Disease Detection based on Symptoms

Nikunj Agarwal
M.Tech(CSE)
IIIT Delhi
MT19093
nikunj19093@iiitd.ac.in

Rahul Maheshwari
M.Tech(CSE)
IIIT Delhi
MT19027
rahul19027@iiitd.ac.in

Anand Sharma
M.Tech(CSE)
IIIT Delhi
MT19059
anand19059@iiitd.ac.in

1 Problem Definition

The paper proposes a model in which the user can input unstructured symptoms or select the symptoms suggested by the system, based on which, a list of probable diseases is provided back to the user. Further, the user can select any of the output diseases to get more information about its other symptoms, causes, diagnosis, possible treatment, etc. to help the user better understand the disease and current medical condition. The system also suggests other symptoms based on the ones that the user has input.

The system can be used by a person with restricted medical knowledge as well with ease and can come handy in early disease detection and diagnosis. It can also benefit users that are reluctant to visit hospitals on the onset of minor symptoms. This will provide them with a basic idea of the severity of the disease.

2 Background

Machine Learning applications in healthcare and biomedical domain has lead to early disease detection and better diagnosis. This has enhanced patient care in recent times. Studies have shown that people take the help of the internet for any possible health-related issues. The problem with this approach is that the search engines provide bulk information in scattered format from which it is difficult to conclude.

There are many disease prediction systems available such as heart disease prediction, neurological disorders prediction, and skin disease prediction. But universal prediction system for diseases based on symptoms is rarely in practice. It is very helpful for doctors or medical experts to diagnose diseases at an early stage based on symptoms. When a query is given, probable diseases are suggested to the user based on the highest probability and scores.

With the use of the internet and all resources available to the user, proper diseases are used, and based on that proper medication is done which is very beneficial to all human beings. It is very helpful for doctors and patients to know better about the disease without any medical tests or anything else.

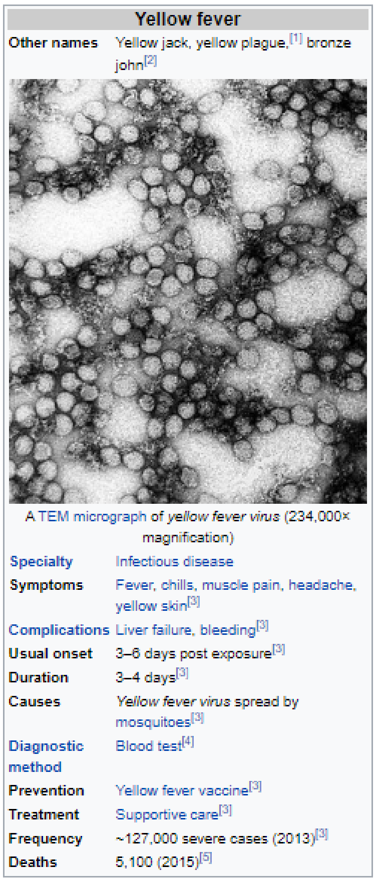
The detection of disease based on disease is a complex game. Being unfamiliar with biological terms, the users feed the symptoms in non-technical or natural terms which add complexity in predicting diseases. The main objective is to develop a novel architecture that could accept and handle such type of user queries by employing techniques like query expansion using a thesaurus, synonym matching, and symptom suggestion that will allow disease prediction with greater accuracy based on user input. We have scraped data from the web and generated dataset which can be used in future research. Query search retrieval and matching are used in such problems to achieve prediction.

3 Dataset Used

The previously available dataset is restricted to a particular part of human body disease and is also smaller in volume. Hence, the dataset of disease and their symptoms has been scraped from the web by running the Python script.

The dataset consists of diseases and their symptoms, which are fetched from the following sources:

- Diseases: The list of diseases has been retrieved from the National Health Portal of India (`https://www.nhp.gov.in/disease-a-z`), developed and maintained by Centre for Health Informatics (CHI). The script fetches the HTML code of the page and extracts the disease list by filtering values in HTML tags.



