Disease Prediction based on symptoms

Ujwal K R
Computer Applications
Data Analytics
National Institute of Technology
Tiruchirapalli, Tamil Nadu, India
krujwalpersonal@gmail.com

Dr. Srinivasalu Reddy
Computer Applications
National Institute of Technology
Tiruchirapalli, Tamil Nadu, India
usreddy@nitt.edu

Mr. Purushottam Kumar
Computer Applications
National Institute of Technology
Tiruchirapalli, Tamil Nadu, India
405121001@nitt.edu

***Abstract*— We've relied on doctors and physicians for years to diagnose our illnesses and prescribe proper treatment depending on the diagnosis. However, no offense intended, this technique demands the presence of a doctor in a nearby location; in many cases, patients are compelled to visit a doctor, which is exceedingly difficult in many situations, especially during disasters. There are various ways to bring doctors to your home via video conferencing and other technology, but you'll need a faster internet connection. This study describes a website where users can enter symptoms, and the website will forecast diseases using a dataset obtained from a variety of sources, as well as powerful research and machine learning algorithms applied to the dataset. We are working to enhance our disease forecasts, which are accurate 90% of the time. Users would be able to predict disease at an early stage in any ailment if this technique were employed, and they would be able to see a doctor as soon as possible.**

Keywords—Decision Tree, KNN, Random Forest, Stochastic Gradient Descent, Logistic Regression, Ada Boost, Naïve Bayes, Light Gradient Boosting Machine (LGBM), Support Vector Machine, Multi-Layer Perceptron.

I. INTRODUCTION

For millennia, doctors have been treating patients and providing guidance. Hippocrates, the world's first documented doctor/physician, classified sickness into four categories: acute, chronic, endemic, and epidemic, and laid the groundwork for his oath, which mandates doctors all over the world to help and protect people of all ages, genders, and races. The main goal is to show that doctors have improved their ability to treat patients and that customers trust their advice. Doctors use a variety of techniques and drugs to diagnose and treat their patients. These procedures have been modified to incorporate traditional knowledge of such activities as well as modern machinery and appliances to aid in decision-making processes in order to determine the fundamental cause of the condition and provide recommendations or prescriptions based on that information. However, in recent years, the practice of people meeting with a doctor in person to get information about a health-related problem has been called into question, and solutions have been proposed to address situations where doctors are unable to meet with their patients in person due to differences in location, condition, or treatment quality. Numerous solutions, such as virtual conferencing between two such parties, virtual surgery of such a patient in critical care scenarios, and so on, are implicated in real-world diagnostic and therapeutic techniques. There are several challenges with such changes, such as network connectivity and patient interface issues, but people are working to address them.

As a result, we're presenting this project, which depicts an application that employs machine-learning algorithms to predict disease based on symptoms.

II. LITERATURE SURVEY

In this project work they have concentrated more towards communities where the diseases are frequent, and they are chronic. For this purpose, they have used only a specific set of Algorithms to predict the disease. In our system, there are three Modules: Admin, User (Patient), and Doctor. Every new user must get registered through admin. After successful registration user needs to enroll first before login. Users will need to enroll only once [1]. In this research, machine learning techniques were used to design an efficient automated disease diagnostic model. They chose three essential diseases in an individual, such as heart disease, Corona Virus, and Diabetes, and all these diseases are predicted based on parameters like travel history, age, gender, and blood pressure and answer to these questions. In this model, an android app is used to take the data in. As usual the process is like what we have already seen before. We get the data and do an analysis with already existing data which is the in database. The model is already trained on similar data hence the expectation is that the model gives correct output. Later the model is also deployed in real time Application [2]. In this paper they have mainly used classification algorithms namely Decision Tree, SVM, Random Forest, KNN and Niave Bayes for Disease Prediction. They have also not chosen the entire features available but have restricted themselves to a limited number of features for better results. The total available features were 132 but they have just considered 95 for better results [3]. In the proposed paper, to accurately predict possible diseases they have used ML Algorithms. Support Vector Machine (SVM) was used for classification and Multilinear Regression (MLR) for predicting the result. Diseases predicted are Malaria, Dengue, Covid-19, Normal flu, and Filaria disease [4]. In this proposed system, three algorithms, i.e., Decision Tree model, LightGBM model, and Random Forest classifier model, were used to predict the diseases so that the predictive analysis study is proposed at the end of the study by exploring its speed, efficiency, and performance of the various algorithms for the input dataset [5]. In this paper, to the already existing work they have done something extra which will help the medicine industries in developing medicines for diseases which

are viral using the Machine Learning Techniques. As usual, first they will use the common technique of predicting the diseases based on symptoms and later suggest a best medicine for the disease [6]. In this work, again Patients health condition is checked by symptoms they have received but only using Decision Tree Algorithm. The dataset is the same dataset which was used in the previous works. One more peculiar thing about this paper is that prescription report is generated using NLTK at the end of the diagnosis [7]. Here they are trying to get the disease for the given symptoms. This system uses a decision tree classifier for evaluating the model. This system is used by end-users. The system will predict disease based on symptoms. For Disease prediction, the system has used Decision Tree Classifier Machine Learning Algorithm [8]. In the proposed system, four machine learning algorithms such as Random Forest, Naïve Bayes, KNN, and Decision Tree were used to evaluate the user prompted five symptoms which helped in getting accurate prediction. One way this system is improving and getting better results is through exploring more data sets from different diversities and communities of people [9]. The proposed system already has a search engine. User can use this search engine to search for a disease and what are the treatments available for the disease. Pre-processing of symptoms are done before hand to make the entire search process easier which in turn helps in identifying a disease very easily. What happens is, there is a database of medical records associated to a disease which will help in determining the disease when a particular term is given as an input. With this, diseases are predicted, and a pattern is found so that it will help in finding the correct disease [10].

III. MACHINE LEARNING ALGORITHMS

A. SUPPORT VECTOR MACHINE(SVM)

A Support Vector Machine also known as SVM is a classification as well as a regression machine learning technique that analyses data. One of the machine learning algorithms that classify data into groups once it has been analyzed is SVM. An SVM builds a map of the sorted data with the greatest margins feasible between the two. SVMs are used in text classification, image classification, handwriting recognition, and a variety of other scientific applications. A support vector machine (SVM) or a support vector network (SVN) is a supervised learning algorithm [16].

B. K NEAREST NEIGHBOUR(KNN)

There are many supervised learning algorithms and one such algorithm is KNN which stands for K Nearest Neighbour. We can use it to categorize things (most commonly and sometimes for Regression). It is a very versatile technique that may be used to compute missing values as well as resample datasets. One of the advantages of KNN Data Points is we can use this classify as well as predict a continuous value when a new datapoint arrives [15].

The algorithm's learning is:

1. It is an Instance based Learning: Here we use the complete dataset to train the model and then to predict for an unseen data which is unlike other algorithms where a training dataset is used for learning.
2. We don't train the model prior like we do in other algorithms. We train the model when there is a requirement for predictions. Hence it is called Lazy Learning,
3. There is no predefined form of mapping; as a result, it is non-parametric

C. RANDOM FOREST

For Classification, we can use this Algorithm. This is an ensemble learning technique. Here we need to train many Decision Trees. Once the training is done the results are averaged and returned for Regression Tasks. This Machine Learning Technique reduces the chance of over fitting the training data set. This gives better results over Decision Tree most of the time and also this is an ensemble technique. One of the peculiar and most interesting things about Random Forest is, we can use this Algorithm on both Classification and Regression Problems. So, this gives an advantage for Random Forest when it comes to Classification Problems [14]

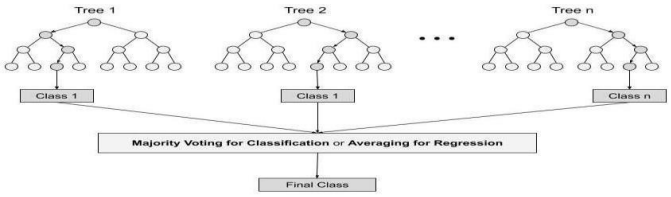

Fig 1. Random Forest Model

D. LOGISTIC REGRESSION

This is a statistical analysis model that may be utilized in machine learning applications and is frequently used for predictive analytics and modelling. The dependant variable can be finite, such as X or Y in binary Regression, or multinomial, such as X, Y, Z, and more in multinomial Regression. Its principal applications are when employing probabilistic estimation with the logistic regression function to determine the dependent variable with regard to another independent variable. This aids in determining the likelihood of an occurrence occurring or the likelihood of a decision being made.

