iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 Issue: VIII Month of publication: August 2020

DOI: https://doi.org/10.22214/ijraset.2020.31167

Study of Machine Learning Algorithms to Implement a Disease Prediction Application

Shravan Ramakrishna[1], Renukrishna Hegde[2], Suraj S[3], Vijay MB[4]
[1, 2, 3]*Dept. of Computer Science JSSSTU, Mysore, India*
[4]*Prof. Dept. of Computer Science JSSSTU, Mysore, India*

Abstract: There exist several models that are used in today's healthcare system to predict the probability of a specific disease by monitoring the patient's symptoms over a period. These specific diseases have popularly included heart-related diseases, diabetes, cancer prediction, etc.

There exist very few studies pertaining to machine learning algorithms that can be used to predict a general set of diseases, not restricted to a specific field of medicine. Given a completely new set of symptoms, we can generalize the area into which it falls. This aims to be one of the key motivations for the project. In this study, we aim to look into various machine learning algorithms such as Na¨ıve Bayes, Decision Trees, RF and SVM in order to figure out how to predict a general set of diseases using symptoms, following which the data may be used by other disease-specific models to make further sense of it. We will study various algorithms and the accuracy of them predicting the diseases correctly, thus trying to discover the optimum algorithm to address this issue.

The Machine Learning models can then be trained using big data and can use these sets of algorithms to further zone in on the area of a medical study. This allows several medical benefits and helps in early disease detection and better patient care in the medical industry.

Index Terms: Big Data, Machine Learning, Na¨ıve Bayes, Decision Trees, Random Forest, Support Vector Machine, Disease Prediction

I. INTRODUCTION

Many medical complications stem from the misdiagnosis or delayed diagnosis of a medical condition, illness, or injury. A patient's condition can be made much worse than before if the misdiagnosis or delayed diagnosis leads to delayed, incorrect, or no treatment at all. This signifies the severity and influence of the point at issue, the point being why using machine learning algorithms are critical to making sure misdiagnosis is reduced. Several solutions have been introduced. The number of internet users has been growing exponentially over the years.

People post their health-related queries (such as asking about what kind of disease that they might be suffering from) on various healthcare forums as the price of healthcare is increasing exponentially. There are other groups of people who leave their responses to these posts with predictions of possible diseases. However, these predictions may not be always accurate, and there is no assurance that users will always get a reply to their post. Moreover, some posts are fabricated or made up which can drive the patient in the wrong direction.

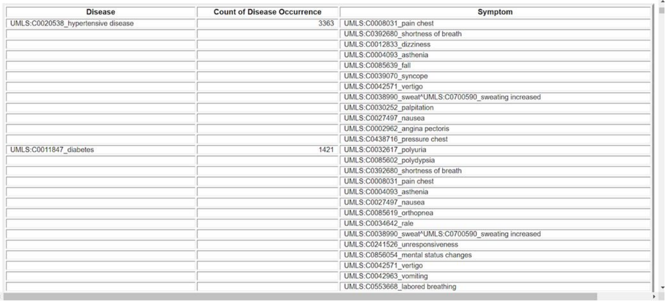
Recently, hospital websites have integrated disease predic- tion using symptoms to their websites and several applications also exist which help in aiding this process. Machine learning runs behind the scenes, analyzing the patient, allowing for several statistics to be introduced which help in information being spawned to make the appropriate diagnosis, run consid- erable tests, or suggestions of preventive screening. Another issue crops up here, that being the obstacle of obtaining clean, informative, and noiseless data to train the model. Medical records are not easily accessible and privacy concerns are omnipresent.

A handful of machine learning algorithms exist to imple- ment disease prediction using symptoms once the data has been preprocessed and obtained for training. The variety of algorithms includes the Naive Bayes Algorithm, KNN Algo- rithm, Random Forest Classifier, SVMs, CNN Text Classifica- tion, etc. These are well known Machine Learning Algorithms that are widely used in several contexts and problems. In this project, we aim to use these algorithms to predict diseases using knowledge of the patient's symptoms.

II. SYSTEM DESIGN

A. Database used for Training/Testing

Disease-Symptom Knowledge Database – The following knowledge database shows the disease-symptom associations of patients from the New York-Presbyterian Hospital from the year 2004. The information is based on the textual discharge summaries. The columns mainly specify the disease, the num- ber of discharge summaries, and the symptoms associated with that disease. The number of diseases in the following is 41. It uses the MedLEE NLP system to obtain the codes (UMLS) for the diseases and symptoms. The associations have been found by using statistics, mainly concepts based on frequencies and co-occurrences [3].