Yellow fever	
Other names	Yellow jack, yellow plague,[1] bronze john[2]
A TEM micrograph of *yellow fever virus* (234,000× magnification)	
Specialty	Infectious disease
Symptoms	Fever, chills, muscle pain, headache, yellow skin[3]
Complications	Liver failure, bleeding[3]
Usual onset	3–6 days post exposure[3]
Duration	3–4 days[3]
Causes	*Yellow fever virus* spread by mosquitoes[3]
Diagnostic method	Blood test[4]
Prevention	Yellow fever vaccine[3]
Treatment	Supportive care[3]
Frequency	~127,000 severe cases (2013)[3]
Deaths	5,100 (2015)[5]

Figure 1: Wikipedia's Infobox for Yellow Fever

- Symptoms: The script uses the Google Search package to perform searching and fetch the disease's Wikipedia page among the various search results obtained. The HTML code of the page is processed to fetch the symptoms of the disease using the 'infobox' available on the Wikipedia page. Figure 1 shows an example of Wikipedia's infobox.

All the symptoms are extracted and a dictionary is created with key as disease and symptoms as value. Further, each disease is treated as the label and all symptoms are treated as specific attributes or columns. Figure 2 shows the systematic flow of steps involved in data scraping.

The scraping script fetches over 261 different diseases that form the label and 500+ symptoms. The symptoms are then pre-processed to remove similar symptoms with different names (For example, headache and pain in the forehead). This is done by finding the synonyms for each symptom and computing Jaccard Coefficient for pairs of symptoms. If the score is greater than the threshold, both the symptoms are very similar and one of them can be removed.

```
if Jaccard(Symptom1,Symptom2)>threshold:
    Symptom2->Symptom1
```

To multiply the dataset, each disease's symptoms are picked up, combinations of the symptoms are created and added as new rows in the dataset. For example, a disease A, having 5 symptoms, now has a total of $(2^5 - 1)$ entries in the dataset. The dataset, after preprocessing and multiplication, contains around 8835 rows with 489 unique symptoms.

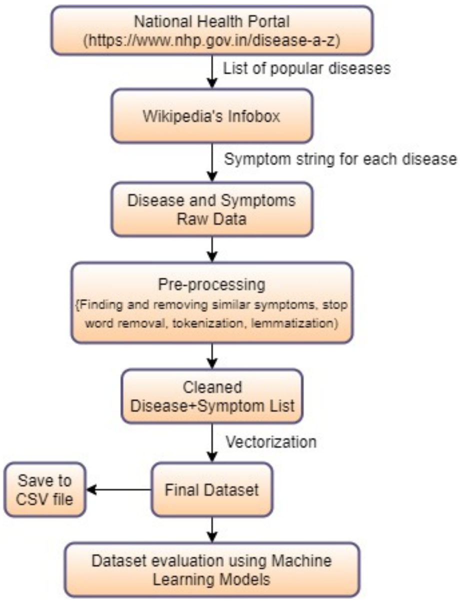

National Health Portal (https://www.nhp.gov.in/disease-a-z)
List of popular diseases
Wikipedia's Infobox
Symptom string for each disease
Disease and Symptoms Raw Data
Pre-processing (Finding and removing similar symptoms, stop word removal, tokenization, lemmatization)
Cleaned Disease+Symptom List
Vectorization
Save to CSV file
Final Dataset
Dataset evaluation using Machine Learning Models

Figure 2: Dataset Scraping and Evaluation

4 Proposed Solution Sketch

On a general note, the system prompts the user to enter symptoms based on which model predicts diseases with the highest probability and scores. Figure 3 describes the process of disease prediction from user input symptoms. The following subsections discuss each module in detail.

4.1 Symptom Preprocessing

The system accepts symptom(s) in a single line, separated by comma(,). Subsequently, the following preprocessing steps are involved:

- Split symptoms into a list based on comma
- Convert the symptoms into lowercase
- Removal of stopwords
- Tokenization of symptoms to remove any punctuation marks

