European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 7, Issue 4, 2020

Symptoms Based Disease Prediction Using Decision Tree and Electronic Health Record Analysis

S Radhika[1], S Ramiya Shree [2], V Rukhmani Divyadharsini[3] and A Ranjitha[4]

Department of Computer Science, R.M.K Engineering College, Thiruvallur, India

Email : sra.cse@rmkec.ac.in ,rami16307.cs@rmkec.ac.in, rukh16314.cs@rmkec.ac.in, ranj16308.cs@rmkec.ac.in

Abstract: In this paper, we seek to predict user's diseases based on their symptoms. To achieve our target, we use the Decision Tree Classifier which helps to detect the patient's health condition after receiving their symptoms by giving the predicted disease. The dataset contains physiological measurements with 40 instances(Diseases) and 132 attributes(Symptoms). Additionally, the respective patient's EHR is also collected for summarizing the prescription/test report using NLTK.

Keywords: Decision tree, Physiological measurements, EHR (Electronic Health Records), NLP (Natural Language Processing), NLTK(Natural Language Toolkit).

1. Introduction

Health practitioners perform too many disease surveys and collect patient information, the seriousness of their disease, and symptoms that allow them to distinguish the patient's disease with common symptoms. Hence useful information hidden in the data set is used to train the model that predicts the disease based on the symptoms. We have constructed the decision tree classifier model which is trained using the dataset in a shorter period by normalizing our data using standardization techniques known as case gradient descent. After normalization, Our trained model is used to predict the disease along with the confidence level, causes, and preventive measures. Our system, on the other hand, obtains an EHR as an input file which is converted into a text file. The text file is summarized using NLTK to help the patient understand the health report. However, various modules such as booking appointments with doctors, storing and retrieving medical records, locating doctors are included along with the two major modules to make this system a complete health monitoring system.

1.1. Data Collection

This paper works with Text summarisation and Decision tree algorithms. The data were obtained from the MIMIC Electronic Health Information database for text review purposes. Columbia University gathered the New York Presbyterian Hospital's textual discharge report to examine the patterns in the document. We have categorized the dataset as testing and training data based on the study.

2. Decision Tree Algorithm

2.1. How does Decision Tree work?

The methodology used in the Decision tree is a commonly used data mining method for establishing classification and prediction systems based on multiple explanatory parameters for developing prediction models for a target instance. This path classifies a population into branch-like segments in a tree that

construct an inverted tree with a root node, internal nodes, and leaf nodes. A decision tree is a non-parametric algorithm which can efficiently deal with huge, complicated data sets without involving multiple parametric structures. If the sample size is large enough, study data can be divided into training and validation data sets. Using the training data set to build a decision tree model and a validation data set decide on the appropriate tree size to achieve the optimal final model.

