**Performance comparison of SVM, MLPClassifier, K-NN and DecisionTree machine learning algorithms for the diagnosis of heart disease.**

**Submitted To: Silvia Ahmed**

**Submitted By: Shakil Ahmed Sumon, Sazid Rahman Shimanto**

**Abstract:**

In this paper, we used the statlog heart disease dataset from University of California Irvine repositary to evaluate the performance of SVM, MLPClassifier, K-NN and DecisionTreeClassifier. The dataset takes a decision whether the patient is prone to heart disease based on some medical data recorded on the patient. We explained the skeleton of the dataset and listed its features. We discussed the types of algorithms used in the process of classification. We observed that the performance of MLPClassifier and SVM are the best on this dataset in terms of accuracy. We evaluated couple of others parameters to determine the performance of the algorithms and found out that MLPClassifier has edge over the other algorithms. We analyzed the algorithms and detected why one algorithm performs better than others on some specific tasks.

**Objective:**

Diagnosing heart disease is a pretty confusing and lengthy process. People often misjudges the symptoms and hence bear severe consequences. It is a major cause of mortality both in developed and developing countries. We designed a classification method for determining heart disease and did a comparative study of four classification algorithms to figure out the most accurate algorithm on the dataset used.

**Related Works:**

Previous works on heart disease prediction was based on different data mining [1] [2] and machine learning algorithms [3]. In their work on [1], researches built an intelligent disease prediction system. They used three algorithms; naïve Bayes, decision tree and neural network and gave a pretty brief comparison of the three algorithms used. In [4], they did a comparative study of Decision tree, Bayesian classification, KNN, neural network and classification based on clustering. They reveal that decision tree outperforms all other and sometimes Bayesian classification has same accuracy as decision tree.

What we did here was that, we dragged SVM into the game and did multiple simple comparison between them on various prediction tasks using scikit-learn [5].

**Dataset:**

We used the heart disease (statlog) dataset from UCI machine learning repository [6]. The dataset has 13 attributes and 270 observations. We tried to feed as much data as we can to train the system designed. The attributes of the dataset are:

1. Age

2. Sex

3. Chest pain type (4 values)

4. Resting blood pressure

5. Serum cholesterol in mg/dl

6. Fasting blood sugar > 120 mg/dl

7. Resting electrocardiographic results (values 0, 1, and 2)

8. Maximum heart rate achieved

9. Exercise induced angina

10. Old peak = ST depression induced by exercise relative to rest

11. The slope of the peak exercise ST segment

12. Number of major vessels (0-3) colored by fluoroscopy

13. Thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

Value to be predicted:

1🡪 absence of heart disease

2🡪 present of heart disease

**Theoretical analysis of Algorithms:**

Multi-Layer Perception (MLP)

This is a neural network algorithm which uses backpropagation technique to learn and apply the learning. It uses neuron like layers to train the system. There are three types of layers; input layer, output layer and hidden layers in between them. Number of hidden layers are variable.

Suppose we have a training data set (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) where x_i \in \mathbf{R}^n and y_i \in \{0, 1\} . Let’s say we have a MLP that has a lone hidden layer with one neuron.

We have a function f(x) = W_2 g(W_1^T x + b_1) + b_2 that needs to be learnt by this MLP. W1, w2🡪 weights of the input and hidden layers. b1, b2🡪 bias added to hidden and output layer.

For our own binary classification model, this function f(x) passes through another function g(z)=1/(1+e^{-z}) which is called logistic function. It output values between 0 and 1. A threshold set to 0.5; values greater than this will assign to positive classes and rest will assign to negative class. In our model we used 500 hidden layers to get the optimum result.

Support vector machine (SVM)

SVM is a kernel based algorithm where the classifier is determined by a separating hyperplane. In a 2D space, the hyperplane is a line separating the classes.

We use the linear kernel in our model but in case the hyperplane is non-linear then a transformation is needed by adding extra axis. SVM in some cases is similar like single layer neural network with one important difference which is the introduction of support vectors in SVM. Support vectors are the input features those play an influential part in determining the hyperplane.

K-Nearest Neighbors (K-NN)

K-NN classifier is to classify unlabeled observations by assigning them to the class of the most similar labeled examples [7]. This algorithm is an instance based learner.

We have a parameter k here which is the number of neighbors specified. Suppose we set k to 5 and we have 2 classes to predict. We will look for the k nearest neighbor and will count the majority assigned to a certain class. Let’s say our class1 has 5 instances and class2 has 3 instances. Now if a new point comes to be labeled then the k-NN will assign it to class1.

We will use the Euclidian distance to get the distance of the new point from the neighbors which can be determined by the equation

D(p,q)=√(p1-q1)2+………..(pn-qn)2 [7]

Decision Tree

Decision tree classifier is non-parametric in the sense that nothing need to be set as the k in K-NN. In this tree model the leaves represent the labels to be predicted. The model predicts the label of the target variable by some decision rules which it created during the training. It splits the data in some non-overlapping regions and make the decision rules by observing them. There are quite a few methods for splitting the trees.

**Results and Findings:**

We trained our model by the training data separated from the dataset and then gave the reserved testing data to the model to predict.

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| MLPClassifier | 90% |
| SVM | 85% |
| K-NN | 70% |
| DecisionTreeClassifier | 65% |

Table 1: Accuracy of algorithms

We see that MLPClassifier has the highest accuracy and then SVM, k-NN and DecisionTreeClassifier in the numerical order.

