International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.82.003

A Comprehensive Approach to Predict Heart Diseases using Data Mining

Amanpreet Kaur
Department of Computer Science and Engineering
GTBIT, New Delhi, Delhi, India

Abstract- The avaibility of huge data available within the healthcare systems has lead to the use of data mining to extract useful information. Analysis of Heart diseases is one of the major research in data mining. Disease diagnosis is one of the applications where data mining tools have shown successful results. In this paper, data mining techniques such as NaïveBayes, Multilayer perceptron (MLP), Random forest and Adaboost have been used to perform early prediction of heart diseases and absolute relative error has been found out in each case.

Keywords – Data Mining, Heart Disease Diagnosis, Classification algorithms.

I. INTRODUCTION

A widely recognized formal definition of data mining is “Data mining is the non- trivial extraction of implicit previously unknown and potentially useful information about data”[1]. Data mining can be defined as extraction of large dataset and its analysis from different perspectives to retrieve useful information about data. Data are facts, numbers, or text which can be processed by a computer. It renders a number of techniques to detect hidden patterns from data that are difficult to trace with traditional statistical methods [2,3].

Data mining in healthcare is an emerging field that emphasis on prognosis and a deeper understanding of medical data. Healthcare applications in data mining include prevention of hospital errors, early sensing and cure of diseases, preventable hospital deaths etc. Number of data mining techniques are being used by researchers in the diagnosis of several diseases such as diabetes, stroke ,cancer,skin and heart diseases. So it is observed that, the data mining could help in the identification or the prediction of high or low risk heart diseases[4,5,6].

The Data mining process includes the following steps :

- Data Collection.
- Data Pre-processing.
- Clustering of collected data.
- Estimate the model.
- Interpret the model and draw conclusion.

The remaining paper is organized into sections. Section 2 describes the various data mining classification methods: NaiveBayes, Multilayer perceptron, Random Forest and AdaBoost. Section 3 includes the details of the dataset used. The result of the analysis implemented in WEKA is shown in Section 4. A brief conclusion is presented in Section 5.

II. DATA MINING CLASSIFICATION METHODS

A. NaïveBayes-

A Naive Bayes model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Bayesian algorithm uses efficient approach to classify data sets due to its simplicity, robustness and with strong independent assumptions [8]. This technique presumes that attributes of a class are neutral. This results that presence of one feature does not affect other features in classification tasks which makes this approach more efficient [9]. The algorithm uses Bayes theorem and assumes all attributes to be independent given the value of the class variable. Naïve Bayes classifiers can strongly be developed by requiring a small amount of training data to estimate the parameters necessary for classification [11].

http://dx.doi.org/10.21172/ijiet.82.003

B. Multilayer Perceptron-

A Multilayer Perceptron is a feedforward artificial neural network model that maps sets of input data on to a set of appropriate output. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. The network consist of hidden neurons that enables the network to learn complexities by extracting useful features from the input patterns. MLP utilizes back propagation for training the network This allows the network to combine the inputs in more complex ways and in turn provide a richer capability in the functions they can model. Non-linear functions like the logistic also called sigmoid function [10,11].

C. Random Forest-

Random Forest Trees are based on a number of prediction trees that are less tolerant to noise compared to “Adaboost” and utilize random selection of features in splitting the trees. It is one of the most accurate algorithm and works well on large data bases. A Random Forest is a collection of CART-like trees for growing, combination, testing and Post-processing [8].

Random forest uses two mechanisms:

1. build a group of trees via bagging with replacement (bootstrap) and a choosing an unspecified features at each node of tree. The first mechanism means that any example selected from the training set can be selected again. Each tree is grown using the obtained bootstrap sample.
2. The second mechanism is to elect a small proportion of features and further break using the best feature from this set. The number of features to be selected within the algorithm are fixed within the execution [12,13].

