**Project Requirement and Specification**

**on**

**Heart Disease Prediction System Using**

**Machine Learning**

# (CSE V Semester MOOC Seminar SCS-501)

**2020-2021**



**Submitted to: Submitted by:**

Ms. Rishika Yadav Mr. Shubham Rana

(CC-CSE-A-V-Sem) Roll. No.: 1018604

**Guided by:** CSE-A-V-Sem

Mr. Kiran Gavali Session: 2020-2021

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GRAPHIC ERA HILL UNVERSITY, DEHRADUN**

# 1.1 About Project

The health care industries collect huge amounts of data that contain some hidden information, which is useful for making effective decisions. For providing appropriate results and making effective decisions on data, some advanced data mining techniques are used. In this study, an effective heart disease prediction system (EHDPS) is developed using neural network for predicting the risk level of heart disease. The system uses 15 medical parameters such as age, sex, blood pressure, cholesterol, and obesity for prediction. The EHDPS predicts the likelihood of patients getting heart disease. It enables significant knowledge, eg, relationships between medical factors related to heart disease and patterns, to be established. We have employed the multilayer perceptron neural network with backpropagation as the training algorithm. The obtained results have illustrated that the designed diagnostic system can effectively predict the risk level of heart diseases.

# 1.2 Requirement of Project

## 1.2.1 Problem Statement

In this project we aim to predict heart diseases using data set of health care industries.

## 1.2.2 Hardware Requirement

The whole project is implemented on HP14s, specifications of the laptop Intel(R) Celeron(R) N4000 CPU @ 1.10GHz, 8.00 GB RAM, 64-bit operating system.

## 1.2.3 Software Requirement

The project is implemented using Jupyter Lab.

# 1.3 Methodology

The experiment was carried out on a publicly available database for heart disease. The dataset contains a total of 303 records that were divided into two sets, training set (40%) and testing set (60%). A data mining tool named Weka 3.6.11 was used for the experiment. Additionally, multilayer perceptron neural network (MLPNN) with backpropagation (BP) was used as the training algorithm.

### **MLPNN**

MLPNN is one of the most significant models in artificial neural network. The MLPNN consists of one input layer, one or more hidden layers and one output layer.[3](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/#b3-ijn-13-121) In MLPNN, the input nodes pass values to the first hidden layer, and then nodes of first hidden layer pass values to the second and so on till producing outputs as shown in [Figure 1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/figure/f1-ijn-13-121/).

[[An external file that holds a picture, illustration, etc.
Object name is ijn-13-121Fig1.jpg](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/figure/f1-ijn-13-121/)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/figure/f1-ijn-13-121/" \t "figure)

[Figure 1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/figure/f1-ijn-13-121/)

Multilayer perceptron neural network.

### **BP network**

The BP algorithm has served as a useful methodology to train multilayer perceptron for a wide range of applications.[4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/#b4-ijn-13-121) The BP network calculates the difference between real and predicted values, which is circulated from output nodes backwards to nodes in previous layer. The BP learning algorithm can be divided into two phases, propagation and weight update.[4](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/#b4-ijn-13-121)

First, this learning algorithm provides training data to the network and compares the actual and desired outputs. Then, it calculates the error in each neuron. Based on this, the algorithm calculates what output should be for each neuron and how much higher or lower output must be adjusted for desired output and finally adjusts the weights. The overall process is done to improve weights during processing.

# 1.4 Expected Result

In order to predict the probability of patients having heart disease, a confusion matrix ([Table 1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/table/t1-ijn-13-121/)) was created, where A denotes patients with heart disease, and B denotes patients with no heart disease.

### **Table 1**

A confusion matrix

|  | **A (patients with heart disease)** | **B (patients with no heart disease)** |
| --- | --- | --- |
| A (patients with heart disease) | TP | FN |
| B (patients with no heart disease) | FP | TN |

**Abbreviations:** TP, true positive; FN, false negative; FP, false positive; TN, true negative.

A confusion matrix contains information about real and predicted classifications done by a classification system. The data in the matrix are evaluated to know the performance of such systems.

The confusion matrix contains the following four entries:

* \*TP (true positive): The number of records classified as true while they were actually true.
* \*FP (false positive): The number of records classified as true while they were actually false.
* \*FN (false negative): The number of records classified as false while they were actually true.
* \*TN (true negative): The number of records classified as false while they were actually false.

The overall process of effective heart disease prediction system (EHDPS) is based on the following three steps:

1. Data collection
2. Data pre-processing and
3. The classification of data.