For example, you could want to know how likely a visitor is to choose an offer on your website – or not (dependent variable). Visitors' known characteristics, such as the sites from which they came, repeat visits to your site, and site behaviour, can all be investigated in your study (independent variables). You can use logistic regression models to determine which visitors are most likely to accept or reject your offer. As a result, you'll be in a better position to make judgments about how to advertise your offer or the offer itself [13].

E. STOCHASTIC GRADIENT DESCENT (SGD)

A system or process is described as stochastic if it has an arbitrary probability associated with it. Due to this, large number of samples are taken at random in this case instead of taking the complete dataset each time. In Grade Descent, the term "batch" refers to the total number of samples from a dataset used to determine the grade for each replication. In

typical Grade Descent optimization, such as Batch Grade Descent, the batch is taken to be the complete dataset. However, using the entire data set is useful for getting to the minima in a less noisy and random manner, but this becomes an issue as our datasets grow larger.

If you utilize a typical Grade Descent optimization approach and have a million samples in your dataset, you'll have to use all of them to complete one replication while executing the Grade Descent, and you'll have to do this for each replication until the minima are reached. As a result, conducting becomes computationally valuable.

The solution to this problem is Stochastic Gradient Descent. In SGD, each replication is done with a single sample, resulting in a batch size of one. In preparation for duplication, the sample is randomly thrown together and given a name [12].

F. DECISION TREE

One of the most extensively used too for categorization is the Decision tree, as it is powerful too. A tree like structure with internal nodes and leaf nodes representing features and conclusions of test cases respectively. A leaf node also has class label in it. We use a technique called recursive partitioning to divide the source into subgroups and this division is based on the attribute's value. This method is called recursive partitioning. The recursion is complete when all the subsets at a node and the target variable have the same value or when splitting has no impact on adding values to the predictions. A decision tree classifier is highly suited to exploratory knowledge discovery because it does not require domain expertise or parameter selection. Decision trees can handle high-dimensional data. The decision tree classifier's accuracy is generally good. Decision tree induction is a common inductive approach for learning classification information. By sorting instances along the tree to a leaf node from the root, which offers the classification, decision trees are used to classify them. As seen in the figure above, an instance is categorized by starting at the tree's root node, checking the attribute specified by this node, and then progressing along the tree branch based on the attribute value. The same procedure is used for the subtree rooted at the new node [20].

G. NAÏVE BAYES

Naïve Bayes is mainly for classification problems. The heart of the classifier is Bayes Theorem.

$$P(A/B) = P(B/A) * P(A)/P(B) \quad (1)$$

With the hypothesis as A and evidence as B, using Bayes theorem we can calculate what are the chances of A occurring with B has already occurred. It is assumed that all the features or predictors are independent which means to say that presence of one does not bear any consequence on the other which makes it Naïve [21].

Equation 1 gives the formula for the Naïve Bayes Theorem.

H. MULTI-LAYER PERCEPTRON

Artificial neural networks are also known as neural networks or multi-layer perceptron, after the most often used type of neural network. A perceptron is a single-neuron model that paved the way for larger neural networks. It's an area of computer science that investigates how simple biological brain models may be used to solve difficult computational challenges like machine learning predictive modelling. Rather than creating actual brain models, the goal is to build strong algorithms and data structures that may be used to describe complex scenarios.

The ability of neural networks to learn how to properly match the representation in your training data to the output variable you want to predict is their strength. In this sense, neural networks learn mapping. They have been shown to be a approximation algorithm which is universal and can learn any mapping function mathematically. Neural networks' predictive potential comes from their hierarchical or multi-layered structure. The data structure could learn to recognize (represent) features at different scales and resolutions and combine them to form higher-order features, for example, from lines to groupings of lines to shapes [22].