D. AdaBoost-

AdaBoost, is also know as "Adaptive Boosting", is a machine learning algorithm used with several types of learning algorithms to improve their performance. The first successful boosting algorithm developed for binary classification was Adaboost that can be used to boost the performance of any machine learning algorithm. Boosting consists in combining low quality classifiers with a voting scheme to produce a classifier better than any of its components. The most common version of Boosting is called AdaBoost. Adaboost has been compared to greedy back fitting of extended additive models in logistic regression problems[14]. The Adaboost algorithm has been applied to learn fuzzy rules in classfication problems, and other backfitting algorithms to learn fuzzy rules in modeling problems but, up to our knowledge, there are not previous works that extend the Logitboost algorithm to learn fuzzy rules in classification problems.

III. DATASET USED

The data set used for heart diseases prediction has been taken from UCI Machine Learning Repository. The Cleveland dataset has been used to predict heart disease. This database contains 76 attributes, but only a subset of 14 attributes has been used for prediction. Some of the attributes used are age , sex, cp (chest pain), trestbps (resting blood pressure), chol (cholestoral), fbs (fasting blood sugar), restecg (resting electrocardiographic), thalach(maximum heart rate achieved), exang(exercise induced angina), num etc. The data is available at https://archive.ics.uci.edu/ml/datasets/Heart+Disease.

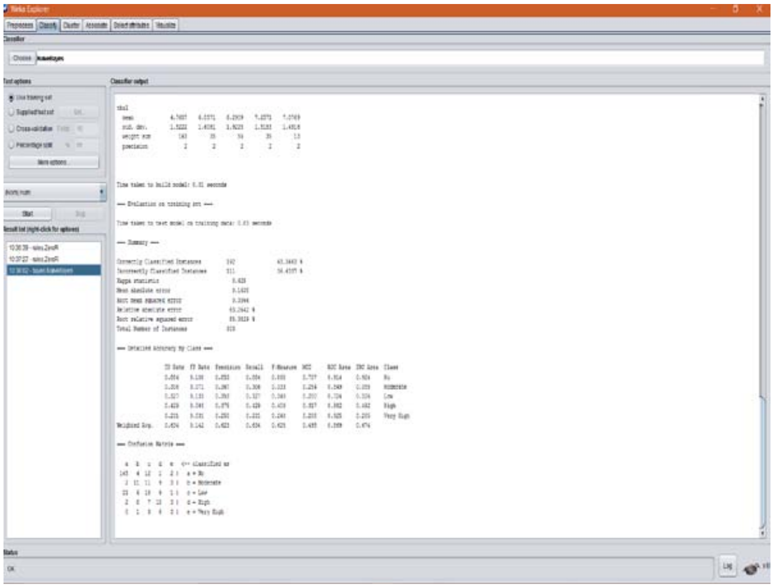
IV. RESULTS

The heart disease prediction on the Cleveland dataset has been carried out using (Table1). The average absolute error, correctly classified instances and incorrectly classified instances have been found out in each case.

http://dx.doi.org/10.21172/ijiet.82.003

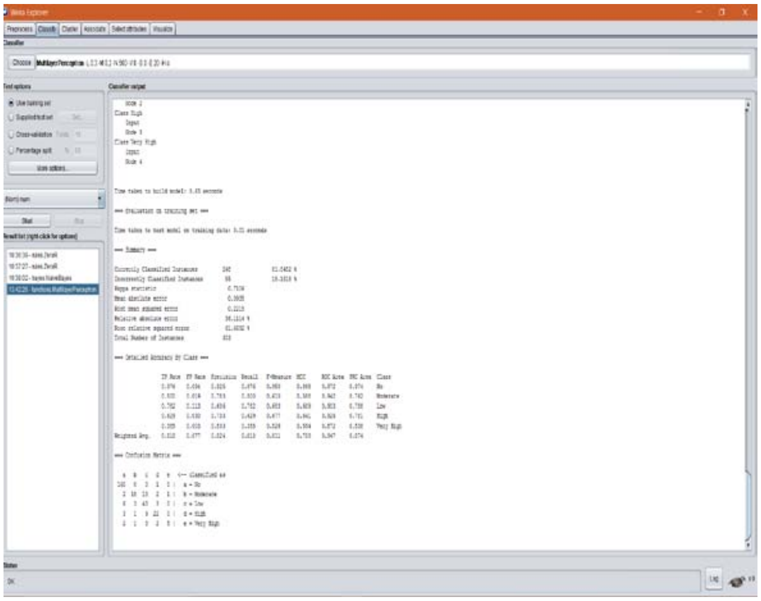
S.No.	**Data Mining Classifier**	**Data using Training Dataset**		
		Correctly Classified Instances	Incorrectly Classified Instances	Relative Absolute Error
1.	**Naïve Bayes**	63.366%	36.633%	63.264%
2.	**Multilayer Perceptron**	81.848%	18.151%	36.111%
3.	**Random Forest**	100%	0%	28.788%
4.	**AdaBoost**	54.125%	45.876%	121.035%