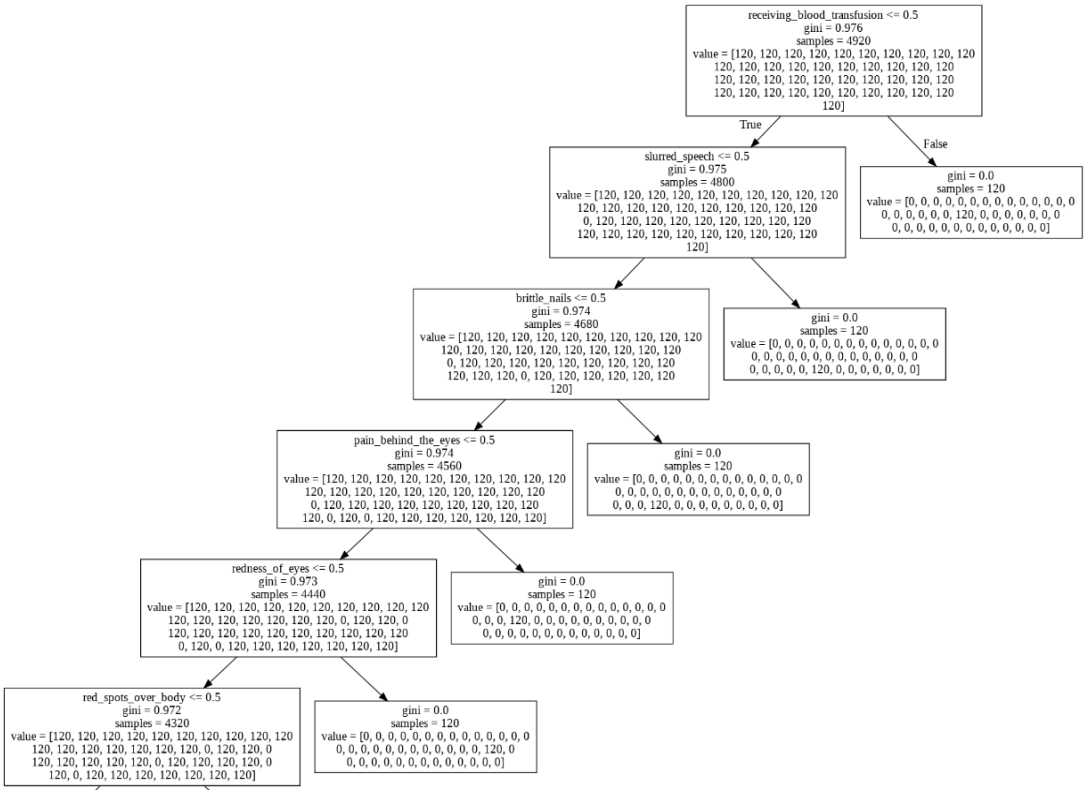


Fig. 2.1. Decision tree execution of the data set.

2.2. Execution

The Decision tree works with the underlying symptoms and predicts a disease. Initially, we get the user's top five symptoms and put it in an array with the value assigned as 1 across these values. This is passed as an input to the model for predicting the disease. This array matches the disease data collection and ends at a common leaf node with the highest degree of trust.

2.3. Recursive Part

In the recursive part, we repeat the above mentioned approach with increasing tree-level in order to construct the tree. We set the current node as a leaf node when there is no question to ask if the output is published for the symptoms given. We also use electronic health records to expand the dataset with more disease-symptom pairs for better prediction of the disease based on the symptoms.

2.4. General rules for Building Decision Tree

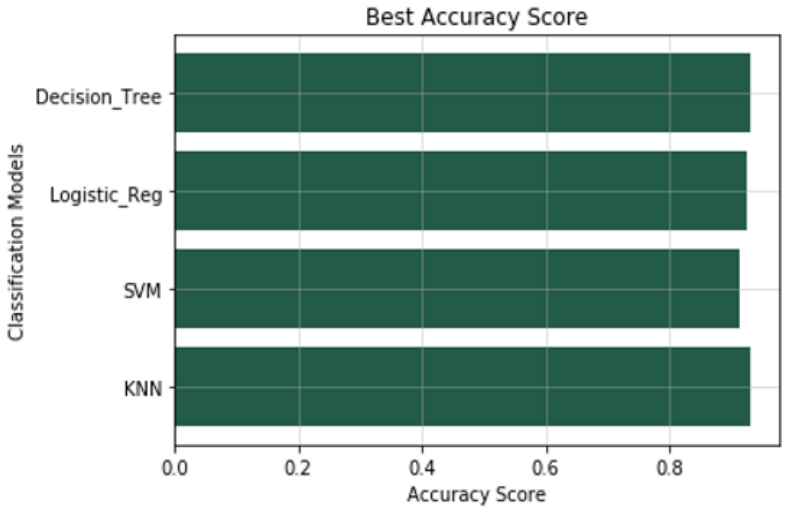
1. Choose the best attribute / feature.
2. The best attribute is the one that best separates or divides the data into subsets.

