A Machine Learning Approach on Classifying Orthopedic Patients Based on Their Biomechanical Features

SUBMISSION DATE: 05.04.18

SUBMITTED BY:

Kamrul Hasan (13101102)
Safkat Islam (13301099)
Md. Mehfil Rashid Khan Samio (13101029)

Department of Computer Science and Engineering

Supervisor:

Amitabha Chakrabarty, Ph.D
Assistant Professor
Department of Computer Science and Engineering

Declaration

We, hereby declare that this thesis is based on results we have found ourselves. Materials of work from researchers conducted by others are mentioned in references. This thesis neither in parts nor as a whole have been submitted previously by anyone of any institute or university for the award of any degree.

Signature of Supervisor

Amitabha Chakrabarty, Ph.D
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Signature of Authors

Kamrul Hasan (13101102)

Safkat Islam (13301099)

Md. Mehfil Rashid Khan Samio (13101029)

ABSTRACT

A person's orthopedic health condition can be detected from his biomechanical features. Application of machine learning algorithms in medical science is not new. Different algorithms are applied to detect diseases and classify patients accordingly. This paper aims to assist specialists to predict the type of orthopedic disease. In this paper we have applied various machine learning algorithms to find out which one works most accurately to detect and classify orthopedic patients. Each of the patients in the dataset is represented by six biomechanical attributes derived from the shape and orientation of pelvis and lumbar spine. We performed our operation in two stages and got an average accuracy of more than 90 percent for most of the algorithms, whereas Decision Tree (DT) algorithm stood out from the rest providing 99% accuracy.

Keywords— Algorithm, Biomechanical features, Classification, Disease, Machine Learning, Orthopedic, Prediction.

ACKNOWLEDGEMENT

First of all, we are so much grateful to ALLAH and to our parents. Then, we are grateful to our supervisor, Dr. Amitabha Chakrabarty, for his immense support throughout the work. He made us to get out of our comfort zones and to push the limit. Without his guidance, it would have been impossible to get introduced to the amazing research prospect of Machine Learning.

Again, we admire the patronage and moral support extended with love, by our parents as well as our friends. They helped us with their direct or indirect suggestions which aided in achieving our goal. We would also like to acknowledge the assistance we received from numerous resources over the Internet especially from fellow researchers' work.

Nevertheless, we would like to express our gratitude to Department of Computer Science and Engineering, BRAC University and our teachers for helping us with all the necessary support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

ADB– Adaptive Boosting

ANN– Artificial Neural Network

DT– Decision Tree

GP– Gaussian Process

K-NN – K-Nearest Neighbor

LR – Logistic Regression

MLP – Multi-Layer Perceptron

NB – Naïve Bayes

PNN – Probabilistic Neural Network

QDA – Quadratic Discriminant Analysis

RF – Random Forest

SVM – Support Vector Machine

CHAPTER 1

INTRODUCTION

Machine learning has been implemented in various medical fields and proven to be very accurate in classifying and predicting diseases. Use of machine learning is spreading widely with the growth of medical data in medical field to improve medical service and diagnosis of diseases. Since every medical study is including these techniques, we tried to implement it in the field of orthopedics. We hope, in future our study will help doctors to predict orthopedic diseases in a faster and easier way.

1.1 Motivation

In the modern era of technology, almost every work space has recognized the blessings of machine learning techniques. Through these techniques, works have become easier and faster. Machine learning techniques have been used in medical fields for a very long time and proven to be very successful. Different sectors of medical field has implemented different machine learning algorithms for

numerous purposes. Disease prediction, classification, medicine and cure recommendations for patient etc. are being done by algorithms very extensively. Seeing all the sectors of medical field using machine learning we get motivated to implement similar types of machine learning system in orthopedic to classify patients from their biomechanical characteristics.

1.2 Objectives

Since every medical field is now practicing machine learning techniques, we also want to apply a number of techniques to enhance the classification process by finding out best performing algorithms. This paper shows the classification of orthopedic patients based on the dataset that have been used. Our paper aims to assist specialists to identify the type of orthopedic diseases faster without their interposition.

1.3 Thesis Orientation

Chapter 1 is **INTRODUCTION**. The Motivation and Objectives of the thesis are described here.

Chapter 2 is **LITERATURE REVIEW**. This chapter consists of “Background” which defines the problem space of our thesis. Again, “Literature Review” indicates our information collection repository. This chapter also consists of “Supervised Learning” which is a brief description of how we initiated our system.

Chapter 3 is **METHODOLOGY,** where a description of our dataset can be found. Moreover, this chapter also contains “Implementation” where ten algorithms of Machine Learning are described.

Chapter 4 is **RESULT ANALYSIS,** where we have shown how different algorithms performed with respect to our dataset.

Chapter 5 is **CONCLUSION & FUTURE WORK** consisting “Conclusion” and “Future Work".

CHAPTER II

LITERATURE REVIEW

This chapter contains the background research/ related existing work summaries.

2.1 Background

Biomechanics is the study of the movement of living beings using the science of mechanics. According to the science of mechanics, motion is created by force. Living beings create motion using force. Biomechanics gives mathematical and theoretical tools which are essential to understand how living things make moves [1]. The condition when the gel like material (Nucleus Pulposus) get squeezed out through fractures in outer wall of intervertebral disc is known as Lumbar Disk Herniation [2]. Spondylolisthesis is a medical condition in which one of the bones of a person's back (vertebrata) slides forward over the bone below it.