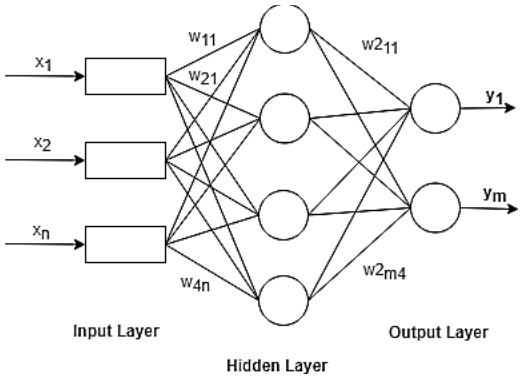

Fig. 2. Multi-Layer Perceptron

I. ADA BOOST

There are many ensemble boosting algorithms and one such algorithm is Ada Boost, or it is sometimes also called as Adaptive Boosting. To improve classifier accuracy, it mixes several classifiers. AdaBoost is repetitive method creation method. AdaBoost is a powerful classifier and has high accuracy and it does this with the help of combination of many low performing one's. It has a peculiar way of establishing weights to classifiers by training the data sample in each iteration. This kind of activity helps in finding reliable prediction when an unfamiliar data is found. As a simple classifier, any machine learning algorithm that accepts weights on the training set can be utilized. Adaboost must meet two requirements:

1. We used various training examples which are weighted and used to train this Classifier
2. To minimize the error in each iteration as every other Algorithms it also tries to provide an excellent fit [23].

J. LIGHT GRADIENT BOOSTING MACHINE

LightGBM is a Decision Tree based model which uses less memory to enhance the efficiency of the model. There are two techniques which are used here: Gradient based One Side

Sampling and Exclusive Feature Bundling (EFB). EFB overcomes the typical shortcomings of GBDT which is Gradient Boosting Decision Tree. These two techniques are the main components which defines LGBM. They together help in making the model work efficiently over other frameworks [24].

K. VOTING CLASSIFIER

Voting Classifier is a classifier which gives output based on a probability [27]. The probability is considered by taking numerous algorithms and giving the results based on them. There are basically two types:

1. Hard Voting is based on highest number of votes. Suppose two out of three classifiers gave same output, then that output is taken into consideration.
2. Soft Voting is based on averages. Here output is taken into consideration by taking the averages of all the classifiers output.

In our proposed system we have used Soft Voting for computation, and we have taken Decision Tree, Naïve Bayes, Random Forest, Logistic Regression, LGBM, KNN and Ada Boost as estimators to get the required result.

IV. VISUAL PLOTS

A. CONFUSION MATRIX

This is a matrix used to analyze the performance of our model, and the dimension of this matrix is NxN. Here, to see how well the model performs we try to find a relation between actual and predicted values, and it gives an aggregated view of the performance of the model. This matrix will also help in figuring out the errors the model is making in prediction [11].

Here our confusion matrix consists of a total of 42 rows and columns. The values are represented as follows:

- Actual Values = Columns
- Predicted Values = Rows

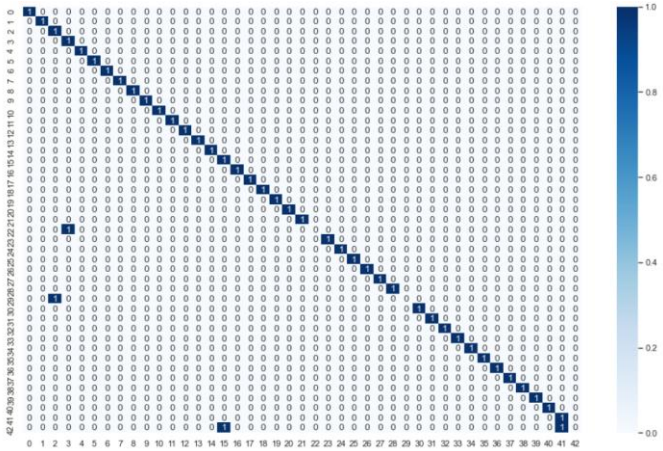

Fig. 3. Confusion Matrix

Fig 3 represents the Confusion Matrix from our Decision Tree model.

In the above Confusion Matrix from our Decision Tree model, we can see that there are four cases where the Tee has incorrectly classified the data and in the remaining 40 cases the Tree has correctly classified the data.
This Confusion Matrix is one of the easiest ways to check how well your Model is performing along with the help of True Positive, True Negative, False Negative and False Positive. If both expectation and Prediction are same, then it comes under True Category else it comes under False Category.

B. HISTOGRAM / BAR GRAPH

A histogram has rectangles along with their bases which are connected, and they represent various types of data. A histogram also represents class boundary distances. Height of rectangles is in proportion to the similarity between the different classes [19].