3. The recursive process ends when all elements belong to the same attribute, no more attributes and instances are left.

Table 2.1. List of diseases

Fungal infection	Alcoholic hepatitis
Allergy	Tuberculosis
GERD	Common Cold
Chronic cholestasis	pneumonia
Drug Reaction	Dimorphic haemorrhoids(piles)
Peptic ulcer disease	Heart attack
AIDS	Varicose veins
Diabetes	Hypothyroidism
Gastroenteritis	Hyperthyroidism
Bronchial Asthma	Hypoglycaemia
Hypertension	Osteoarthritis
Migraine	Arthritis
Cervical spondylosis	Vertigo
Paralysis	Acne
Jaundice	Urinary tract infection
Malaria	Psoriasis
Chicken pox	Impetigo
Dengue	Hepatitis B
Typhoid	Hepatitis C
hepatitis A	Hepatitis D

2.5. Comparative Analysis



Graph 2.1. Accuracy score comparison for decision tree

3. Decision Tree Algorithm

In this section of the paper, we will discuss summarizing the user input files. For this, we fetch the medical record from the user, which can be in any format. The record is converted into a text file and that is given as input to the summarization module. We use NLTK - the Natural Language Toolkit libraries to provide the summary of the health record.

3.1. NLTK Libraries

We are using two libraries in order to organize the various terms and its frequency factor. It is also used for maintaining the stop words.

Corpus 3.1.1: Collection of text data is called corpus. It can be a collection of poems of a poet or blog related to some topic. Here it works as a collection of predefined stop words.

Tokenizers 3.1.2: Tokenizers are used for sentences. They are used for making a series of tokens and also used for creating regex.

3.2. Steps to build summarization

1. Initially, stop words are removed from the record.
2. A frequency table for words is created- how many times each word appears in the file with the medical term weight analysis that gives importance to the medical terms.
3. Assigning score to each sentence based on the term score in the frequency table.
4. Analysed summary is generated based on the sentence score that is above a certain threshold.

3.1. Need for summarization

Technologies enabling a clear description include factors such as time, writing style, and syntax. The main principle for summarizing is to consider a specific subset of the data, which includes the information of the whole array. Text overview is commonly used to manage summaries of email messages, achieve action items, and shorten content by compressing sentences used to arrange information and promoting Web search engines.This paper works on summarizing medical health records and operative reports. Usually as mentioned earlier all types of summarization were held only for native language summarization process and news headline purposes, but in this paper, we took it to all new different worlds of the medical industry, where it helps in mining a report which will immensely help a patient to understand what his/her medical report says without consulting a medical expert. It also works along with the other modules of the application, where we assign the nearest health center and also allow the patient to book an appointment with doctors.

3.3. Equation

$$\text{Sentence score} = \text{frequency score} + \text{medical score} \quad (1)$$

$$\text{final score} = S + N \quad (2)$$

S - Sentence score
N - Number of words in sentence
This finally provides the best score for each sentence and builds the summary.

4. Proposed System

Disease/Symptoms database is collected from the Kaggle repository. As mentioned earlier, it contains several attributes (symptoms) and classes (diseases). Using this we create the training set and testing set to train the model. We get the symptoms from the user and predict the disease using the trained model. The patient record is collected on the other hand which generates a summary of the health report based on the highly important symptoms corresponding to a particular disease. This is used to expand the disease-symptom pair dataset. Usually any prediction system will only look with the predefined platform of symptom and disease pair but that would partially find the result with low confidence level. In order to reach a higher level of confidence, in this paper we go with health record analysis which gives personalised input and as well as user interaction with the system. The overall system has two modules, one is disease prediction and the other goes with a health record. In order to achieve a higher confidence level, the second module is used as one of the training data sets to provide the best result. This process is integrated with the user interface to get the user input and let them know their status.