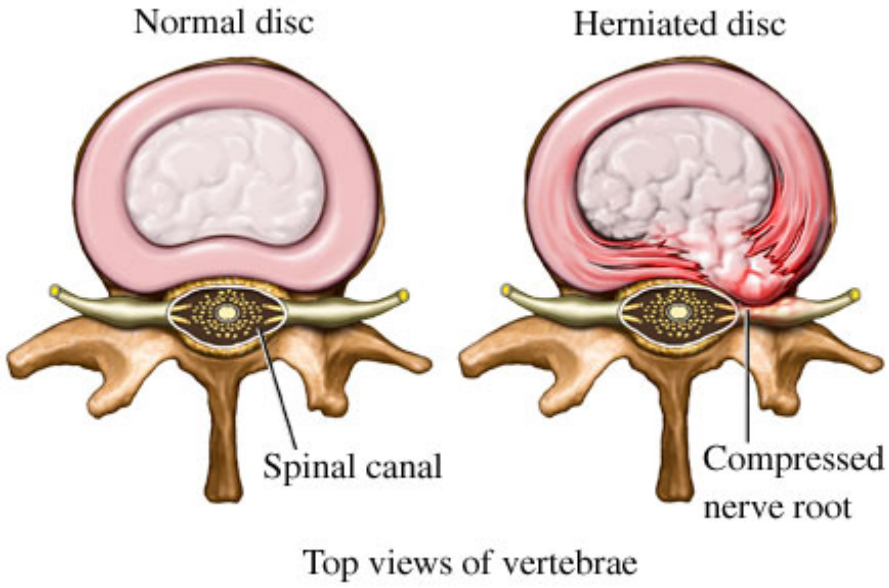


Figure 2.1.1. An Example of Disk Herniation.

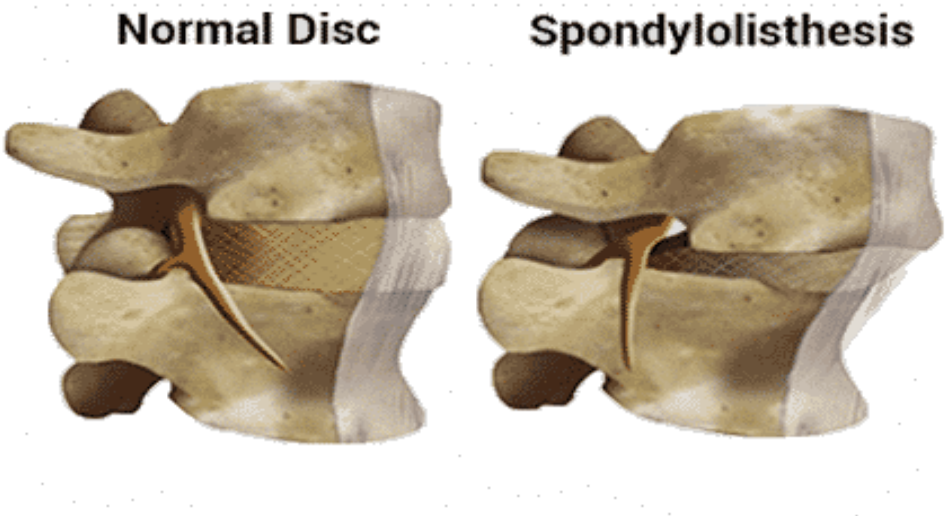


Figure 2.1.2. An Example of Spondylolisthesis.

Most of the time it occurs in lower spine or lumbosacral area [3]. Both conditions can squeeze the spinal cord or nerve roots and cause pain.

2.2 Previous Works

For the past few years, different algorithms of machine learning has been applied to predict and classify different sorts of diseases. Algorithms like K-Nearest Neighbor (k-NN), Logistic Regression (LR), Random Forest (RF) and Support Vector Machine (SVM) were used to classify cardiac arrhythmia where SVM out performed others [4]. Again, Random Forest was used to classify Pulmonary Tuberculosis and Sarcoidosis where it provided a very good accuracy [5]. SVM has been used to classify the types of Leukemia [6]. Machine learning techniques are being used extensively to detect diseases in easier way. Algorithms like Artificial Neural Network (ANN) and Probabilistic Neural Network (PNN) have been compare to predict osteoporosis where ANN provided better results than PNN [7]. Application of ANN can be found in many medical fields including radiology [8], oncology [9], urology [10] - [13], cardiology

[14]. ANN has been proven to be very useful to develop existing medical technique. [15] - [18].

2.3 Supervised Learning

Maximum practical machine learning uses supervised learning. In supervised learning we have input variable(X), output variable (Y) and then we use an algorithm to learn the mapping function from input and output.

$$Y = f(X) \quad (1)$$

The goal is to approximate the mapping function so well that when there is new input data (x), the algorithm can predict the output variables (Y) for that data. It is called supervised learning because the learning process of an algorithm from the training dataset can be thought of as a teacher supervising the learning process, where the teacher knows the correct answers; the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance [19]

CHAPTER 3

METHODOLOGY

3.1 Dataset

The dataset we used in this study contains 620 instances, each containing six features named pelvic incidence, pelvic tilt numeric, lumbar lordosis angle, sacral slope, pelvic radius and degree spondylolisthesis. Table I. shows a sample of the dataset used in the system.