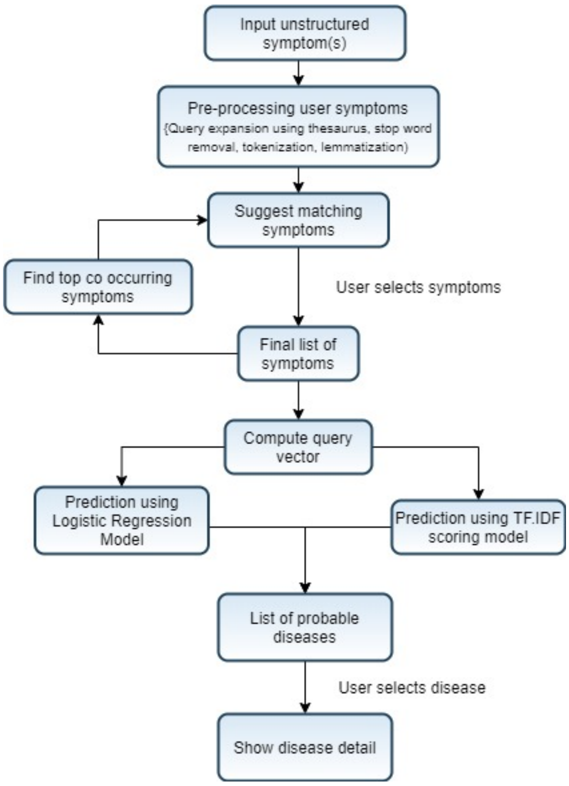
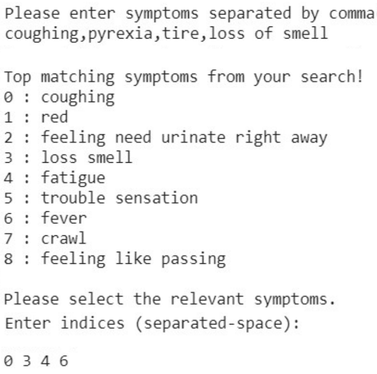

Input unstructured symptom(s)
Pre-processing user symptoms (Query expansion using thesaurus, stop word removal, tokenization, lemmatization)
Suggest matching symptoms
Find top co occurring symptoms
User selects symptoms
Final list of symptoms
Compute query vector
Prediction using Logistic Regression Model
Prediction using TF.IDF scoring model
List of probable diseases
User selects disease
Show disease detail

Figure 3: System Architecture

- Lemmatization of tokens in the symptoms

The processed symptom list is then used for symptom expansion.



```
Please enter symptoms separated by comma(,):
coughing,pyrexia,tire,loss of smell

Top matching symptoms from your search!
0 : coughing
1 : red
2 : feeling need urinate right away
3 : loss smell
4 : fatigue
5 : trouble sensation
6 : fever
7 : crawl
8 : feeling like passing

Please select the relevant symptoms.
Enter indices (separated-space):

0 3 4 6
```

Figure 4: Symptom suggested to user

4.2 Symptom Expansion using Synonyms

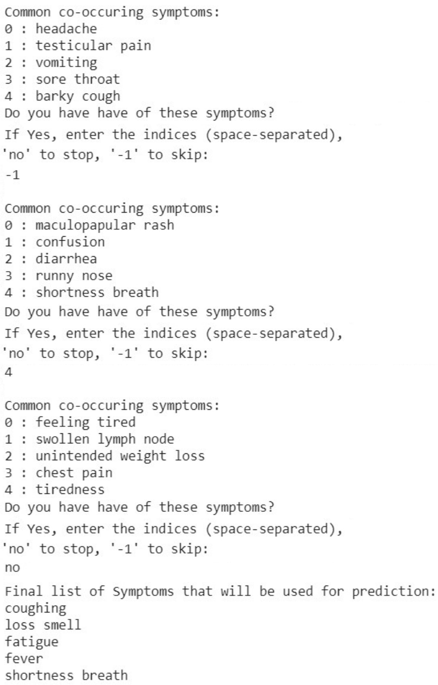
Each symptom is expanded by appending a list of its synonyms. The synonyms are taken from thesauras.com (`https://www.thesaurus.com/`) and Princeton University's WordNET (`https://wordnet.princeton.edu/`) available in Python. Each symptom is broken into its combinations for finding the synonyms set. Figure 4 shows user input symptoms and symptoms found in the dataset that matches the synonym string.