Disease	Count of Disease Occurrence	Symptom
UMLS:C0020538_hypertensive disease	3363	UMLS:C0008031_pain chest
		UMLS:C0392680_shortness of breath
		UMLS:C0012833_dizziness
		UMLS:C0004093_asthenia
		UMLS:C0085639_fall
		UMLS:C0039070_syncope
		UMLS:C0042571_vertigo
		UMLS:C0038990_sweat^UMLS:C0700590_sweating increased
		UMLS:C0030252_palpitation
		UMLS:C0027497_nausea
		UMLS:C0002962_angina pectoris
		UMLS:C0438716_pressure chest
UMLS:C0011847_diabetes	1421	UMLS:C0032617_polyuria
		UMLS:C0085602_polydypsia
		UMLS:C0392680_shortness of breath
		UMLS:C0008031_pain chest
		UMLS:C0004093_asthenia
		UMLS:C0027497_nausea
		UMLS:C0085619_orthopnea
		UMLS:C0034642_rale
		UMLS:C0038990_sweat^UMLS:C0700590_sweating increased
		UMLS:C0241526_unresponsiveness
		UMLS:C0856054_mental status changes
		UMLS:C0042571_vertigo
		UMLS:C0042963_vomiting
		UMLS:C0553668_labored breathing

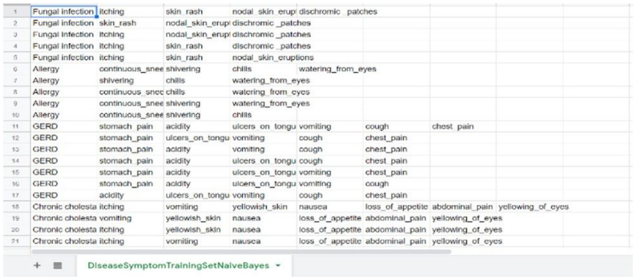
Fig. 1. A snapshot of the database used

B. Proposed Database Design

The Dataset had to be transformed into a form that could be easily be passed to the algorithms that we implement. The data transformation thus resulted in 4 important files, i.e. the list of diseases, the list of symptoms, the training, and testing sets.

An important factor that we need to take into consideration is the fact that given a set of symptoms that identify a disease, even the presence of one of these symptoms indicate a slight probability/presence of that disease. Thus, given disease, we need to be able to perform a combination of the symptoms so that we obtain a set, and then based on that further filter only the ones which address that disease in the post-processing step.

Given a Disease D which is identified by the symptoms S1, S2 and S3, then, even the presence of a single symptom such as S1 or S2 or S3 indicates that the person could be infected with the disease D. Therefore, we need to take into consideration the set S1 , {{S2 } {S3 ,} {S1, S2{ , S1, S3{ , S2, S3 } {S1; S2, S3 }} However, the symptom set {S2, S3 may not identify disease D at all. Therefore, we need to remove that from our set before we proceed. Thus, in this fashion, based on medical expertise, the dataset needs to be created appropriately before training the model.

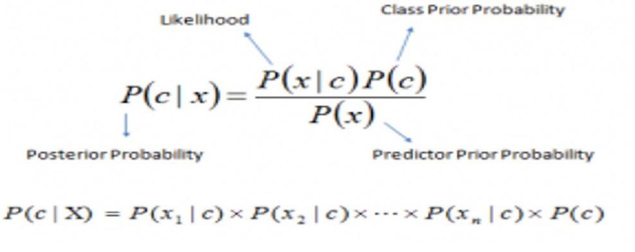


1	Fungal infection	itching	skin rash	nodal skin erup	dischromic patches			
2	Fungal infection	skin_rash	nodal_skin_erup	dischromic _patches				
3	Fungal infection	itching	nodal_skin_erup	dischromic _patches				
4	Fungal infection	itching	skin_rash	dischromic _patches				
5	Fungal infection	itching	skin_rash	nodal_skin_eruptions				
6	Allergy	continuous_snee	shivering	chills	watering_from_eyes			
7	Allergy	shivering	chills	watering_from_eyes				
8	Allergy	continuous_snee	chills	watering_from_eyes				
9	Allergy	continuous_snee	shivering	watering_from_eyes				
10	Allergy	continuous_snee	shivering	chills				
11	GERD	stomach_pain	acidity	ulcers_on_tongu	vomiting	cough	chest_pain	
12	GERD	stomach_pain	ulcers_on_tongu	vomiting	cough	chest_pain		
13	GERD	stomach_pain	acidity	vomiting	cough	chest_pain		
14	GERD	stomach_pain	acidity	ulcers_on_tongu	cough	chest_pain		
15	GERD	stomach_pain	acidity	ulcers_on_tongu	vomiting	chest_pain		
16	GERD	stomach_pain	acidity	ulcers_on_tongu	vomiting	cough		
17	GERD	acidity	ulcers_on_tongu	vomiting	cough	chest_pain		
18	Chronic cholesta	itching	vomiting	yellowish_skin	nausea	loss_of_appetite	abdominal_pain	yellowing_of_eyes
19	Chronic cholesta	vomiting	yellowish_skin	nausea	loss_of_appetite	abdominal_pain	yellowing_of_eyes	
20	Chronic cholesta	itching	yellowish_skin	nausea	loss_of_appetite	abdominal_pain	yellowing_of_eyes	
21	Chronic cholesta	itching	vomiting	nausea	loss_of_appetite	abdominal_pain	yellowing_of_eyes	

