}{7} = 0.57$

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

Since,we have no randomness,
$Entropy(S_1) = 0$

but, for itching=0 we have 1 cases of allergy and 3 cases of fever,
$Entropy(S_0) = -(\frac{1}{4})log_2\frac{1}{4} - (\frac{3}{4})log_2\frac{3}{4} = 0.81$

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

$IG(S, Skinrash) = H(S) - P(S_0)H(S_0) - P(S_1)H(S_1)$
$IG(S, Skinrash) = 0.985 - 0.57 * 0.81 - P(S_1) * 0 = 0.52$

Similarly, for Shivering

Among all 7 examples we can see we have 2 places where Shivering is 1 and 5 places where Shivering is 0,

$P(S_1) = \frac{Shivering_1}{Total}$
$P(S_1) = \frac{2}{7} = 0.28$

$P(S_0) = \frac{Shivering_0}{Total}$
$P(S_0) = \frac{4}{7} = 0.57$

Now we have 2 examples where Jointpain = 1 we have 2 cases of **fever** and 1 case of allergy So, we have,

therefore, $Entropy(S_1) = -(\frac{2}{3})log_2\frac{2}{3} - (\frac{1}{3})log_2\frac{1}{3} = 0.91$

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

$Entropy(S_0) = -(\frac{3}{4})log_2\frac{3}{4} - (\frac{1}{4})log_2\frac{1}{4} = 0.81$

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

$IG(S, Jointpain) = H(S) - P(S_0)H(S_0) - P(S_1)H(S_1)$
$IG(S, Jointpain) = 0.985 - 0.42 * 0.91 - 0.57 * 0.81 = 0.14$

So, we can see

1. $IG(S, Itching) = 0.29$
2. $IG(S, Skinrash) = 0.52$
3. $IG(S, Shivering) = 0.47$
4. $IG(S, Jointpain) = 0.14$

Since Skinrash has the highest Information Gain we will use Skinrash as the root node.

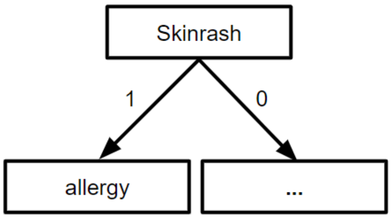


Figure 5.4: Decision Tree Phase One

Now, since skinrash 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,

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

Table 5.2: Symptoms to disease data after phase 1

For the following table Skinrash=1, we have,

$Entropy(S) = -(\frac{3}{4})log_2\frac{3}{4} - (\frac{1}{4})log_2\frac{1}{4} = 0.81$

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

1. $H(S_1)$
2. $H(S_0)$
3. $P(S_1)$
4. $P(S_0)$
5. $H(S)$=0.81 which we calculated above

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

$P(S_1) = \frac{Itching_1}{Total}$
$P(S_1) = \frac{1}{4} = 0.25$

$P(S_0) = \frac{Itching_0}{Total}$
$P(S_0) = \frac{3}{4} = 0.75$

Now we have 1 examples where **Itching** = 1, we have result fever
Since,we have no randomness,

$Entropy(S_1) = 0$

but, for itching=0 we have 3 cases of fever,

$Entropy(S_0) = 0$

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

$IG(S, Itching) = H(S) - P(S_0)H(S_0) - P(S_1)H(S_1)$
$IG(S, Itching) = 0$

Similarly, for Shivering
Among all 4 examples we can see we have 2 places where Shivering is 1 and 2 places where Shivering is 0,

$P(S_1) = \frac{Shivering_1}{Total}$
$P(S_1) = \frac{2}{4} = 0.50$

$P(S_0) = \frac{Shivering_0}{Total}$
$P(S_0) = \frac{2}{4} = 0.50$

Now we have 1 examples where Shivering = 1, we have result fever

Since,we have no randomness,

$Entropy(S_1) = 0$

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

$Entropy(S_0) = -(\frac{1}{2})log_2\frac{1}{2} - (\frac{1}{2})log_2\frac{1}{2} = 0.5$

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

$IG(S, Shivering) = H(S) - P(S_0)H(S_0) - P(S_1)H(S_1)$
$IG(S, Shivering) = 0.81 - 0.5 * 0.5 - 0$ =0.56

Similarly, for Jointpain
Among all 4 examples we can see we have 2 places where Jointpain is 1 and 2 places where jointpain is 0,