4.3 Symptoms Suggestion and Selection

The expanded symptom query is used to find the related symptoms in the dataset. To find such symptoms, each symptom from the dataset is split into tokens and each token is checked for its presence in the expanded query. Based on this, a similarity score is calculated and if the symptom's score is more than the threshold value, that symptom qualifies for being similar to the user's symptom and is suggested to the user.

```
tokenA->tokens(Symptom A)
tokenSyn->tokens(synonym string)
matching->intersect(tokenA,tokenSyn)
score->count(matching)/count(tokenA)
if score>threshold:
    select Symptom A
```

The user selects one or more symptoms from the list. Based on the selected symptoms, other symptoms are shown to the user for selection which are among the top co-occurring symptoms with the ones selected by the user initially. The user can select any symptom, skip, or stop the symptom selection process. Figure 5 shows an example of the symptom suggestion and selection process. The final list of symptoms is then obtained for computing symptom vector which is used for prediction.



```
Common co-occuring symptoms:
0 : headache
1 : testicular pain
2 : vomiting
3 : sore throat
4 : barky cough
Do you have have of these symptoms?
If Yes, enter the indices (space-separated),
'no' to stop, '-1' to skip:
-1

Common co-occuring symptoms:
0 : maculopapular rash
1 : confusion
2 : diarrhea
3 : runny nose
4 : shortness breath
Do you have have of these symptoms?
If Yes, enter the indices (space-separated),
'no' to stop, '-1' to skip:
4

Common co-occuring symptoms:
0 : feeling tired
1 : swollen lymph node
2 : unintended weight loss
3 : chest pain
4 : tiredness
Do you have have of these symptoms?
If Yes, enter the indices (space-separated),
'no' to stop, '-1' to skip:
no
Final list of Symptoms that will be used for prediction:
coughing
loss smell
fatigue
fever
shortness breath
```

Figure 5: Symptom suggestion and selection process

4.4 Disease Prediction

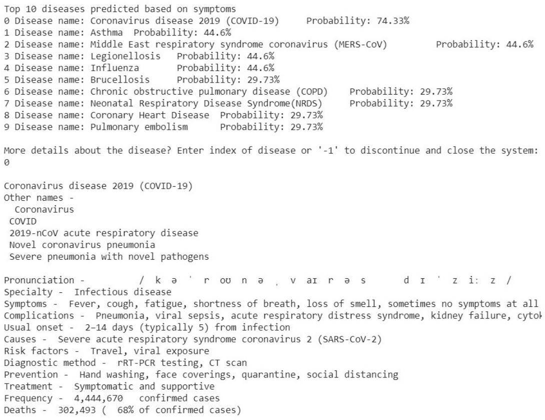
Using the final symptom list, vectors are computed specific to the model and disease prediction is done.

4.4.1 Prediction using Machine Learning Model

A binary vector is computed that consists of 1 for the symptoms present in the user's selection list and 0 otherwise. A machine learning model is trained on the dataset, which is used here for prediction. The model accepts the symptom vector and outputs a list of top K diseases, sorted in the decreasing order of individual probabilities. The probability of a disease is calculated as below.

```
ModelAccuracy->accuracy(model used)
DisASymp=Symptoms(DiseaseA)
match->intersect(DisASymp,userSymp)
matchScore->match/count(userSymp)
prob(DiseaseA)=matchScore*modelAccuracy
```