To represent data in a graphical format we can use Histogram because we can use contiguous ranges of numbers on data which in turn can be represented by a rectangular bar which is vertical in shape.

1. The X axis displays the total range of the numbers.
2. The Y axis (frequency) represents how much data is present in each these ranges of numbers.

The number ranges depend upon the data that is being used.

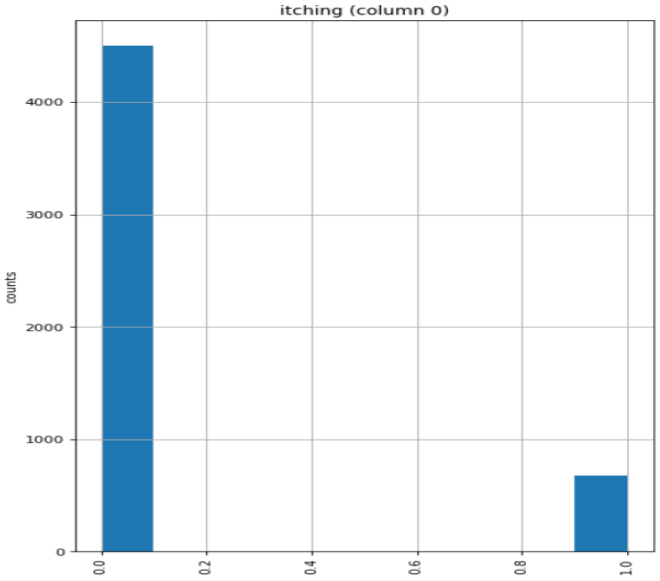

Fig. 4. Histogram for the count of itching

Fig 4 shows the total number of itching, a feature in the training dataset. There are a total of 678 rows out of 5180 rows which has an itching value of 1.

C. SCATTER AND DENSITY PLOTS

A scatter plot is a chart that is used to visually illustrate and observe the relationship between variables. A scattergram, scatter graph, or scatter chart is another name for it. On a scatter plot, the data points or dots represent the specific values of each data point and allow pattern detection when looking at the data holistically. The scatter plot is most used to show the relationship between two variables and to investigate the nature of that relationship. The observed associations can be positive or negative, non-linear, linear, strong, or weak [25].

V. TESTING PARAMETERS

A. *R2 Value*

R-squared value helps us in knowing how good our model is when we fir the data. When we add the variables which are independent, the value or figure will show the percentage of variance that the independent factors account for the dependent variables. This uses a scale of 0-100 to know the strength of the model and the dependent variable [17].

B. *Accuracy*

To know how well the model is performing one of the best metrics we can use is Accuracy. Accuracy is mainly used in Classification problems [18]. It basically means that how many times is our model correctly predicting the output. It goes by the below definition:

Accuracy = Number of Correct Predictions/Total number of predictions. (2)

VI. DATA SET DESCRIPTION

In our proposed system, we needed symptoms and diagnosis of patients. To get this data, we have taken the hep of hospitals and online resources. The data we have collected is structured. We are using true datasets that give higher accuracy.

In the proposed system, we utilize machine learning algorithms to predict diseases based on patients' symptoms. In this system, we are predicting 43 diseases based on 131 symptoms, and in total, we have 5180 rows of data for training the models. For testing the model, we have 43 rows of data, or we have a one-row dataset for each disease.

VII. PROPOSED WORK

The proposed system is a Desktop GUI that consists of labels, message boxes, buttons, text, titles, and choosing menus and was created using a simple Tkinter [26] interface as the GUI. At the start, the user is requested to put in his name. The user will be unable to progress unless he enters his name. Symptoms are on the options menu, and a user can enter a maximum of 5 Symptoms and a minimum of two. The disease cannot be predicted unless the user provides at least two symptoms. After entering the symptoms, the user is requested to select one of the four Algorithms by clicking the Predict button. Following that, the outcomes are shown. The user should click on the Reset Inputs option to reset the inputs, then the Exit system button to exit the GUI.