Table 3.1. Sample of Dataset

pelvic incidence	pelvic tilt numeric	lumber lordosis angle	sacral slope	pelvic radius	degree spondylo listhesis	class
63.02782	10.06099	39.60912	40.47523	98.67292	98.67292	Abnormal
39.05695	22.21848	25.01538	28.99596	114.4054	114.4054	Abnormal
38.50527	18.96843	35.11281	21.54098	127.6329	127.6329	Normal
44.36249	9.652075	46.9021	35.41706	129.2207	129.2207	Normal
49.71286	13.92191	28.31741	40.06078	108.1687	108.1687	Hernia
40.2502	15.86434	25.12495	26.32829	130.3279	130.3279	Hernia
53.43293	32.70443	37.16593	37.56859	120.5675	120.5675	Hernia
89.68057	9.433234	83.13073	56.97613	129.9555	129.9555	Spondyloli sthesis
44.52905	9.433234	52.28371	35.09582	134.7118	134.7118	Spondyloli sthesis
77.69058	21.38064	64.42944	56.30993	114.8188	114.8188	Spondyloli sthesis

The class contains four values, "Normal", “Abnormal", "Hernia" and "Spondylolisthesis". Each patient is represented in this dataset by these six biomechanical features and their conditions are represented as the class. Pelvic incidence is a morphological parameter for pelvic. It is a positional parameter to represent three dimensional orientation of pelvis. The pelvic incidence, with sacral slope and pelvic tilt, determines the conditions of the principle of biomechanical economy. The measure of pelvic incidence is a complementary approach of the standing posture, as in scoliosis, low back pain, spondylolisthesis, spine and hip surgery, obesity, and postural impairments. Pelvic tilt is the orientation of the pelvis in respect to the thighbones and the rest of the body. The pelvis can tilt towards the front, back, or either side of the body. Pelvic tilt numeric is the parameter that shows the orientation in numeric value. Again, Lumbar lordosis angle is used to measure lumbar and spinal curvatures. Sacral slope is the angle between a sacral plate and horizontal line. Moreover, pelvic incidence is the sum of pelvic tilt and sacral slope. Pelvic radius is another parameter which measures angle between a line drawn between hip axis and the posterior corner of pelvis. Lastly, degree

spondylolisthesis is a parameter of measuring how much of a vertebral body has slipped forward over the body beneath it. Each parameter has been used as the column of the dataset and the dataset has been converted into a CSV file. This file has been used as the input of the system for classification.

3.2 Implementation

3.2.1 Adaptive Boosting

Adaptive boosting classifier combines weak classifier algorithm in order to form strong classifier. It is an ensemble classifier. The equation for choosing the final classifier,

$$h_s(x) = sign(\sum_{t=1}^{T} w_t h_t(x)) \quad (2)$$

Here, h_t=the output of the weak classifier‘t’, α_t=weight applied to classifier ‘t’.

Linear combination of all the weak classifiers is the final output. The final decision is made according to the ‘sign’ of the sum. The

classifiers are trained one at a time. After that output weight is computed using the equation given below:

$$w_t = \frac{1}{2}\log(\frac{1}{\epsilon_t} - 1) \quad (3)$$

After weight computation the training example weights are updated using the formula given below:

$$D_i^{(t+1)} = \frac{D_i^{(t)}\exp(-w_t y_i h_t(x_i))}{\sum_{j=1}^{m} D_j^{(t)}\exp(-w_t y_j h_t(x_j))} \quad (4)$$

Pseudo code:

AdaBoost

input:

training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$

weak learner WL

number of rounds T

initialize $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$.

for $t = 1, \ldots, T$:

invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$

compute $\epsilon_t = \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$

let $w_t = \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$

update $D_i^{(t+1)} = \frac{D_i^{(t)}\exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^{m} D_j^{(t)}\exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$

output the hypothesis $h_s(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=1}^{T} w_t h_t(\mathbf{x})\right)$.

3.2.2 Decision Tree

Decision Tree is a widely used classifier. It partitions data into subsets. The partition continues until there is no partition possible. The partition is done with Binary Split. The main purpose of Decision Tree is to shrink the training dataset in the smallest tree [19]. Decision Tree uses the entropy function for characterizing impurity of a dataset.

$$entropy(dataset) = -(p_+ * \log_2(P-)) \quad (5)$$

The equation for information gain given below,

$$IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - remainder(A) \quad (6)$$

$$remainder(A) = \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} I\left(\frac{p_i}{p+n}, \frac{n_i}{p+n}\right) \quad (7)$$

Pseudo code:

```
GrowTree(D, A) // tree with attribute for the training dataset
Input: data D; set the attributes A
Output: attribute tree T associated with labeled leaves.
If Homogeneous(D)
return label(D);
S = BestSplit(D,F);
Split D into substes Di;
If Di is not empty
Ti = GrowTree(Di,F);
else
Ti is a leaf labeled with Label(D);
return tree; // root labeled with S and children are Ti
Bestsplit(D,F) // best split for the decision tree
Input: data D; //set of attributes
Output: attributes A to split
Imin = 1;
for each f ∈ F do
split D into subsets D1; // D1 based on the values of f
if impurity({D1,……,Dn}) <Imin then
Imin = Impurity({D1,……..,Dn});
fbest = f;
end
end
returnfbest;
```

3.2.3 Gaussian Process

Gaussian Process is a non-parametric classification method. GPs can be applied to integration, global optimization, and mixture of expert model, unsupervised learning models and more. Equation of posterior probability distribution given below:

$$P(y(x)|t_{n,}\ X_n) = \frac{P(t_n|y(x),(X_n)P(y(x))}{P(t_n,X_n)} \quad (8)$$

$P(t_n|y(x),(X_n)$ is the probability of the target values which is given in the function y(x). P(y(x)) this part is the prior distribution on function which is assumed by the model. For predicting the future values of 't' only assumed prior P(y(x)) is needed. The main idea of Gaussian Process modeling is to put P(y(x)) directly on the space of function without parameterizing y(x). The simplest type of prior over functions is called a Gaussian Process. Similar to Gaussian distribution which is specified by its mean and covariance matrix, Gaussian process is specified by a mean and a covariance function.