Figure 6 shows the predicted list of diseases with a probability that is output by Logistic Regressor.



```
Top 10 diseases predicted based on symptoms
0 Disease name: Coronavirus disease 2019 (COVID-19)     Probability: 74.33%
1 Disease name: Asthma  Probability: 44.6%
2 Disease name: Middle East respiratory syndrome coronavirus (MERS-CoV)         Probability: 44.6%
3 Disease name: Legionellosis   Probability: 44.6%
4 Disease name: Influenza       Probability: 44.6%
5 Disease name: Brucellosis     Probability: 29.73%
6 Disease name: Chronic obstructive pulmonary disease (COPD)    Probability: 29.73%
7 Disease name: Neonatal Respiratory Disease Syndrome(NRDS)     Probability: 29.73%
8 Disease name: Coronary Heart Disease  Probability: 29.73%
9 Disease name: Pulmonary embolism      Probability: 29.73%

More details about the disease? Enter index of disease or '-1' to discontinue and close the system:
0

Coronavirus disease 2019 (COVID-19)
Other names -
  Coronavirus
 COVID
 2019-nCoV acute respiratory disease
 Novel coronavirus pneumonia
 Severe pneumonia with novel pathogens

Pronunciation -         / k ə ˈ r oʊ n ə ˌ v aɪ r ə s      d ɪ ˈ z iː z /
Specialty -  Infectious disease
Symptoms -  Fever, cough, fatigue, shortness of breath, loss of smell, sometimes no symptoms at all
Complications -  Pneumonia, viral sepsis, acute respiratory distress syndrome, kidney failure, cytok
Usual onset -  2–14 days (typically 5) from infection
Causes -  Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)
Risk factors -  Travel, viral exposure
Diagnostic method -  rRT-PCR testing, CT scan
Prevention -  Hand washing, face coverings, quarantine, social distancing
Treatment -  Symptomatic and supportive
Frequency -  4,444,670   confirmed cases
Deaths -  302,493 (  68% of confirmed cases)
```

Figure 6: Prediction using Logistic Regression along with disease detail

4.4.2 Prediction using TF.IDF and Cosine Similarity Model

TF.IDF scoring model is trained by computing TF and IDF of the symptoms in the dataset.

- TF (term frequency) is the count of occurrence of the symptom in the disease
- DF (document frequency) is the count of the symptom across the diseases. Inverse DF is computed as shown in equation 1.

$$IDF = log10(\frac{count(AllDiseases)}{DF}) \quad (1)$$

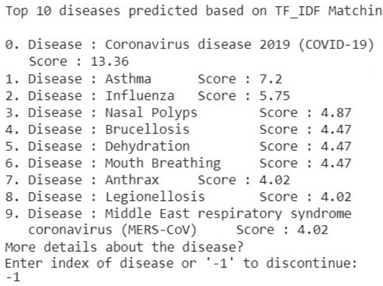
TF.IDF is computed as shown in equation 2. Each vector element is computed using the same formula.

$$TF.IDF(sym, dis) = log10(1 + TF)) * IDF \quad (2)$$

Similarly, the symptom vector for user symptoms is also computed. Both the vector are multiplied to obtain TF.IDF score for user symptom query and disease as shown in equation 3. A higher score means a high similarity between the two vectors.

$$TF.IDFScore(Q, A) = dot(Q, A) \quad (3)$$

The scores are sorted based on a decreasing score and a list of top K diseases is obtained. Figure 7 shows the predicted list of diseases with TF.IDF score.



```
Top 10 diseases predicted based on TF_IDF Matching :

0. Disease : Coronavirus disease 2019 (COVID-19)
   Score : 13.36
1. Disease : Asthma      Score : 7.2
2. Disease : Influenza   Score : 5.75
3. Disease : Nasal Polyps        Score : 4.87
4. Disease : Brucellosis         Score : 4.47
5. Disease : Dehydration         Score : 4.47
6. Disease : Mouth Breathing     Score : 4.47
7. Disease : Anthrax     Score : 4.02
8. Disease : Legionellosis       Score : 4.02
9. Disease : Middle East respiratory syndrome
   coronavirus (MERS-CoV)     Score : 4.02
More details about the disease?
Enter index of disease or '-1' to discontinue:
-1
```

Figure 7: Prediction using TF.IDF scoring model

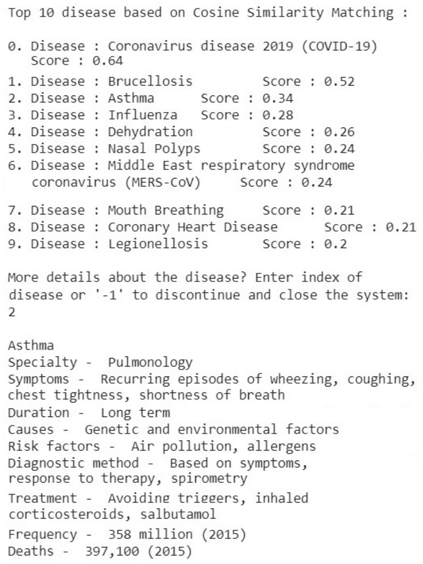
Using the TF.IDF vectors, cosine similarity of the disease, and user symptom vector are computed as shown in equation 4.

$$cos.sim(Q, A) = \frac{dot(Q, A)}{|Q|x|A|} \quad (4)$$

Higher cosine similarity represents a higher similarity between the disease and the query vector. The scores are sorted based on a decreasing score and a list of top K diseases is obtained. Figure 8 shows the predicted list of diseases with a cosine similarity score.