Fig. 2. Our proposed DB design

C. Na ̈ive Bayes

The Na ̈ive Bayes algorithm is given by the equation: attributes as root or internal node.



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

1) *P* (*c x*) is the posterior probability of class (c, target) given predictor (x, at- tributes).
2) *P* (*c*) is the prior probability of class.
3) *P* (*x c*) is the likelihood which is the probability of predictor given class.
4) *P* (*x*) is the prior probability of predictor.

Adopting this logic to our case, we find that we want to find the disease given a set of symptoms. Thereby, our equation will substitute the x value as the set of symptoms that are provided as input and the c will be substituted with the disease. We will need to perform this operation for all the given diseases and account for which one has the highest probability. Example Given a set of symptoms S1, S2, S3 and given two diseases in our data set D1 and D2, then we need to find P(D1 / S1 S2 S3) and P(D2 / S1 S2 S3), and this will tell us the probability of what disease the person has.

Since we have multiple conditions, we transform the for- mula as follows and then apply it. This is the main formula used in training the classifier [1]

$$\begin{aligned} P(y|f_1,\ldots,f_m) &= \frac{P(f_1,\ldots,f_m|y)P(y)}{P(f_1,\ldots,f_m)} \\ &= \frac{P(y)\prod_{i=1}^{m}P(f_i|y)}{P(f_1,\ldots,f_m)} \\ \arg\max_y P(y|f_1,\ldots,f_m) &= \arg\max_y \frac{P(y)\prod_{i=1}^{m}P(f_i|y)}{P(f_1,\ldots,f_m)} \\ &= \arg\max_y P(y)\prod_{i=1}^{m}P(f_i|y) \end{aligned}$$

The flowchart, as shown in Fig.3, describes the flow of control and the design aspects. The Laplace smoothing mainly can set a parameter α = 1 and d equal to the total number refers to a technique by which we smooth categorical data. We of features, thus addressing the fact that the probability will never be 0.

D. Decision Trees

1) We consider the whole training set as the root
2) The attributes are assumed to be categorical for the information gain.
3) Based on the attribute values, records are distributed recursively. We then use statistical methods for ordering

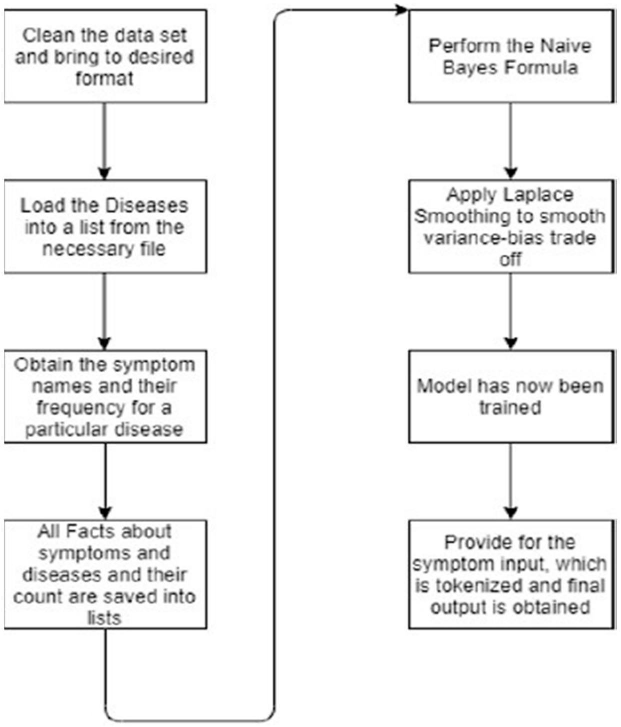

Clean the data set and bring to desired format
Load the Diseases into a list from the necessary file
Obtain the symptom names and their frequency for a particular disease
All Facts about symptoms and diseases and their count are saved into lists
Perform the Naive Bayes Formula
Apply Laplace Smoothing to smooth variance-bias trade off
Model has now been trained
Provide for the symptom input, which is tokenized and final output is obtained