$P(S_1) = \frac{jointpain_1}{Total}$
$P(S_1) = \frac{2}{4} = 0.50$

$P(S_0) = \frac{jointpain_0}{Total}$
$P(S_0) = \frac{2}{4} = 0.50$

Now we have 1 examples where jointpain = 1, we have result fever.
Since,we have no randomness,

$Entropy(S_1) = 0$

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

$Entropy(S_0) = -(\frac{1}{2})log_2\frac{1}{2} - (\frac{1}{2})log_2\frac{1}{2} = 0.5$

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

$IG(S, Jointpain) = H(S) - P(S_0)H(S_0) - P(S_1)H(S_1)$
$IG(S, jointpain) = 0.81 - 0.5 * 0.5 - 0 = 0.56$

So, we can see

$IG(S, Itching) = 0$
$IG(S, Shivering) = 0.56$
$IS(S, JointPain) =).56$

Since Shivering and Jointpain have same Information Gain we can take any one of them, In our case lets make Shivering as next node.

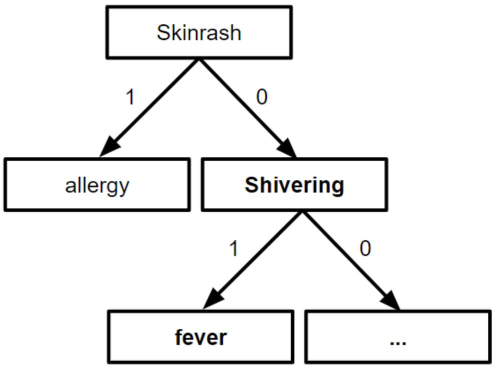


Figure 5.5: Decision Tree Phase two

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

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

Table 5.3: Symptoms to disease data after phase two

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 graph is given below,

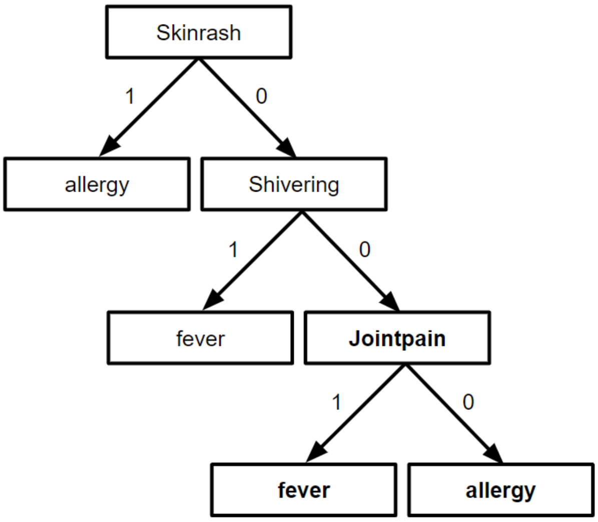


Figure 5.6: Final Decision Tree

5.1.3 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 specialty for the specified disease provided by the user. For this purpose we have made a MySQL table with two columns:

- Disease
- 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 specialtiy.

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).

5.1.4 Doctor ranking

5.1.4.1 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 preprocessing 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.

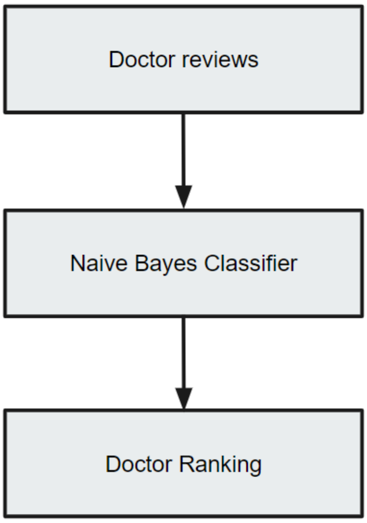


Figure 5.7: Ranking

- Tokenization
- Cleaning the Data
- Removing Stopwords

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! :