4.4.3 Disease Detail

The user can select any of the disease output by the model and view the details of that disease in the console. Figure 6 and 8 show the disease details as selected by user. Users can skip the step by entering '-1' as shown in Figure 7.



```
Top 10 disease based on Cosine Similarity Matching :

0. Disease : Coronavirus disease 2019 (COVID-19)
   Score : 0.64
1. Disease : Brucellosis         Score : 0.52
2. Disease : Asthma      Score : 0.34
3. Disease : Influenza   Score : 0.28
4. Disease : Dehydration         Score : 0.26
5. Disease : Nasal Polyps        Score : 0.24
6. Disease : Middle East respiratory syndrome
   coronavirus (MERS-CoV)     Score : 0.24

7. Disease : Mouth Breathing     Score : 0.21
8. Disease : Coronary Heart Disease      Score : 0.21
9. Disease : Legionellosis       Score : 0.2

More details about the disease? Enter index of
disease or '-1' to discontinue and close the system:
2

Asthma
Specialty -  Pulmonology
Symptoms -  Recurring episodes of wheezing, coughing,
chest tightness, shortness of breath
Duration -  Long term
Causes -  Genetic and environmental factors
Risk factors -  Air pollution, allergens
Diagnostic method -  Based on symptoms,
response to therapy, spirometry
Treatment -  Avoiding triggers, inhaled
corticosteroids, salbutamol
Frequency -  358 million (2015)
Deaths -  397,100 (2015)
```

Figure 8: Prediction using cosine similairy scoring model along with disease detail

5 Literature Review

There has been a lot of work carried out to predict diseases based on potential patient's symptoms and other medical health data.

Y. Zhang and B. Liu present a paper on "Semantic text classification of disease reporting" (Zhang and Liu, 2007) wherein they trained a model with sentence-level semantics for predicting infectious diseases and obtained good results.

The work proposed in (Wang X, 2008) focuses on disease prediction from clinical data provided by New York's Presbyterian Hospital. As these are clinical data, automated disease prediction is relatively different and easier than predicting from user text input. The authors observed that since users refrain from using clinical terms which indicate greater complexity while matching symptom name with the user's input.

Sen et al. (Kumar Sen, 2013) implements a system to predict coronary heart diseases by processing symptoms and other patient's specific text.

Petrov et al. (Slav Petrov, 2013) proposes a natural language processing model to take feedback, rate, and analyze comments to improvise model.

6 Baseline

Initially, we used the Multinomial Naïve Bayes model to predict top diseases as it works very well with discrete values and gives good accuracy. We achieved an accuracy of 74% in the prediction of diseases. The same task was performed with TF-IDF and cosine similarity also and the results were somewhat average. The results greatly improved after the dataset was cleaned to remove similar symptoms and with the use of better Machine Learning models.

Compared to the system proposed earlier, following are the additions/improvements:

- Initially, we were only able to work on the symptoms if they were given exactly as present in the dataset which we improved by incorporating synonyms and query expansion procedures.

- Independent probability for each disease is also calculated which shows the confidence with which the model predicts the disease.

- Functionality of suggestion of co-occurring symptoms (affinity of 2 symptoms to occur together) is added which provides more flexibility to provide a list of symptoms to the system.

- More details about the predicted diseases and treatment recommendations were not implemented initially. It was added to the system to make it a complete medical system.

A similar work described in the study (Laskar et al., 2016) reports an accuracy of 88.89% which is lesser than the accuracy of our system. Hence, our system has a better prediction capability than the previously existing system.