Table 1. Classification for Heart Survival Dataset using taining Dataset.



```
=== Summary ===

Correctly Classified Instances         192               63.3663 %
Incorrectly Classified Instances       111               36.6337 %
Kappa statistic                          [illegible]
Mean absolute error                      [illegible]
Root mean squared error                  [illegible]
Relative absolute error                 63.2642 %
Root relative squared error              [illegible]
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
                                                                                          No
                                                                                          Moderate
                                                                                          Low
                                                                                          High
                                                                                          Very High
Weighted Avg.

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
                     |   a = No
                     |   b = Moderate
                     |   c = Low
                     |   d = High
                     |   e = Very High
```

Figure1. NaïveBayes

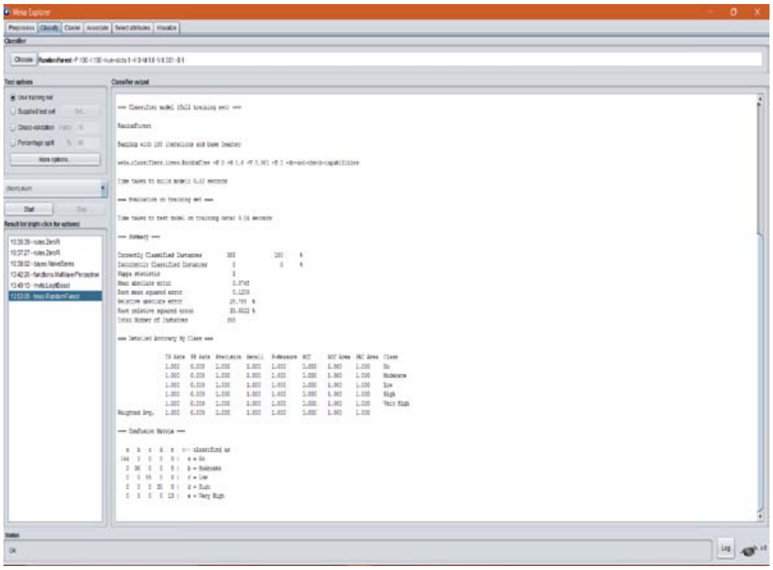


Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Result list:
10:30:36 - rules.ZeroR
10:57:27 - rules.ZeroR
10:58:02 - bayes.NaiveBayes
13:42:26 - functions.MultilayerPerceptron

```
    Node 2
Class High
    Input
    Node 3
Class Very High
    Input
    Node 4


Time taken to build model: [illegible] seconds

=== Evaluation on training set ===

Time taken to test model on training data: [illegible] seconds

=== Summary ===

Correctly Classified Instances       [illegible]         [illegible] %
Incorrectly Classified Instances     [illegible]         [illegible] %
Kappa statistic                       [illegible]
Mean absolute error                   [illegible]
Root mean squared error               [illegible]
Relative absolute error              [illegible] %
Root relative squared error          [illegible] %
Total Number of Instances            [illegible]

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 [illegible]                                                                   No
                 [illegible]                                                                   Moderate
                 [illegible]                                                                   Low
                 [illegible]                                                                   High
                 [illegible]                                                                   Very High
Weighted Avg.    [illegible]

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 [illegible]             |   a = No
 [illegible]             |   b = Moderate
 [illegible]             |   c = Low
 [illegible]             |   d = High
 [illegible]             |   e = Very High
```

Figure2. Multilayer Perceptron

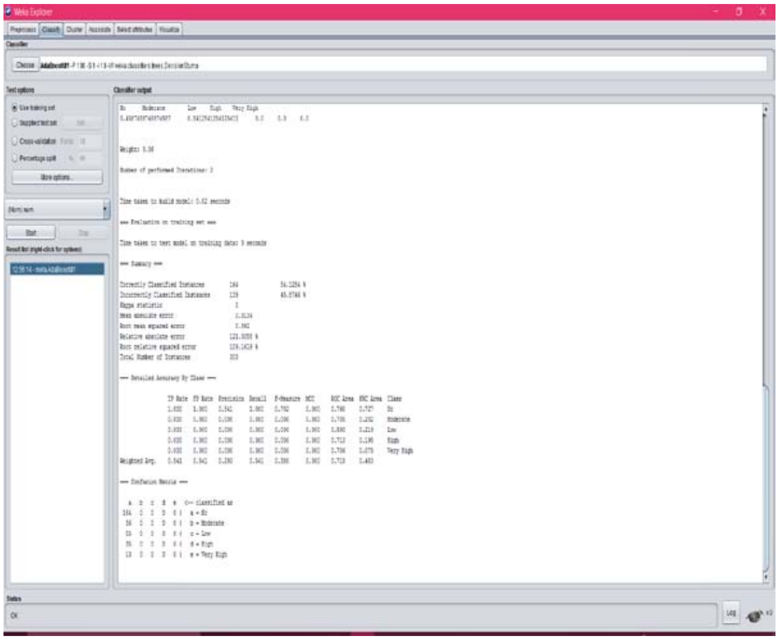


Classifier: RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Result list:
10:30:36 - rules.ZeroR
10:57:27 - rules.ZeroR
10:58:02 - bayes.NaiveBayes
13:42:26 - functions.MultilayerPerceptron
13:49:12 - meta.LogitBoost
13:53:35 - trees.RandomForest

```
=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: [illegible] seconds

=== Evaluation on training set ===

Time taken to test model on training data: [illegible] seconds

=== Summary ===

Correctly Classified Instances       [illegible]         100      %
Incorrectly Classified Instances       0                   0      %
Kappa statistic                        1
Mean absolute error                    [illegible]
Root mean squared error                [illegible]
Relative absolute error               [illegible] %
Root relative squared error           [illegible] %
Total Number of Instances            [illegible]

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     No
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Moderate
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Low
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     High
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Very High
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 144   0   0   0   0 |   a = No
   0  36   0   0   0 |   b = Moderate
   0   0 [illegible] 0   0 |   c = Low
   0   0   0  35   0 |   d = High
   0   0   0   0  13 |   e = Very High
```

Figure3. Random Forest



AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump

Test options: ● Use training set ○ Supplied test set ○ Cross-validation Folds 10 ○ Percentage split % 66

Result list: 12:56:14 - meta.AdaBoostM1

```
No        Moderate        Low    High    Very High
0.[illegible]     0.[illegible]      0.0    0.0    0.0


Weights: 0.36

Number of performed Iterations: 2


Time taken to build model: 0.02 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         164               54.1254 %
Incorrectly Classified Instances       139               45.8746 %
Kappa statistic                          0
Mean absolute error                      0.3134
Root mean squared error                  0.392
Relative absolute error                121.3058 %
Root relative squared error            139.1419 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.541      1.000    0.702      0.000    0.768     0.727     No
                 0.000    0.000    0.000      0.000    0.000      0.000    0.706     0.252     Moderate
                 0.000    0.000    0.000      0.000    0.000      0.000    0.690     0.219     Low
                 0.000    0.000    0.000      0.000    0.000      0.000    0.712     0.196     High
                 0.000    0.000    0.000      0.000    0.000      0.000    0.736     0.079     Very High
Weighted Avg.    0.541    0.541    0.293      0.541    0.380      0.000    0.733     0.483

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 164   0   0   0   0 |   a = No
  55   0   0   0   0 |   b = Moderate
  36   0   0   0   0 |   c = Low
  35   0   0   0   0 |   d = High
  13   0   0   0   0 |   e = Very High
```

Status: OK

Figure 4. AdaBoost

V. CONCLUSION AND FUTURE SCOPE

Various classification techniques were compared that includes NaiveBayes, Multilayer perceptron, Random Forest and AdaBoost on heart survival dataset that shows that Random forest method surpass the remaining methods. Absolute relative error for Random Forest is minimum as compared to all other algorithms and Adaboost algorithm presents the maximum error. More classification algorithms can be applied to more data sets for early prediction of diseases in future applications.

REFERENCES

[1] K.Sudhakar and Dr. M. Manimekalai, “Study of Heart Disease Prediction using Data Mining.” vol 4, Issue 1, January 2014.

[2] Mai Shouman1, Tim Turner1, Rob Stocker1, “USING DATA MINING TECHNIQUES IN HEART DISEASE DIAGNOSIS AND TREATMENT” , 173978-1-4673-0484-9/12/$31.00 c □2012 IEEE.

[3] Muhamad Hariz Muhamad Adnan, Wahidah Husain and Nur'Aini Abdul Rashid, “Data Mining for Medical Systems: A Review”, Proc. of the Int ernational Conference on Advances in Computer and Information Technology- ACIT 2 0 12.

[4] Aqueel Ahmed, Shaikh Abdul Hannan, ” Data Mining Techniques to Find Out Heart Diseases: An Overview.” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-1, Issue-4, September 2012.

[5] Sellappan Palaniappan and Rafiah Awang, “ Intelligent Heart Disease Prediction System Using Data Mining Techniques”, 978-1-4244-1968-5/08/$25.00 ©2008 IEEE.

[6] Seyed Abbas Mahmoodi, Kamal Mirzaie and Seyed Mostafa Mahmoudi, “A new algorithm to extract hidden rules of gastric cancer data based on ontology”, Mahmoodi et al. SpringerPlus (2016).

[7] Jyoti Soni, Ujma Ansari, Dipesh Sharma and Sunita Soni, “Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction”, International Journal of Computer Applications (0975 – 8887) Volume 17– No.8, March 2011.

[8] DSVGK Kaladhar, B. Chandana and P. Bharath Kumar, “Predicting Cancer Survivability Using Classification Algorithms”, International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 2, No. 2, April 2011.

[9] Yugal kumar and G. Sahoo, “Analysis of Bayes, Neural Network and Tree Classifier of Classification Technique in Data Mining using WEKA”, ResearchGate Conference Paper May 2012.

[10] Gaurang Panchal, Amit Ganatra, Y P Kosta and Devyani Panchal, “Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers”, International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011.

[11] Poongodi S, Radha N, “Classification of user Opinions from tweets using Machine Learning Techniques”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, September 2013.

[12] Piero Bonissone, Jose M.Cadenas, M.Carmen Garrido and R. Andres Diaz-Valladares, “ A fuzzy random forest”, International Journal of Approximate Reasoning, ©2010 Elsevier.

[13] Dr. N. Venkatesan and G. Priya, " Study and Implementation of Random Forest Algorithm with WEKA Too", International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-4, Issue-5) May2015.

[14] Jos´e Otero · Luciano S´anchez, “Induction of descriptive fuzzy classfiers with the Logitboost algorithm ”, Published online: 12 October 2005 © Springer-Verlag 2005.