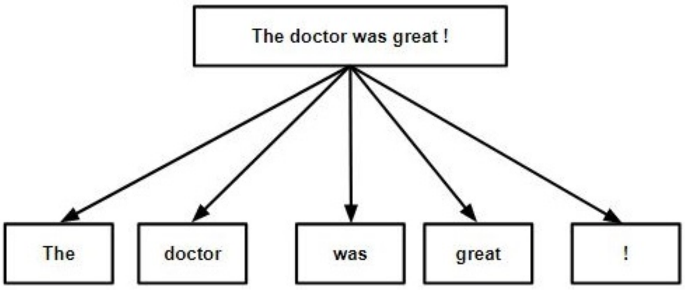


Figure 5.8: Tokenization

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

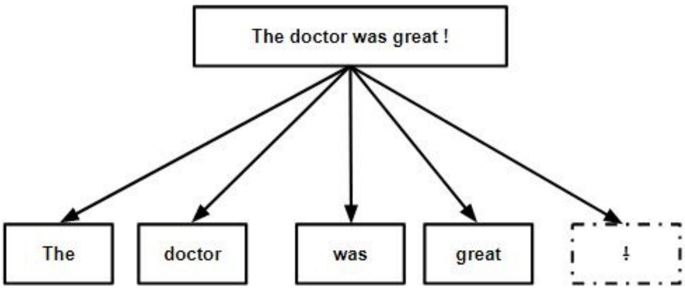


Figure 5.9: Cleaning the Data

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. Though stopwords usually refer to the most common words in a language. For some search engines, there are some of the most common, short function words, such as the, is, at, which, and so on. The StopWords are then removed and the processed data is ready for classification. Example:

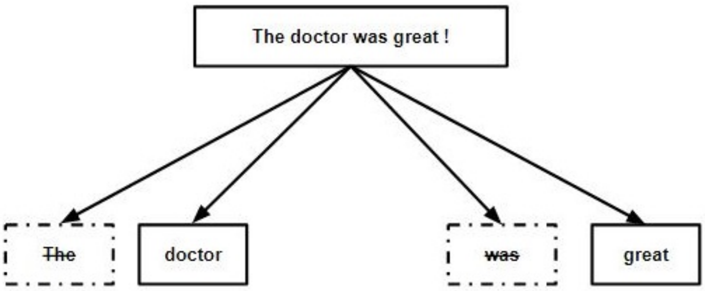


Figure 5.10: Removing Stop Words

5.1.4.2 Naive Bayes Classifier

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. In spite of the great advances of the Machine Learning in the last years, it has proven to not only be simple but also fast, accurate and reliable. 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. In other words, the conditional probabilities are inverted so that the query can be expressed as a function of Chapter 5 16 measurable quantities. The approach is simple and the adjective naive has been attributed not because these algorithms are limited or less efficient, but because of a fundamental assumption about the causal factors that we will discuss. Naive Bayes is multi-purpose classifiers and its easy to find their application in many different contexts. However, the performance is particularly good in all those situations when the probability of a class is determined by the probabilities of some causal factors. 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. The Naive Bayes equation is given below:

$$p(y|x_1, x_2, x_3...x_m) = \alpha p(y) \prod p(x_i|y) \tag{5.3}$$

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:

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

Table 5.4: Review containing stopwords

After 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

Table 5.5: Removing Stop Words

Now, which tag does the sentence "A great doctor and friendly one" belong to? Since Naive 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**

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 (like we are doing here), because it provides us with a way to reverse them:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{5.4}$$

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

$$P(positive|a\ great\ doctor\ and\ friendly\ one)$$

$$= \frac{P(a\ great\ doctor\ and\ friendly\ one|positive) \times P(positive)}{P(a\ great\ doctor\ and\ a\ friendly\ one)}$$

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

$$P(a\ great\ doctor\ and\ friendly\ one|Positive) \times P(Positive)$$

With

$$P(a\ great\ doctor\ and\ friendly\ one|Negative) \times p(Negative)$$

- **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:

$$P(Agreatdoctorandfriendlytoeveryone) = P(a) \times P(great) \times P(doctor) \times P(and) \times P(friendly) \times P(one)$$

$P(A\ great\ doctor\ and\ friendly\ to\ everyone|positive) = P(a|positive) \times P(great|positive) \times P(doctor|positive) \times P(and|positive) \times P(friendly|positive) \times P(one|positive)$

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

$P(positive)\ is\ \frac{3}{5}$
$P(negative)\ is\ \frac{2}{5}$

CALCULATING FINAL PROBABILITY

Words	P(Word—Positive)	p(Word—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	$(0+1)/(6+10) = 0.06$	$(0+1)/(4+10) = 0.07$

Table 5.6: Calculating Probability for first sentense

$P(great|positive) \times P(doctor|positive) \times P(friendly|positive) \times P(one|positive) = 0.000155$
$P(great|negative) \times P(doctor|negative) \times P(friendly|negative) \times P(one/negative) = 0.000048$

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

Words	P(Word—Positive)	p(Word—Negative)
tons	$(0+1)/(6+10) = 0.06$	$(0+1)/(4+10) = 0.07$
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	$(0+1)/(6+10) = 0.06$	$(1+1)/(4+10) = 0.14$

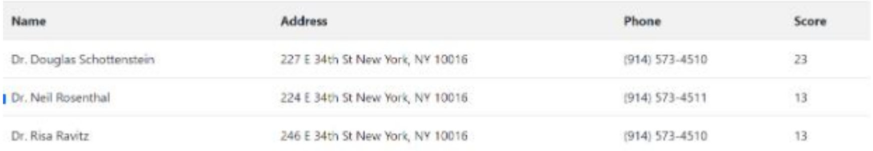
Table 5.7: Calculating probability for second sentence

$P(tons|positive) \times P(meds|positive) \times P(worst|positive) \times P(treatment|positive) = 0.0000129$
$P(tons|negative) \times P(meds|negative) \times P(worst|negative) \times P(treatment/negative) = 0.00019$

Hence, our classifier gives a result of Negative.

5.1.4.3 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 as shown in the figure below.



Name	Address	Phone	Score
Dr. Douglas Schottenstein	227 E 34th St New York, NY 10016	(914) 573-4510	23
Dr. Neil Rosenthal	224 E 34th St New York, NY 10016	(914) 573-4511	13
Dr. Risa Ravitz	246 E 34th St New York, NY 10016	(914) 573-4510	13

Figure 5.11: Recommended Doctors

Chapter 6

Results and Discussions

6.1 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. We also tried to train a CNN for this purpose but was not successful using it since it took a bit of time and our requirement was to provide the solution in real-time.

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. After the symptoms had been mapped to a specific speciality has to be selected accordingly according to logic.

On the other hand, the Health Information Artifacts (Reviews provided by the patients) is the input to the linguistic processing module. The reviews are to be classified into positive and negative which is done by the Naive Bayes classifier. But we found that doctors don't have many negative reviews since doctors are respected highly in society. So the patients choose to not give any reviews. And the doctors who have good success had a lot of positive reviews. 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.

6.2 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%. The final output of the whole system is as given below.

6.2.1 System Interface

Whenever we open the home of the symptoms to disease prediction we are greeted with the following page:

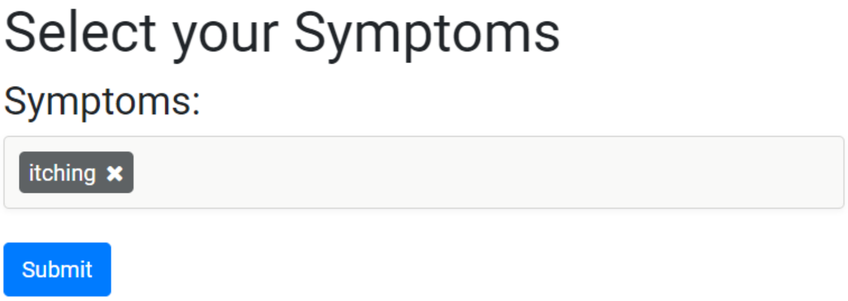


Figure 6.1: Symptoms Selection

The first step is to choose the symptoms of the illness user is suffering from. Our system also suggests the symptoms which the users can choose easily.