3.2.4 K-Nearest Neighbor

Most of the algorithms in machine learning are parametric but K-Nearest Neighbor is non-parametric. It classifies object based on majority of vote it gets from its closest neighbors. This algorithm predicts the class by using the Euclidean distance. Choosing ‘K’ is a critical problem. The value of ‘K’ influence result. For smaller value of k means noise will have higher influence on result. On the other hand large value of ‘K’ increase the computation cost. The equation for selecting ‘K’ is given below:

$$K = n^{\frac{1}{2}} \tag{9}$$

Pseudo code:

Let (X_i, C_i) where i = 1, 2……., n be data points. X_i denotes feature values & C_i denotes labels for X_i for each i; assuming the number of classes as 'c'. Ci ∈ {1, 2, 3, ….., c} for all

Let x be a point for which label is not known, and we would like to find the label class using k-nearest neighbor algorithms.

1. Calculate "d(x, x_i)" i =1, 2, …, **n**; where **d** denotes the Euclidean distance between the points.
2. Arrange the calculated **n** Euclidean distances in non-decreasing order.
3. Let **k** be a +(ve) integer, take the first **k** distances from this sorted list.
4. Find those **k**-points corresponding to these **k**-distances.
5. Let $\mathbf{k}_i$ denotes the number of points belonging to the i^{th} class among **k** points i.e. $k \geq 0$
6. If $k_i > k_j \; \forall \; i \neq j$ then put x in class i.

3.2.5 Logistic Regression

Logistic Regression searches the whole datasets to find the hyper plane which fits the most for identifying the classes. The core of logistic regression is “logistic function”. Logistic function is also called the sigmoid function. This Function was mainly developed for describing the properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It is a ‘S’ shaped curve which can take real-valued number and map it into a value between 0 and 1. The function given below:

$$1 / (1 + e\text{^} - \text{value}) \qquad (10)$$

Here ‘e’ is the base of natural logarithms and value represents numerical value which we want to transform. For example let us a range (numeric) from -5 to 5. The transformation plot into the range between 0 to 1 using the logistic function is given below,

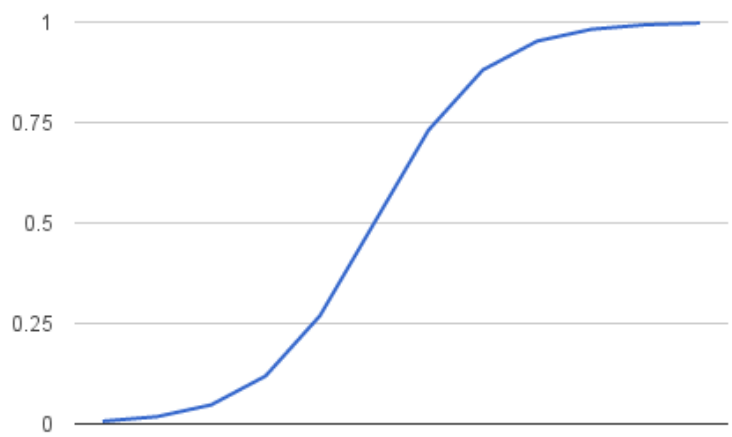


Figure 3.2.5.1. Logistic function curve.

Now the LR equation is –

$$y = e^{(b0 + b1 * x)} / (1 + e^{(b0 + b1 * x)}) \quad (11)$$

Here, y = predicted output, b0 = bias or intercept term, b1 = co-efficient for the single input value(x). Every column of the input data has an associated ‘b’ co-efficient which needs to be learned from training data.

3.2.6 Multi-Layer Perceptron

Multilayer perception is a supervised learning algorithm. MLP has multiple nodes arranged in interconnected layers named input, hidden and output layers. In MLP group of inputs are mapped into a set of desired outputs. The structure of MLP given below