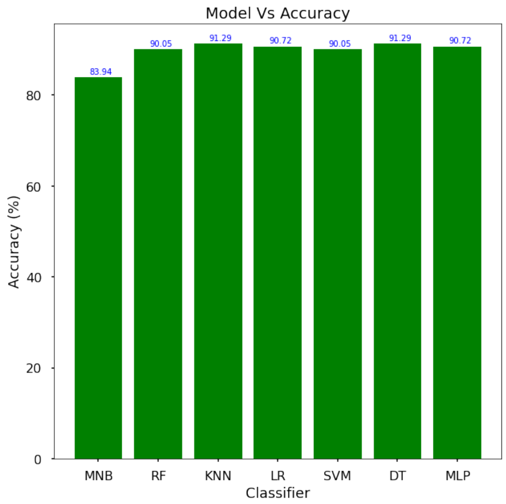
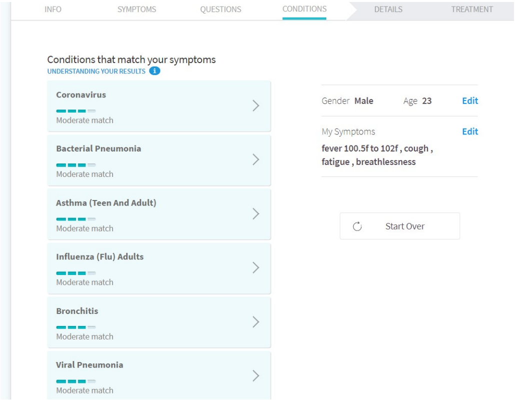


Figure 9: Model Accuracy Comparison

7 Results

Evaluation of the dataset is done by applying various machine learning algorithms and comparing the accuracy obtained from them. Figure 9 shows a comparison between different model accuracies. The highest accuracy is reported by K Nearest Neighbor (91.29%) and Decision Tree (91.29%) while the lowest is of Multinomial Naive Bayes (83.94%).

The system's performance is evaluated by comparing the predicted diseases that were obtained in Figure 6, 7, and 8 with the one obtained from WebMD's Symptom Checker Module (`https://symptoms.webmd.com/default.htm`) and it showed similar results. Figure 10 shows the predicted diseases that were obtained for the same set of symptoms as shown in Figure 4.



Conditions that match your symptoms
UNDERSTANDING YOUR RESULTS

Coronavirus
Moderate match

Bacterial Pneumonia
Moderate match

Asthma (Teen And Adult)
Moderate match

Influenza (Flu) Adults
Moderate match

Bronchitis
Moderate match

Viral Pneumonia
Moderate match

Gender Male Age 23

My Symptoms
fever 100.5f to 102f , cough , fatigue , breathlessness

Figure 10: Prediction by WebMD's Symptom Checker

References

Dr. D. P. Shukla Kumar Sen, Shamsher Bahadur Patel. 2013. A data mining technique for prediction of coronary heart disease using neuro-fuzzy. *International Journal Of Engineering And Computer Science*, 2:2663–2671.

Md Tahmid Rahman Laskar, Md Hossain, Abu Kamal, and Nafiul Rashid. 2016. Automated disease prediction system (adps): A user input-based reliable architecture for disease prediction. *International Journal of Computer Applications*, 133:24–29.

Ryan McDonald Slav Petrov, Dipanjan Das. 2013. A universal part of-speech tagset. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012).*

`https://symptoms.webmd.com/default.htm`. Webmd's symptom checker. Accessed: 2020-05-15.

`https://wordnet.princeton.edu/`. Princeton university's wordnet. Accessed: 2020-05-15.

`https://www.nhp.gov.in/disease-a-z`. National health portal(nhp), developed and maintained by centre for health informatics (chi). Accessed: 2020-05-15.

`https://www.thesaurus.com/`. Thesaurus. Accessed: 2020-05-15.

Elhadad N Friedman C Markatou M. Wang X, Chused A. 2008. Automated knowledge acquisition from clinical narrative reports. *AMIA Annu Symp Proc. 2008 Nov 6;2008:783-7. PMID: 18999156; PMCID: PMC2656103.*

Yi Zhang and Bing Liu. 2007. Semantic text classification of disease reporting. *Proceedings of the International ACM SIGIR Conference, 2007.*