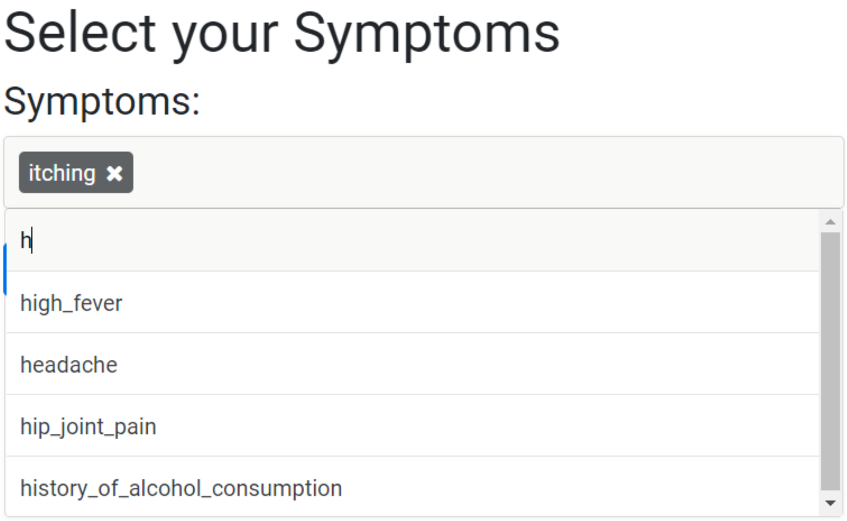
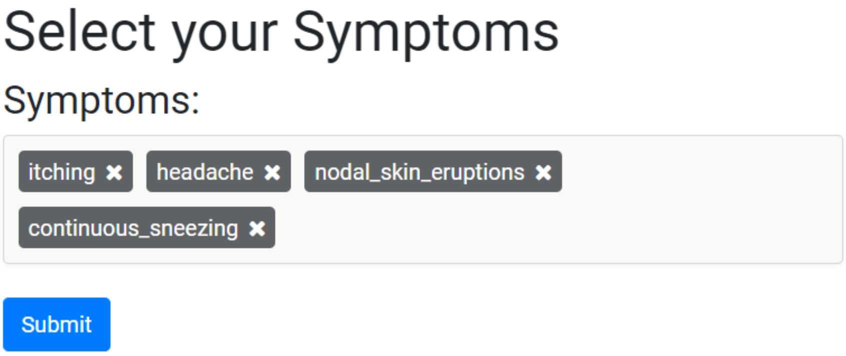


Figure 6.2: Symptoms Suggestions

The most important feature is that users can select multiple symptoms and search for the disease they might be suffering from. The number of symptoms is directly proportional to the accurate disease.



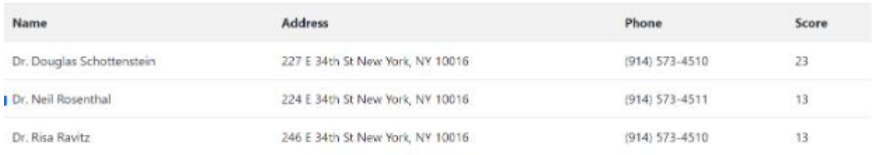
Select your Symptoms

Symptoms:

itching ✖ headache ✖ nodal_skin_eruptions ✖ continuous_sneezing ✖

FIGURE 6.3: Multiple Symptoms Selection

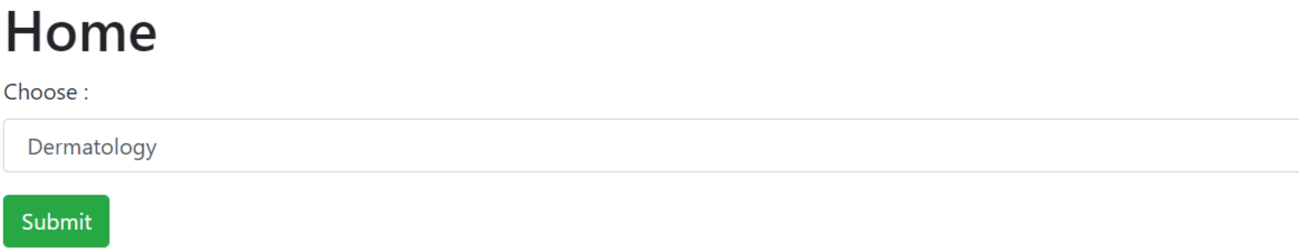
When the symptoms are submitted the final output is Recommended doctors.The figure below suggests best dermatologists which have been shown from the rank table that is made using the positive reviews provided by our classifier. The final outcome of this system is to provide the best doctors to the users.



Name	Address	Phone	Score
Dr. Douglas Schottenstein	227 E 34th St New York, NY 10016	(914) 573-4510	23
Dr. Neil Rosenthal	224 E 34th St New York, NY 10016	(914) 573-4511	13
Dr. Risa Ravitz	246 E 34th St New York, NY 10016	(914) 573-4510	13

FIGURE 6.4: Recommended doctors

Another way of finding the recommended doctors is by selecting speciality of doctor.



Home

Choose : Dermatology

FIGURE 6.5: Recommendation of doctors from speciality

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. From that the system makes a rank table which is queried whenever a recomendation process starts. The goal of making a rank table is that we dont have to run the classifier over and over everytime we need to recommend a doctor, the system just searches a doctor from the rank table which is way faster.