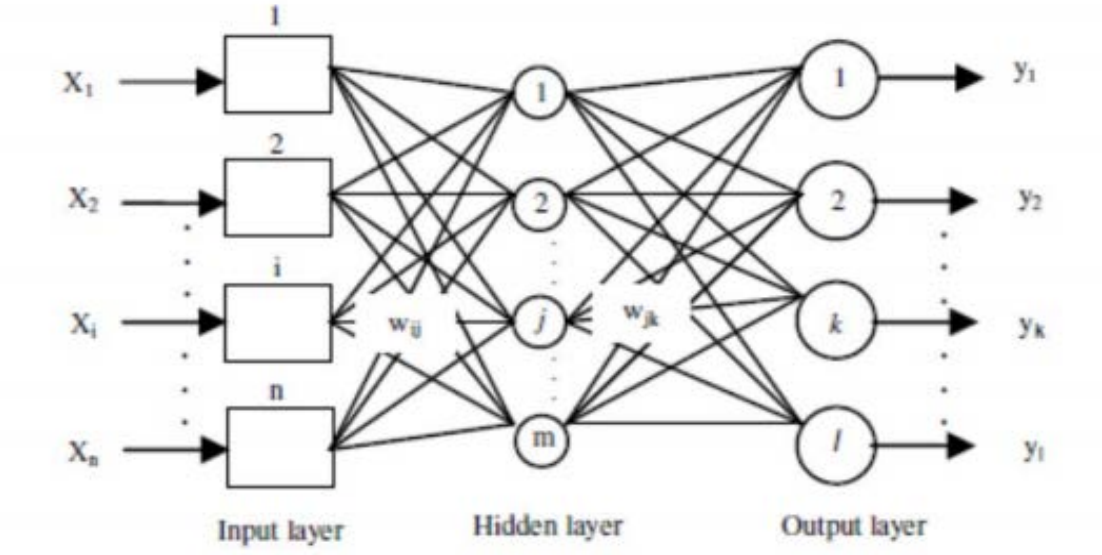


Figure 3.2.6.1. Multi-Layer Perceptron Structure.

The hidden layer cannot be directly accessed. Every layer is made of several neurons. Neurons in different layers are connected with the

use of weight and bias [flood prediction]. Equation for the output of neuron ‘j’ in hidden layer given below,

$$H_j = f(\sum_{i=1}^{n}(w_{ji}x_i + b_i) \quad (12)$$

Here, w_{ji} = weights, b_i = biases, $f()$ = nonlinear activation function

The equation for the network output given below:

$$y = f(\sum_{j=1}^{m} w_{kj}H_i + b_o) \quad (13)$$

Here, f = output layer neuron activation function, w_{kj}= weight, b_o=bias.

Pseudo code:

1: Set $Error_{Max}$,$Iteration_{max}$,$Rate_{learning}$
2: Set $NNLayer_{input}$, $NNLayer_{hidden}$, and $NNLayer_{output}$
3: Read $Data_{Training}$, apply input to the ML-NN network.
// ML-NN training phase
4: For every input,
5: Compute the output
6: Compute error by comparison of the acquired output value with the expected output for the given input
7: Adjust the *weights* for all neurons using the obtained *error*
8: Repeat operation until acquired error reached acceptable value, $Error_{Max}$
9: End For loop
10: End

3.2.7 Naïve Bayes

Naïve Bayes is a probabilistic classifier based on Bayes' Theorem. It utilizes the independent attributes of the data set to make main assumption. Independent attributes is also the main point to make predictions. Bayes rule is the combination of conditional probability with product and sum rule. The equation is given below,

$$p(X = x|Y = y) = \frac{p(X=x,Y=y)}{p(Y=y)} = \frac{p(X=x)p(Y=y|X=x)}{\sum_{x\prime} p(X=x')p(Y=y|X=x\prime)} \qquad (14)$$

A flowchart of Naïve Bayes' classifier is given below:

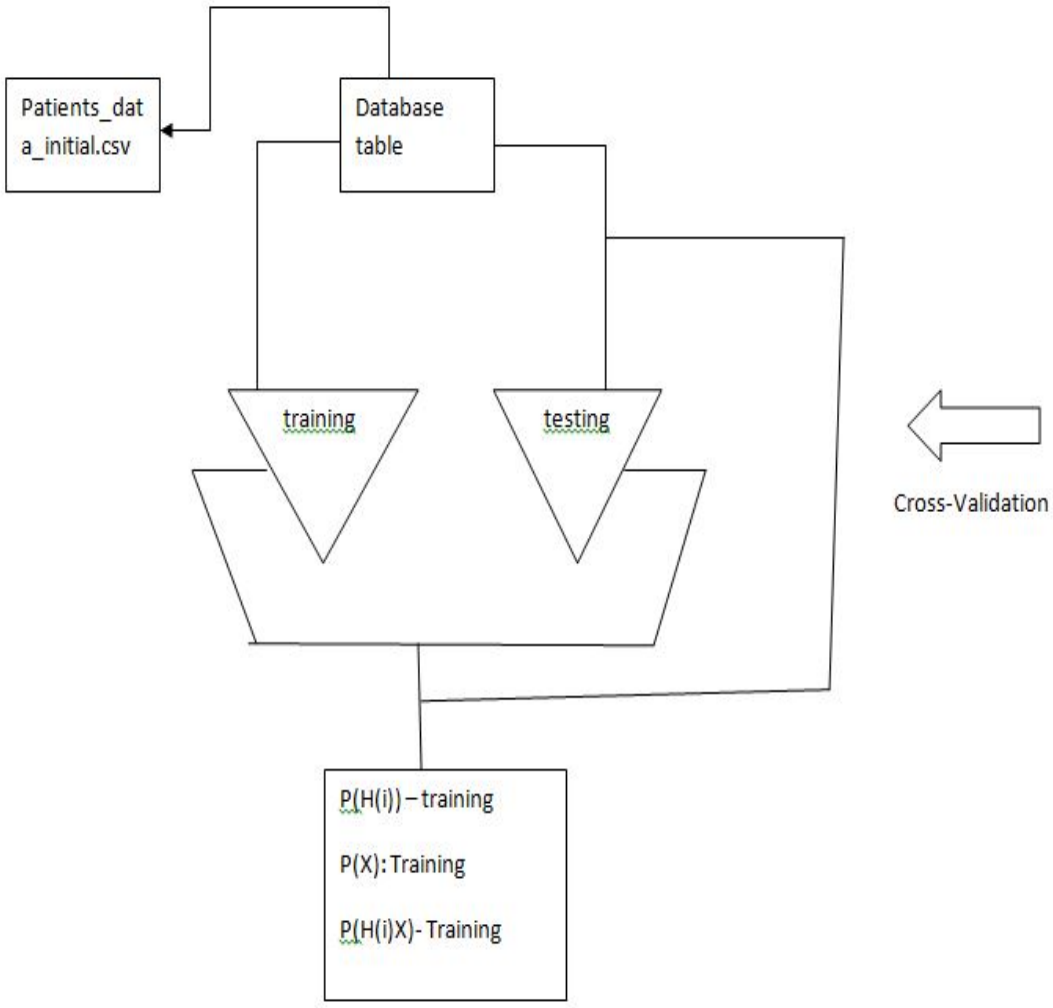


Figure 3.2.7.1. Naïve Bayes Flow chart

3.2.8 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis is closely related to Linear Discriminant Analysis. For the estimation of parameters needed in quadratic discriminant analysis, more computation and data is required. Quadratic discriminant function given below:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k \tag{15}$$

Classification rule:

$$\hat{G}(x) = \arg\max_k \delta_k(x) \tag{16}$$

QDA has separate covariance matrix for every class.

3.2.9 Random Forest

Random forest algorithm is a supervised classification algorithm. Random Forests minimizes the variance which might cause error in decision tree. A set of decision trees are created from randomly selected subset of training set. After that it aggregates the votes from different decision trees to choose the final class of the test object. The more number of trees the higher chance to get accurate results. RF gives high accurate results and it learns very fast. It is very efficient on large data sets. Random forest uses gini index for deciding the final class of each tree. If data set T contains examples from n classes gini index, Gini (T) is defined as

$$Gini(T) = 1 - \sum_{j=1}^{n}(Pj)^2 \quad (17)$$

Flow chart of Random Forest given below:

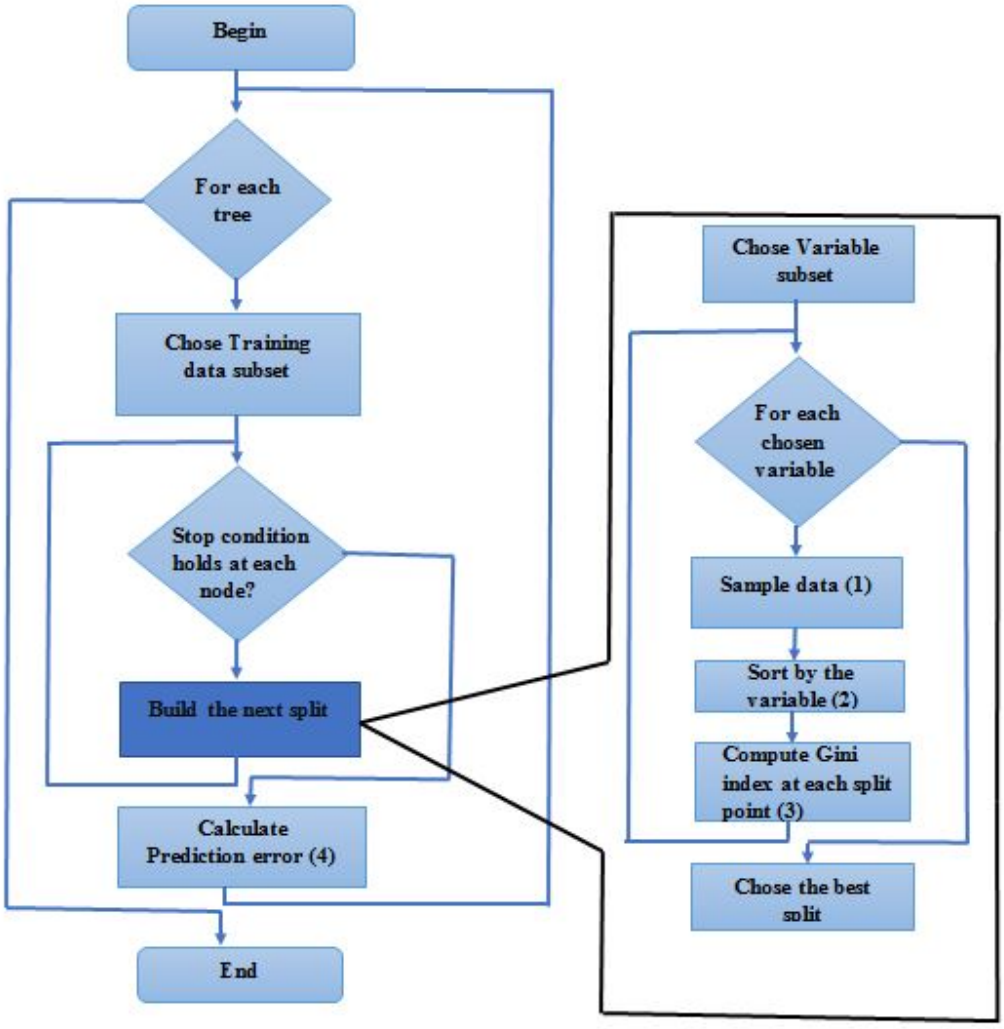


Figure 3.2.9.1. Random Forest Flow Chart.

3.2.10 Support Vector Machine

Support Vector Machine is a supervised Machine Learning algorithm. Basic concept of this algorithm is finding a hyper plane in order to classify the datasets. There are two kinds of SVM classifier –

a) SVM Linear Classifier

b) SVM non-linear Classifier.

The flowchart of SVM classifier is given below:

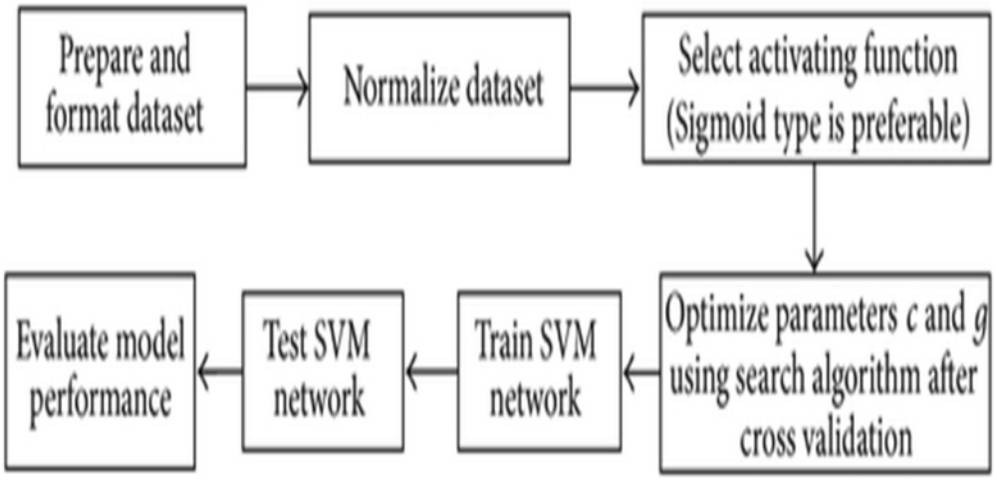


Figure 3.2.10.1. SVM Workflow.

CHAPTER 4

RESULT ANALYSIS

In our study, we have used ten classifiers named Decision Tree (DT), K-Nearest Neighbor (k-NN), Naive Bayes (NV), Adaptive Boosting (ADB), Support Vector Machine (SVM), Random Forest (RF), Quadratic Discriminant Analysis (QDA), Multi-Layer Perception (MLP), Logistic Regression (LR) and Gaussian Process (GP). We have used python and its related packages for classification. The whole dataset was divided into two parts. For the first analysis, we used a dataset with 434 instances and for our second analysis, we used a dataset with 186 instances. In our supervised learning system, we have used about 30% of the data to train the system and the rest for testing. The six features from the dataset have been used as the input of the system for classification. After training the system, firstly we tested the system to check if the patient is in normal state or not. If the patient is not in normal condition, then we tested whether the disease Hernia or Spondylolisthesis. Accuracy score and Confusion

matrix were used as metrics to calculate the performance of the algorithms.

Table 4.1. Confusion Matrix

N	No (Prediction)	Yes (Prediction)
No (Actual)	True Negatives (TN)	False Positives (FP)
Yes (Actual)	False Negatives (FN)	True Positives (TP)

Where, True Positives (TP) = these are cases in which algorithms predicted yes (they have the disease), and they do have the disease. True Negatives (TN) = these are cases in which algorithms predicted no, and they don't have the disease. False positives (FP) = these are the cases in which algorithms predicted yes, but they don't actually have the disease. False Negatives (FN) = these are the cases in which algorithms predicted no, but they actually do have the disease. n = total number of instances. From each algorithms confusion matrix, we used the following equation to derive the accuracy of each algorithm,

$$\text{Accuracy} = (TP + TN)/n \quad (18)$$

We have performed our both analysis for ten machine learning techniques mentioned before. For both of our analysis, we compared the predictions of each algorithm with the actual results from the test set. In this part, we reveal the output of each technique. Firstly we are going to show the comparison between the actual result and the predictions of each algorithm separately for both of our analysis. We have used Matplotlib library to visually represent the results of each algorithm. Fig 4.1.1-4.10.2 shows the comparison between actual result and the predictions of each algorithm for both analysis in the following order, DT, KNN, RF, SVM, ADB, MLP, NB, QDA, LR and GP.

In the graphs, on Y-axis, 0 represents “Abnormal” and 1 represents “Normal” for first analysis. For the second analysis 0 represents “Hernia” and 1 represents “Spondylolisthesis”. The black line represents the actual results and the red line represents the prediction of the each algorithm. Hereby it can be said that any red line away from the black line is considered as wrong predictions. So, the algorithm with the less number of red lines is considered most accurate.