Fig. 3. Na¨ıve Bayes Flow Diagram

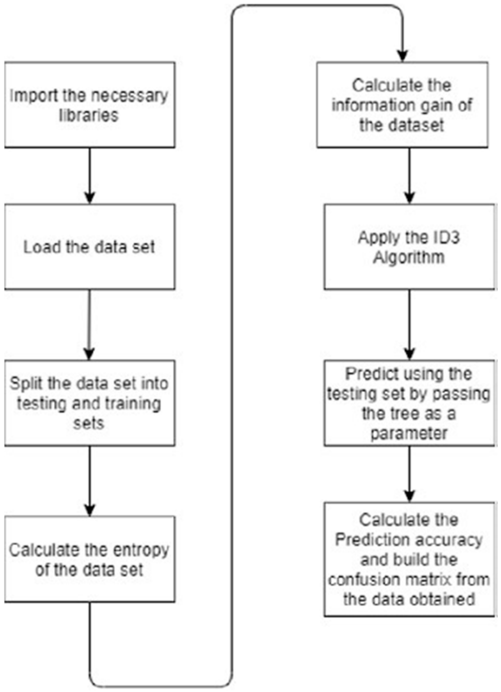

Import the necessary libraries
Load the data set
Split the data set into testing and training sets
Calculate the entropy of the data set
Calculate the information gain of the dataset
Apply the ID3 Algorithm
Predict using the testing set by passing the tree as a parameter
Calculate the Prediction accuracy and build the confusion matrix from the data obtained

Fig. 4. Decision Trees Flow Diagram

E. Random Forest

As splits continuously occur, the accuracy of the decision tree model continues to improve. a given training dataset the accuracy keeps improving with more and more splits. You can easily overfit the data and it does not know when you have crossed the line unless you are using cross-validation (on training data set). It is easy to know what variable is used and which variable is used in splitting the data to predict the outcome. This is a key advantage of decision trees. This is one of the reasons, for a problem like a disease prediction, overfitting is a big issue and since it is not a small data set, there will be a low accuracy. A random forest is like a black box and works as mentioned above. It is a forest you can build and control. You can specify the number of trees you want in your forest (n estimators) and you can specify the maximum num of features to be used in each tree. When it comes to the tree, we do not possess any control over which data point or feature is part of the tree. This in turn implies that we have no control over the randomness. As we increase the number of trees, the accuracy increases in turn as well, however, after a point, it becomes constant. Unlike the decision tree, it will not create a highly biased model and reduces the variance [2]

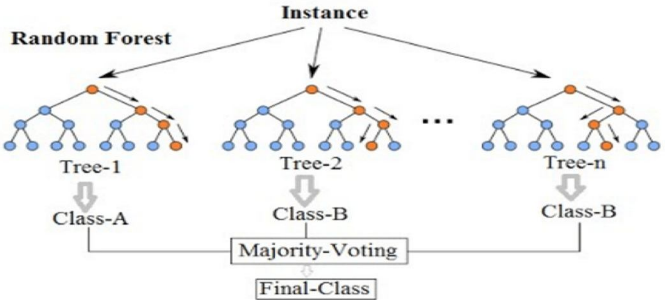


Fig. 5. Random Forest Diagram

F. Support Vector Machines

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features.

III. SYSTEM IMPLEMENTATION

The system was implemented using the Python language. The basis for coding was via the flowcharts that have been presented earlier. The Pandas and NumPy libraries were used to clean and transform the data. NLTK was used for tokenization and parsing Sklearn was used to verify certain important attributes. PyDotPlus was used to generate visual Decision Trees to obtain more visual clarity. This comprised of the backend. The front end mainly consisted of a Flask, which is a micro web framework, which was used along with React, which is a JavaScript library used for building user interfaces. The JSX syntax extension for JavaScript is also used additionally to build templates.