However, the confusion matrix gives us some interesting data to evaluate. The first data of first column represents TP (true positive), second data represents FN (false negative); first data of second column represents FP (false positive) and the second data represents TN (true negative). Given 20 data to predict the model does the following:

|  |  |  |
| --- | --- | --- |
| Real/Pred | Pred True | Pred False |
| Real True | 8 | 2 |
| Real False | 0 | 10 |

Figure 1: Confusion Matrix of MLPClassifier

|  |  |  |
| --- | --- | --- |
| Real/Pred | Pred True | Pred False |
| Real True | 7 | 3 |
| Real False | 0 | 10 |

Figure 2: Confusion Matrix of SVM

|  |  |  |
| --- | --- | --- |
| Real/Pred | Pred True | Pred False |
| Real True | 7 | 3 |
| Real False | 3 | 7 |

Figure 3: Confusion Matrix of k-NN

|  |  |  |
| --- | --- | --- |
| Real/Pred | Pred True | Pred False |
| Real True | 7 | 3 |
| Real False | 4 | 6 |

Figure 4: Confusion Matrix of Decision Tree Classifier

From the above confusion matrices we calculated the TPR (true positive rate) and FPR (false positive rate) by the following equations:

1. TPR= TP/Total true values;
2. FPR= FP/Total False values.

|  |  |  |
| --- | --- | --- |
| Algorithm | TPR | FPR |
| MLPClassifier | 100% | 16.67% |
| SVM | 100% | 23% |
| K-NN | 70% | 30% |
| DecisionTree | 63% | 33.33% |

Table 2: TPR-FPR comparison

What we find is that, in terms of TPR, MLP and SVM both performs overwhelmingly well. If a person is suffering from heart disease, everytime these algorithms can predict it accurately in this dataset. Out of the rest two, k-NN performs better than Decision tree.

We plotted these values from Table 2 to produce Receiver Operating Characteristic (ROC) curve.

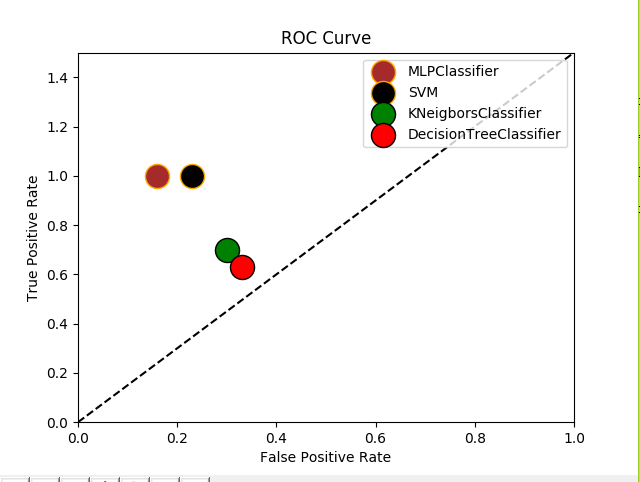


Figure 5: ROC Curve

**Reasons of Varying Performance:**

There is no surprise that SVM and MLPClassifier both shows the same level of accuracy because both of them are neural network algorithm. SVM is a classic example of local neural network. It is a two layer neural network employing hidden layer of radial units and one output neuron. [8] We do not feel SVM like neural network because we do not have to deal with the hidden layers here; kernel function handles that for us. The most distinctive fact about SVM is that the learning task is reduced to quadratic programming by introducing the so-called Lagrange multipliers. [8] All operations in learning and testing modes are done in SVM using kernel functions satisfying Mercer conditions [9]. The slight difference SVM and MLPClassifier has that SVM formulates the problem into quadratic optimization task [8].

SVM and k-NN both are parametric learning algorithms yet they show considerable difference in performance in terms of TPR. K-NN is prone to overfitting and we did not perform any handling tasks to maintain the level-playing field for the algorithms. If standard scaling is done and suitable parameters are chosen both the algorithms performs pretty similar. Kernel function and parameter C are the scaling factor of SVM. Linear kernel is less prone to overfitting [10] what might give SVM the edge over k-NN.

Decision tree performs the worst in terms of TPR and FPR in this dataset. In some previous works, Decision tree outperforms neural network and k-NN but in this dataset lack of feature scaling leads its way to overfitting, hence hindering its way towards achieving higher accuracy.

**Conclusion:**

In our work, we show that when the dataset has less features and samples neural network algorithms like SVM and MLPClassifier performs exceptionally well which opposes the previous idea that SVM is better off with large scale datasets. On the other hand, k-NN and Decision tree algorithms are simple and used for simple datasets which are less prone to overfitting. Our findings differs with that idea too. In this simple dataset, both of the algorithms perform worse than the neural network. Our findings suggest that though both SVM and MLPClassifier hav a huge edge in accuracy over k-NN and Decision Tree classifier MLPClassifier should be used in this dataset as it is more consistent in its predicting.

**References:**

1. Intelligent heart disease prediction system using different data mining techniques.
2. An analysis of heart disease prediction using different data mining techniques.
3. Prediction model using EHR data: challenges, strategies and a comparison of machine learning approaches.
4. Predictive data mining for medical diagnosis: an overview of heart disease prediction.
5. Scikit- learn: a python tool for machine learning.
6. Stalog dataset of UCI.
7. Introduction to machine learning: k-nearest neighbor.
8. MLP and SVM networks: a comparative study.
9. V. Vapnik, “statistical learning theory”
10. A comparative study of the SVM and k-NN machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals.