iJournals: International Journal of Software & Hardware Research in Engineering (IJSHRE)
ISSN-2347-4890
Volume 9 Issue 1 January 2021

Heart Disease Prediction System

Authors: Ashish Sunuwar[1], Ritesh Adhikari[2], Kaustuv Giri[3], Aruj Prasad Pradhan[4], Asst.Prof. Prasanth T[5]

Affiliation: Jain (Deemed to be University), Faculty of Engineering and Technology, Bengaluru, KA, India

Email: 17btrcs143@jainuniversity.ac.in

DOI: 10.26821/IJSHRE.9.1.2021.9128

ABSTRACT

World Health Organization states that health is "a state of complete physical, mental and social well-being and not merely the absence of disease or infirmity." Better health is central to human happiness and well-being. Determining the disease at very first is one of the challenging tasks for most of the people. In this project, we will develop a system which will predict the possible diseases according to the symptoms feed by the user i.e. the user will give the symptoms as an input to the system and as a result system will predict the disease and give it as an output. The aim is to develop the prediction system which predict the disease with help of machine learning algorithm i.e. Back Propagation Neural Network.

Keywords: Disease prediction, Back Propagation, Artificial Neural Network, Confusion Matrix

1. INTRODUCTION

Our application aims to predict the possible diseases by taking symptoms from the user as input and using data mining technique to do provisional diagnosis as practiced in telemedicine centers. Similar to that in telemedicine center, output generated by the system doesn't provide fully diagnosed results. Further, the user needs to get treatment from the doctor. For reducing the risk of disease, prediction should be done. Disease can be identified based on symptoms, physical examinations and signs of patient body. Identifying disease through several factors is a multi-layered problem which may lead to negative presumptions and unpredictable effects. So, Industries in the healthcare creates large amounts of complex data about patients, hospitals resources, disease diagnosis, electronic patient records, medical devices etc. Hence, large amount of data is a key resource to be processed and analyzed for knowledge extraction that enables support for cost-savings and decision making.

2. IMPLEMENTATON

2.1 Data Collection

Finding the datasets that can be used to trained our model for solving our problem is difficult. Since our proposed system predicate the disease based on the symptoms so, we required dataset that symptoms as features and disease name as predicated class. For our project we got dataset from UCI machine learning repository and since dataset is not in format that we want so we need to format dataset as per our requirement.

2.2 Selection of Optimal Machine Learning Algorithm:

Since our project is based on Classification problem, we end up with multiple good models to choose from. Each model will have different performance characteristics.Various performance metrics are used to evaluate different Machine Learning Algorithms. For our project we used Confusion matrix which is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of the model.Classification problem is where it is used since the output can be of two or more types of classes.

2.3 Terms Associated with Confusion Matrix:

- True Positives (TP): When the actual class of the data point was 1(True) and the predicted is also 1(True)

• True Negatives (TN): When the actual class of the data point was 0(False) and the predicted is also 0(False)

• False Positives (FP): When the actual class of the data point was 0(False) and the predicted is 1(True)

• False Negatives (FN): When the actual class of the data point was 1(True) and the predicted is 0(False).

For each algorithm the accuracy, precision and sensitivity are observed which are described as follows:

1.Accuracy: Accuracy in classification problems is the number of correct predictions made by the model over all kind predictions made.

$$\text{accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

2.Precision: This is the fraction of true positives in contrast to the overall correct results is calculated.

$$\text{Precision} = \frac{(TP)}{(TP+FP)}$$

3.Sensitivity: Sensitivity is the true positive rate and is defined as the number of positive tuples which are correctly classified.

$$\text{Sensitivity} = \frac{(TP)}{(TP+FN)}$$

The below tables, depicts the various performance metrics of the classification Decision Tree, Random Forest, Back Propagation model on the dataset.

Model	K1 Fold	K2 Fold	k3 Fold	K4 Fold
Back Propagation	93.4959349593496	97.5609756097561	77.1341463414634	85.0609756097561
Decision Tree	54.2682926829268	55.4878048780488	55.7926829268293	55.7926829268293
Random Forest	68.8008130081301	68.1910569105691	64.8373983739837	64.8373983739837

Figure 1. Performance analysis based on accuracy

Model	K1 Fold	K2 Fold	k3 Fold	K4 Fold
Back Propagation	0.8186991869918699	0.948780487804878	0.948780487804878	85.0609756097561
Decision Tree	0.5390243902439025	0.5609756097560976	0.5813008130081301	0.520650406504065
Random Forest	0.6487804878048781	0.7138211382113822	0.6626016260162602	0.6439024390243903

Figure 2. Performance analysis based on sensitivity

Model	K1 Fold	K2 Fold	k3 Fold	K4 Fold
Back Propagation	0.716573835870205	0.878176406702329	0.918315517138057	0.96936726158433
Decision Tree	0.586713380945964	0.562259306803594	0.574009855759225	0.611216591207828
Random Forest	0.628004157542957	0.702177906238043	0.643626279355847	0.623647460268178

Figure 3. Performance analysis based on precision

Figure 4. Graphical representation of accuracy

The above graph, Figure 1 &4 depicts that the Back Propagation ANN have the highest accuracy when compared to the other algorithms.

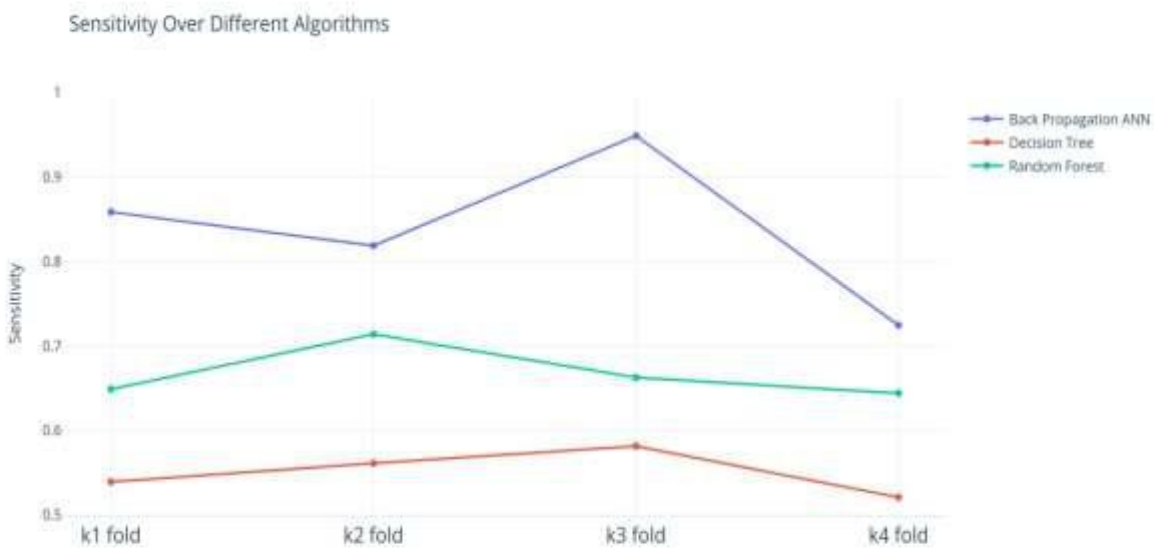

Figure 5. Graphical representation of sensitivity

The above graph, Figure 2& 5 depicts that Back Propagation ANN models have the highest sensitivity.

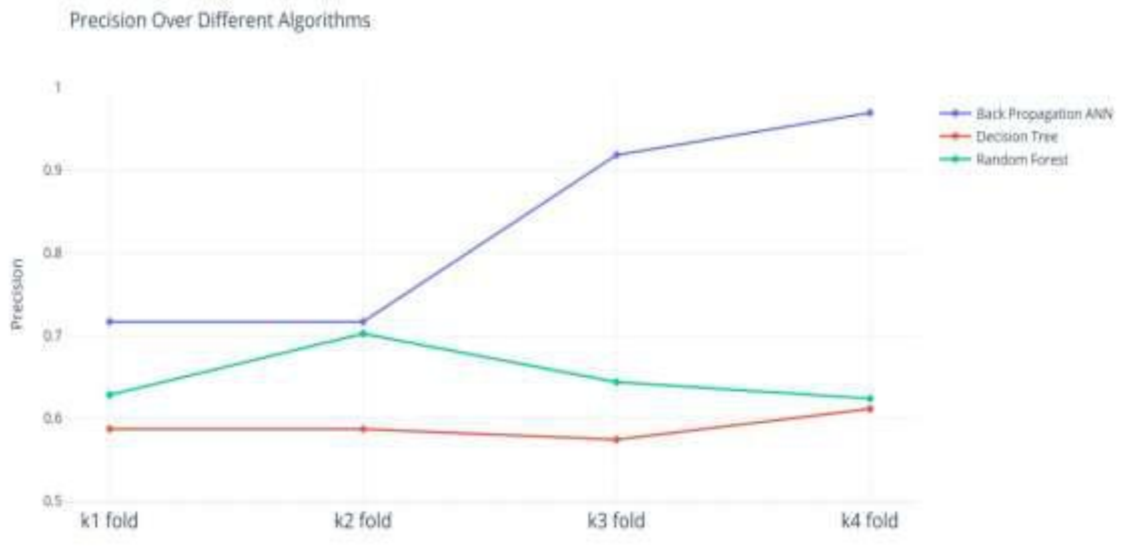

Figure 6. Graphical representation of precision

The above graph, Figure 3& 6 depicts that Back Propagation ANN models have the highest sensitivity.

2.4 Reason for Using Back Propagation ANN algorithm

From the above analysis of different classification algorithm on our dataset we conclude that the proposed method effectively predicates the diseases when compared to the other approaches. So, our proposed system is able predict the diseases with 92% accuracy using 6 hidden layers. Our prediction system gave improved results by using a higher number of hidden layers, but training model with higher hidden layer require a lot of computational power.

2.5 Selection of Optimal Hyper Parameter Value for Modeling

Before training this model, we need to pass certain hyper parameter such as learning rate, epoch.Learning rate value is highly dependent on problem at hand and in order to find suitable learning rate value for our project we explore how learning rate affect the model performance. For this we trained our neural Network with stochastic gradient descent for different value of learning rate and Activation function such as sigmoid, relu, tanh. We used cross validation test method to determine the performance of model on different parameters and we conclude that sigmoid activation function and learning rate=0.6 is optimal for our neural Network model as we got around 95% accuracy while increasing or decreasing learning rate values reduce the accuracy of model.

Number of Hidden Layer	Accuracy
3	40.9044715447155
4	85.7215447154472
5	94.8170731707317
6	95.9471766627125

Figure 7. Hidden Layers and Accuracy

At the beginning of the project, we trained data with 3 hidden layers which took 3 hours and gave 40.98 % accuracy which was not applicable to predict the disease and get optimal result. Due to unsatisfactory result, we again trained with 4 hidden layers which took 6 hours and gave 85.72 % accuracy which was applicable but not satisfactory as per our expectation and to predict the disease and get optimal result. We again trained with 5 hidden layers which took 14 hours and gave 94.81 % accuracy which was much satisfactory result. In order to obtain little more accuracy, we tried to trained with 6 hidden layers which eventually took 24 hours and gave 95.94 % which is high accuracy than previous hidden layers but it limits our computer system and to evaluate 6 hidden layers our computer system gets heated and had to resume training many times.

So, we stopped due to our system failure and limitation. At last, we concluded by training 6 hidden layers and 95.94 % accuracy to predict the disease and get optimal result. Another reason for not increasing hidden layer size from 6 is since our dataset is relatively small, increasing the hidden layer will cause our model to be over fitted to our dataset and hence reduce accuracy.

2.6 Back Propagation Algorithm implementation using Python

2.6.1 Initialize Network:

Each neuron consists of weights that are needed to be maintained. A weight for every input connection and an additional weight for the bias.Initially, we randomly set the weights of neuron.

A function creates a new neural network ready for training. It accepts three parameters, number of inputs, number of hidden layer and the number of outputs. Each neuron has n inputs + 1 weights, one for each input column in a dataset and an additional one for the bias.

2.6.2 Forward Propagate:

Output can be calculated from a neural network by propagating an input signal through each layer until the output layer outputs its values. We call this forward-propagation.

Forward propagation down into three parts:

a) Neuron Activation.
b) Neuron Transfer.
c) Forward Propagation.

a) Neuron Activation:

The first step is to calculate the activation of one neuron on given an input.

Neuron activation is calculated as:

Activation= sum(weight*input) + bias

b) Neuron Transfer:

After the neuron is activated, the activation needs to be transferred to see what the neuron output actually is. We used Sigmoid Activation function as transfer function.

It can obtain any input value and produce a number between 0 and 1 on an S-curve. It is also a function of which we can easily calculate the derivative (slope) that we will need later when back propagating error.

Output of neuron using sigmoid activation function,

Output= 1/ (1+ exp-activation)

c) Forward Propagation:

Output of each neurons is calculated and output of one neuron become input to another neuron. A functionimplements the forward propagation for a row of data from our dataset with our neural network. The output value is stored in the neuron. The outputs for a layer are collected in an array that becomes the array inputs and is used as inputs for the following layer.

2.6.3 Back Propagate Error:

The calculation of error is done between the expected outputs and the outputs forward propagated from the network. Then through the network these errors are back propagated from the output layer to the hidden layer, assigning blame for the error and updating weights as they go.

This part is broken down into section

a) Transfer derivative
b) Error backpropagation

a) Transfer derivative

Given an output value from a neuron, we need to calculate its slope. Sigmoid transfer function is used, the derivative of which can be calculated as follows:

derivative = output * (1.0 - output)

b) Error Back Propagation:

Here, we first calculate the error of each neuron, this will give error signal to propagate backward through the network.

The error of neuron can be calculated as:

Error= (expected-output) *transfer derivatives

where,

expected= Expected value of neuron

output= Output value of neuron

Error calculation is used for neuron in output layer and error calculation for hidden layers is bit different and complicated then output layer. The error for the neuron in the hidden layer is determined by the accumulation of the error signal that is back-propagated, as follows:

error = (weight k * error j) * output

Here, error j is the error signal from the jth neuron in the output layer, weight k is the weight that connects the kth neuron to the current neuron and output is the output for the current neuron.

2.6.4 Train Network:

Here we train our network with training datasets and for each row data we propagate the data, back propagate error and update the weights.

Update weights:

Afterthe calculation of errors for each neuron in the network via the backpropagation method above, they can be used to update weights.

Network weights are updated as follows:

weight = weight + learning rate * error * input

Where weight is a given weight, learning rate is a hyper parameter we specify, error is the error calculated by the backpropagation procedure for the neuron and input is the input value that caused the error. For the updating of the bias weight same procedure can be used, except there is no input term, or input is the fixed value of 1.0.

3. CONCLUSION

Our system takes symptoms as input, processes them and provide provisional diagnosed diseases as results to users.Our proposed system of disease prediction with appropriate diagnosis has been framed up using Artificial Neural Network. It is proven from the results that the proposed method, back propagation artificial neural network effectively predicts accuracy around 95.42% and hence, chosen as an optimal algorithm for our project.

4. ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals, especially our project guide Dr. Prasanth T. who have supported us and are responsible for the successful completion of this project work.We take this opportunity to express our sincere gratitude to Faculty of Engineering & Technology, Jain (Deemed to be University) for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.

5. REFERENCES

[1] Prof. Shubhangi Patil, Shraddha Subhash Shirsath, (2018),International Journal of Innovative Research in Science, Engineering and Technology, “Disease Prediction Using Machine Learning Over Big Data”, **Vol. 7, Iss. 6**, pp 6752-6757

[2] Nikita Kamble, ManjiriHarmalkar, Manali Bhoir, Supriya Chaudhary (2017), International Journal of Scientific Research in Computer Science, Engineering and Information Technology, “Smart Health Prediction System Using Data Mining”, **Vol. 2, Iss. 2**, pp 1020-1025

[3] Dipa Barad, Priyanka Raval, Nilesh Borisagar, (2017), Proceedings of International Conference on Communication and Networks, “Chronic kidney disease prediction using Back Propagation Neural Network Algorithm”,pp 295-303

[4] Rafiah Awang, SellappanPalaniappan, (2008), International Journal of Computer Science and Network Security (IJCSNS), IEEE, “Intelligent Heart Disease Prediction System Using Data Mining Technique”,**Vol.8 No.8**

[5] Arti Gupta, Maneesh Shreevastava, (2011), International Journal of Emerging Technology and Advanced Engineering (IJETAE), “Medical Diagnosis using Back Propagation Algorithm”, **Volume 1, Issue 1**

[6] B Gomathy, M.A.Nishara Banu, (2013), International Journal of Technical Research and Applications, “Disease Predicting System Using Data Mining Techniques”, **Vol. 1, Iss. 5**, pp 41-45

[7] Mohammad Ismail, V. Harsha Vardhan, V. Aditya Mounika, K. Surya Padmini, (2019), International Journal of Innovative Technology and Exploring Engineering (IJITEE), “An Effective Heart Disease Prediction Method using Artificial Neural Network”, **Volume-8 Issue-8**

[8] Ranjit Shrestha, Asst. Prof. Jyotir Moy Chatterjee(2019), Lord Buddha Education Foundation, Kathmandu, Nepal, “Heart Disease Prediction System Using Machine Learning”, **Volume 1, Issue 2**

[9] Mrs. K. Uma Maheswari Assistant Professor, Ms. J. Jasmine PG Scholar, (2017), International Journal of Engineering Research & Technology (IJERT),“Neural Network based Heart Disease Prediction”, Special Issue – 2017, ISSN: 2278-0181

[10] Riddhi Kasabe, Prof. Dr. Geetika Narang, (2020), International Journal of Engineering Research & Technology (IJERT), “Heart Disease Prediction using Machine Learning”, **Vol. 9 Issue 08**