ID	Name	Number of Positive Reviews	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
230	Dr. Paul Phillips	28	35.44303797
252	Dr. Roshan Shah	27	34.17721519
50	DR. CHRISTIAN ZELLNER	26	32.91139241
129	Dr. Howard Rose	26	32.91139241
135	DR. ISH SINGLA	26	32.91139241
168	Dr. Larry Sapetti	26	32.91139241
301	Dr. Yaser El-Gazzar	25	31.64556962
4	Dr Claudine T Gillison	24	30.37974684
251	Dr. Ronald Lehman Jr	24	30.37974684
299	Dr. William Severino	24	30.37974684
8	Dr. Adam Shoman	22	27.84810127
82	Dr. Douglas Schottenstein	22	27.84810127
165	Dr. Kevin Schlee	22	27.84810127
214	Dr. Navjot Ghotra	22	27.84810127
207	Dr. N. Patrick Hennessey	21	26.58227848
16	Dr. Allan Ho	15	18.98734177
67	Dr. David Fields	15	18.98734177
88	Dr. Edwin Su	15	18.98734177
250	Dr. Robin Valenti	15	18.98734177

Table 6.1: Top Doctors with their respective scores from our rank table

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.

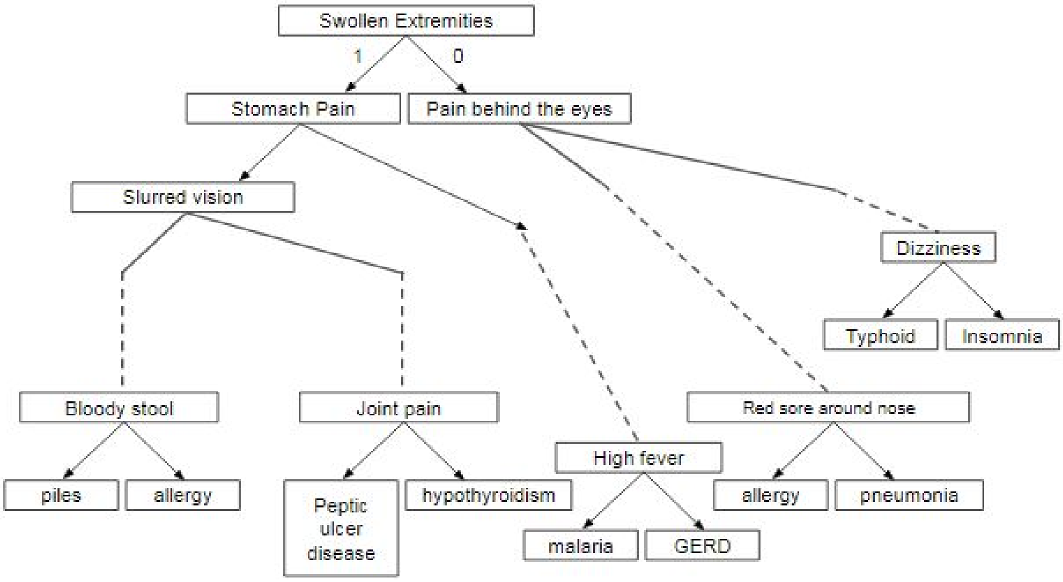


Figure 6.6: A snippet of the result Decision Tree

6.3 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. So the lexicon-based approach is a approach for the sentiment analysis. The major advantage of this approach is that since the annotation is performed by humans, correctness is guaranteed barring an actual error in the annotation. 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. However, the problem with this approach is the immense investment of time.

Other than the Naive Bayes classifier we have used Support Vector Machine initially but the result was not satisfactory. The accuracy of SVM was 82% and the data fitting time was 0.28125ms. The limitation of SVM in our model was that the accuracy was low as compared to the Naive Bayes classifier because it works good with dataset with less features and for our model the features are all the distinct words except the stopwords 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.

Chapter 7

Conclusion And Future Scope

7.1 Conclusion

We have successfully 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 which are connected to each other using a bridge of python logic and the required output is top doctors based on the users input.

7.2 Future Scope

An account system is to be implemented in our system where patient information can be saved which can improve our Doctor Recommendation System to a great extent. Patients can log in to see their past appointments with doctors they have visited and can comment using their name or anonymously.