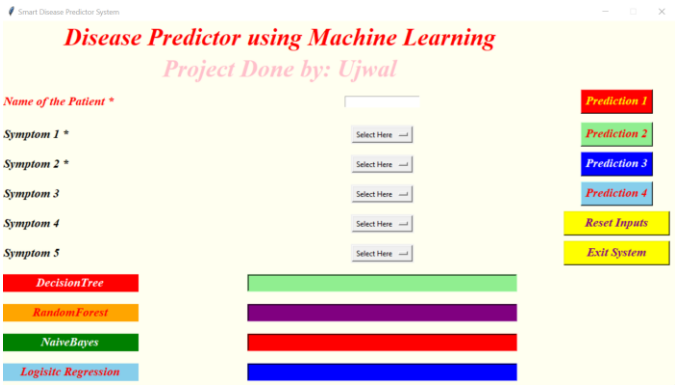

Fig. 5. GUI for Disease prediction

Fig 5 shows the GUI where the user can enter his details along with symptoms and can get the results for four different algorithms.

VIII. METHODOLOGY

In this proposed system if Disease Prediction we have used mainly four Machine Learning Algorithms such as:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Logistic Regression
4. Naïve Bayes Classifier

First, we used all four classifiers to train our disease prediction system independently, and then we analysed the results. Because accurate diagnosis and prognosis of disease are critical for the proper treatment of a patient. After analysing the data, we discovered that all four classifiers predict the same diseases in many cases based on the symptoms. Similarly, three classifiers indicated one disease, whereas one classifier predicted a different disease. As a result, we considered the results of all four Algorithms and displayed the results.

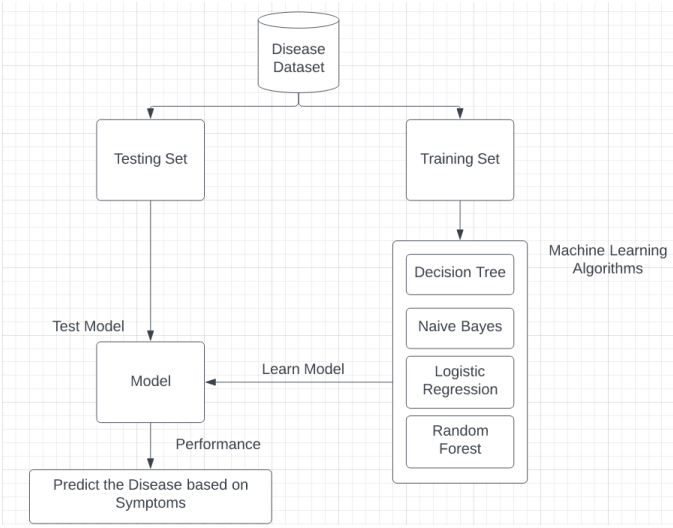

Fig. 6. Workflow for Disease Prediction

Fig 6 shows the complete workflow of the project.

IX. RESULTS AND DISCUSSION

The model was trained on the medical record of 5180 patients. We have decreased the dimensionality by considering only 94 out of 132 symptoms to avoid overfitting.

From the below table, we can infer that all four algorithm shows the excellent result, with all giving an accuracy of 91 percent.

In all these algorithms, even though the accuracy is 91% and is incorrectly predicting four diseases out of 44 diseases, those four incorrectly predicted diseases are different in each model.

TABLE I. Snapshot of results of various models

Model	Accuracy	R2 - Value
Random Forest	0.91	0.74
Logistic Regression	0.91	0.74
Decision Tree	0.91	0.74
Naïve Bayes	0.91	0.74
Support Vector Machine	0.91	0.74
Multi-Layer Perceptron	0.91	0.74
Stochastic Gradient Descent	0.91	0.74
Light Gradient Boosting Machine	0.91	0.74
K Nearest Neighbour	0.89	0.47
Ada Boost	0.64	0.46
Voting Classifier	0.91	0.74

Table 1 has all the Algorithms results which were used in the project.

X. CONCLUSION AND FUTURE WORK

To forecast diseases, we employed Random Forest, Logistic Regression, Decision Tree, and Naïve Bayes in our research. We've also put Support Vector Machine, Multi-Layer Perceptron, KNN, Ada Boost, Stochastic Gradient Descent, LGBM, and more algorithms to the test. Despite my tests, I discovered that the Random Forest, Nave Bayes, Vector Classifier, Multi-Layer Perceptron, and Decision Tree algorithms produce superior outcomes than the others.