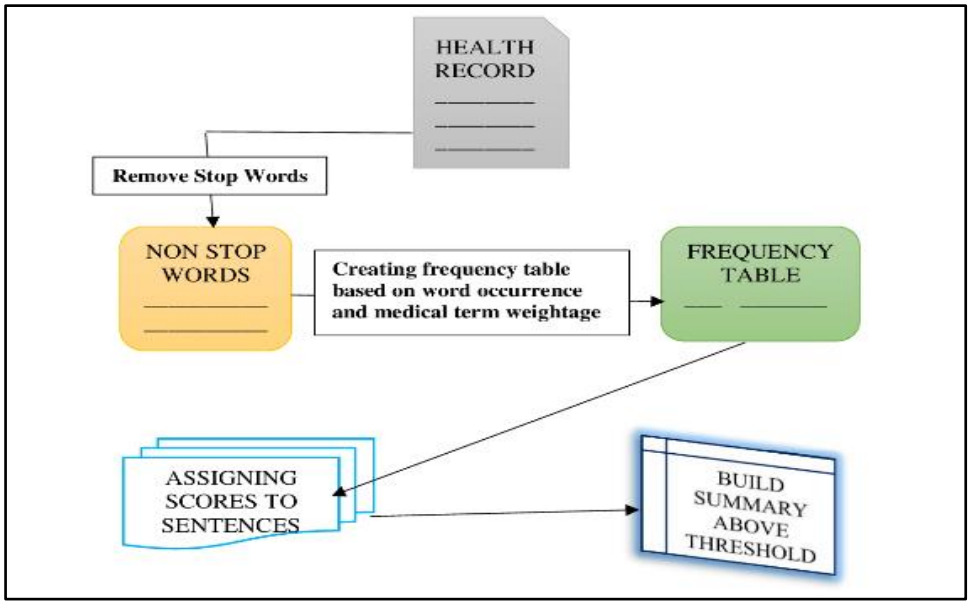


Fig. 4.1. EHR Summarization.

4.1. Prediction

The final output is predicted using the given symptoms. Summarization for the uploaded file is carried on using the NLTK along with medical term weight analysis, which gives out the optimized summary related to the patient or the user. It also helps us in maintaining the back end disease - symptom data set. Decision tree initially predicts the disease with the highest confidence level for each of the predicted diseases.

Fig. 4.2. Output Summary

5. Conclusion

As a future enhancement we also look forward to executing multilingual summarization and multi document summarization. The files which we give as input may also contain native languages, hence health records can be collected from various parts of the world and can be easily summarized using multilingual summarization. As of now the paper proceeds with global language (English). This paper clearly defines the disease prediction using highly personalised training data sets and also some of the related tasks like fixing appointments and tracing the nearest health centre.

6. Acknowledgements

We reveal our sincere thanks to our Professor and Head of the Department, Computer Science and Engineering**,** Dr. T. Sethukarasi, for her commendable support and encouragement for the completion of our project.We convey our deep gratitude and we are very much indebted to our versatile project guide Ms. S Radhika, Assistant Professor for her valuable suggestions and spontaneous guidance to complete our project.

7. References

1. *Books, book chapters and manual* - ResearchGate: Lior Rokach, Oded Maimon, Decision Trees, Tel-Aviv University https://www.researchgate.net/publication/225237661_Decision_Trees
2. *Journal articles* - Saiyed Saziyabegam, Priti S, "Literature review on extractive text summarization approaches", International Journal of computer applications (0975 – 8887), December 2016, India https://www.ijcaonline.org/archives/volume156/number12/26762-2016912574
3. *Conference proceeding* Hlaudi daniel masethe, Mosima anna masethe, "Prediction of heart disease using classification algorithms", proceedings of the world congress on engineering and computer science, october, (2014), USA. http://www.iaeng.org/publication/WCECS2014/ ,Pankaj Gupta, Nirmal Robert, RituTiwari, The International Conference on Connectivity and Signals Processing, (2016), "Meaning Interpre-tation and Text Resume on Web Reviews: A Survey."https://link.springer.com/book/10.1007%2F978-981-10-6890-4
4. *Online resources* - Disease/Symptom dataset, used for module http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html EHR dataset used for summarization in module 2 https://mimic.physionet.org