IV. RESULT

After cleaning the data and bringing it to the form necessary for input to the algorithm, we coded the algorithms and observed the disease prediction capabilities. We performed tests based on the accuracy factor and observed that it was the highest for the Random Forest Classifier Algorithm. The value pertaining to our small dataset (relative to the real world) was 98.2%. This was followed by the Na¨ıve Bayes Algorithm with an accuracy of 97.5%. The simplicity of the Na¨ıve Bayes algorithm once again speaks for itself. Support Vector Machine came in next with an accuracy of 95.08% and the last was Decision Trees with a very poor accuracy of 45.9%. This was mainly due to the reason that given a set of symptoms, even the presence of a single symptom still provides for a small probability of that disease. When working with Multiclass Problems, the Random Forest algorithm is ideally suited to meet the needs of such a dataset. SVM is intrinsically a two- class problem and can be used along the lines of deduction and inference. The multiclass problem of disease prediction will need to be reduced into multiple binary classification problems in that case.

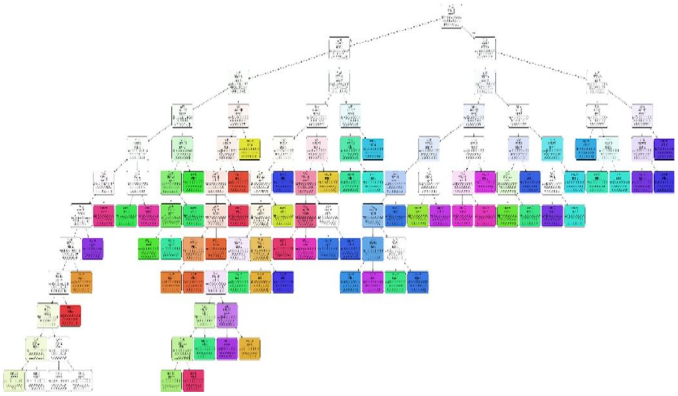


Fig. 6. Output for Decsion Tree

V. CONCLUSION AND FUTURE WORK

The project aims to address the topic of disease prediction using symptoms. In the process of implementing the project, the several steps needed to materialize it into an application are realized methodically. A proper management system inte- grated with ML algorithms will allow a hospital to perform at its optimum and treat as many patients as possible. The future scope for the project is bright and involves optimizing algorithms to make better predictions as well as to recommend medication. Larger, Processed Big Data sources of data may be used to train models to further a more realistic prediction. For a brighter future, healthcare will be one of the most influential industries which will have the most impact in the world. The future work for the project mainly banks on trying to find new algorithms and methods in order to be able to try new approaches that will hone in on the disease in more detail, which in turn will allow medical professionals to handle diseases easier. We have only scratched the surface as there are experts who are now combining algorithms to use the advantages each of them has to offer. As observed in one of the literature surveys, heart disease prediction is now being performed by combining the KNN algorithm along with the Genetic algorithm by taking the best of both. Medicine recommendation systems are also being built, however it will take a lot of work in order to be able to confidently trust a machine learning algorithm to be able to recommend medicine, as it is a matter of life. Further innovations are being brought about to make these systems more of a norm and to be able to provide these services easily via the internet. The final goal of this work however is to find algorithms that will be able to predict the disease accurately, given a set of symptoms. The future work in this aspect will next be to try to combine algorithms as mentioned above, and to be able to obtain better accuracy, precision, and performance.

VI. ACKNOWLEDGMENT

An endeavor is successful only when it is carried out under proper and constant guidance. We would take this opportunity to thank a few people. We extend our gratitude to Dr. S B Kivade, Principal, JSS Science and Technology University, for providing an environment for our education and his encourage- ment during our stay in college. We would like to convey our gratitude to Dr M P Pushpalatha, Head of Department of Com- puter Science and Engineering, for giving us the opportunity to enter into this journey of knowledge. We are grateful to Prof. Vijay M B, Assistant Professor, Department of Computer Science and Engineering, for his guidance, assistance, support and criticism for the improvement of the project. We would also like to thank our friends who supported us directly or indirectly towards the progress of our project.

REFERENCES

[1] R. Abraham, J. B.Simha and S. S. Iyengar, ”A comparative analysis of discretization methods for Medical Datamining with Naive Bayesian classifier,” 9th International Conference on Information Technology (ICIT’06), Bhubaneswar, 2006, pp. 235-236, doi: 10.1109/ICIT.2006.5.

[2] Alickovic E, Subasi A. Medical Decision Support System for Diagnosis of Heart Arrhythmia using DWT and Random Forests Classifier. J Med Syst. 2016;40(4):108. doi:10.1007/s10916-016-0467-8

[3] Wang X, Chused A, Elhadad N, Friedman C, Markatou M. Automated knowledge acquisition from clinical reports. AMIA Annu Symp Proc. 2008. p. 783-7. PMCID: PMC2656103.