The main objective of this work was to help people be it doctors, medical students or even a layman with the medical diagnosis information when they have some symptoms. This could also help people who are into collecting data about diseases and their symptoms

In this study, we discovered that disease prediction accuracy could reach 91 percent for some diseases and as low as 64 percent for some algorithms, but these results were obtained using the smallest amount of data set possible; however, if we feed the system a large amount of data set, disease prediction accuracy can reach 99 percent. Obtaining a massive number of data sets connected to diseases and their symptoms takes a long time and cannot be completed in one or two years; instead, it will take many years to gather those data sets and train the model using those data sets.

The focus of future work will be on providing medical assistance and suitable medication to patients as quickly as possible to construct the best infrastructure and the quickest and easiest approach in the medical sector.

XI. REFERENCES

[1] Nishant Yede, Ritik Koul, Chetan Harde, Kumar Gaurav, Prof. C.S.Pagar, "General disease prediction based on symptoms provided by a patient" 2021, volume 6, Open Access International Journal of Science & Engineering.

[2] Naresh Kumar, Nripendra Narayan Das, Deepali Gupta, Kamali Gupta, and Jatin Bindra, "Efficient automated disease diagnosis using machine learning models," volume 2021, article ID 9983652, Journal of Healthcare Engineering.

[3] K. Venkatesh, K. Dhyanesh, M. Prathyusha, C.H. Naveen Teja, "Identification of disease prediction based on symptoms using machine learning" June 2021, A journal of Composition theory.

[4] Md. Ehtisham Farooqui, Dr. Jameel Ahmad, "Disease prediction system using support vector machine and multilinear regression," Volume- 8, Issue- 4, July- 2020, International Journal of Innovative Research in Computer Science & Technology (IJIRCST).

[5] Talasila Bhanuteja, Kilaru Venkata Narendra Kumar, Kolli Sai Poornachand, Chennupati Ashish, Poonati Anudeep, "Symptoms based multiple disease prediction model using machine learning approach," Aug 2021, article in International Journal of Innovative Technology and Exploring Engineering.

[6] Jay Prakash Gupta, Ashutosh Singh, Ravi Kant Kumar, "A computer-based disease prediction and medicine recommendation system using machine learning approach," Volume 12, Issue 3, March 2021, pp.673-683, International Journal of Advanced Research in Engineering and Technology (IJARET).

[7] S Radhika, S Ramiya Shree, V Rukhmani Divyadharsini, A Ranjitha, "Symptoms based disease prediction using decision tree and electronic health record analysis," Volume 7, Issue 4, 2020, European Journal of Molecular & Clinical Medicine.

[8] Raj H. Chauhan, Daksh N. Naik, Rinal A. Halpati, Sagarkumar J. Patel, Mr. A.D.Prajapati, "Disease prediction using machine learning," e-ISSN: 2395-0056, Volume: 07 Issue: 05, May 2020, International Research Journal of Engineering and Technology (IRJET).

[9] Anuj Kumar, Mr.Analp Pathak, " A machine learning model for early prediction of multiple diseases to cure lives," Vol.12 No.6 (2021), 4013-4023, Turkish Journal of Computer and Mathematics Education.

[10] R. KAVITHA, N. MOHANAPRIYA & Dr.P. PRABAHARAN, "Disease prediction based on symptoms using machine learning algorithm," ISSN NO: 0776-3808, AEGAEUM JOURNAL.

[11]https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

[12] https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/

[13] https://www.ibm.com/topics/logistic-regression

[14] https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[15]https://www.analyticsvidhya.com/blog/2021/04/simple-understanding-and-implementation-of-knn-algorithm/

[16]https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[17] https://statisticsbyjim.com/regression/interpret-r-squared-regression/

[18]https://developers.google.com/machine-learning/crash-course/classification/accuracy

[19] https://www.cuemath.com/data/histograms/

[20] https://www.geeksforgeeks.org/decision-tree/

[21] https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[22]https://machinelearningmastery.com/neural-networks-crash-course/

[23]https://www.datacamp.com/community/tutorials/adaboost-classifier-python

[24]https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/

[25]https://corporatefinanceinstitute.com/resources/knowledge/other/scatter-plot/

[26] https://docs.python.org/3/library/tkinter.html

[27] https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/