4.1 Decision Tree:

The comparison between actual result and the prediction of DT is as follows,

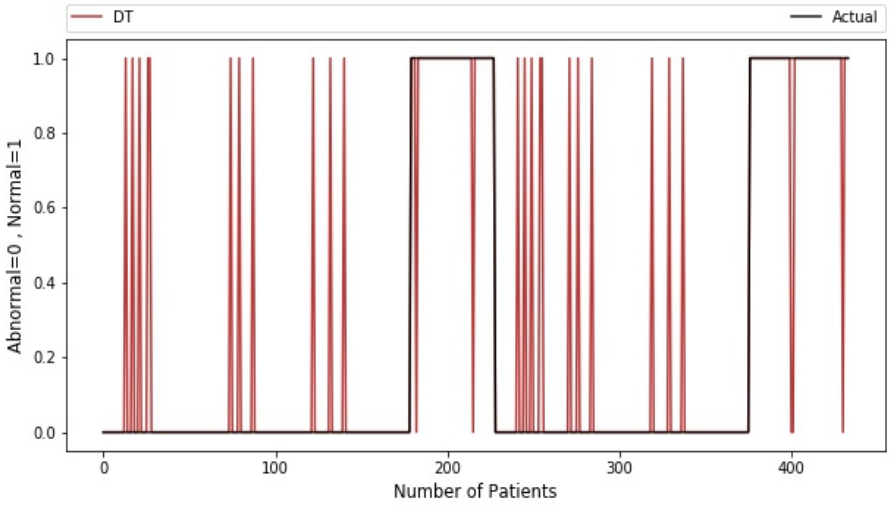


Figure 4.1.1. DT Comparison for first analysis

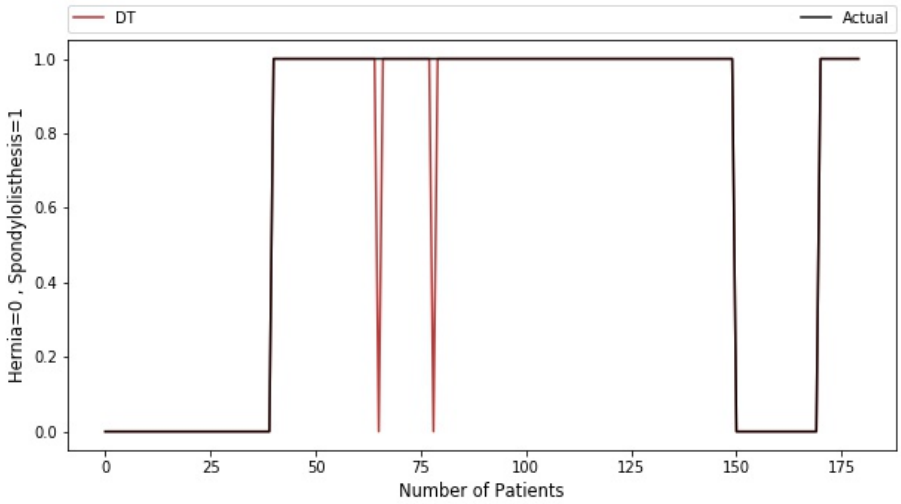


Figure 4.1.2. DT Comparison for second analysis

4.2 K-Nearest Neighbor:

The comparison between actual result and the prediction of KNN is as follows,

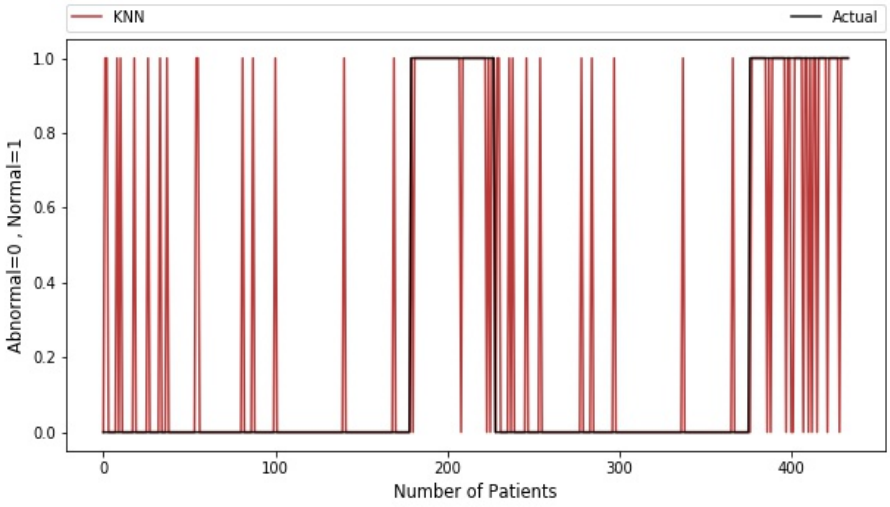


Figure 4.2.1. K-NN Comparison for first analysis

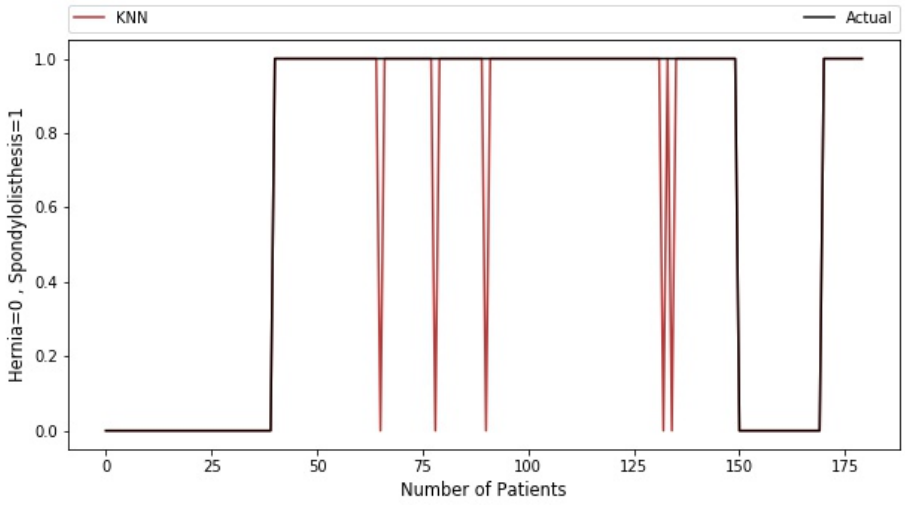


Figure 4.2.2. K-NN Comparison for second analysis

4.3 Random Forest:

The comparison between actual result and the prediction of RF is as follows,

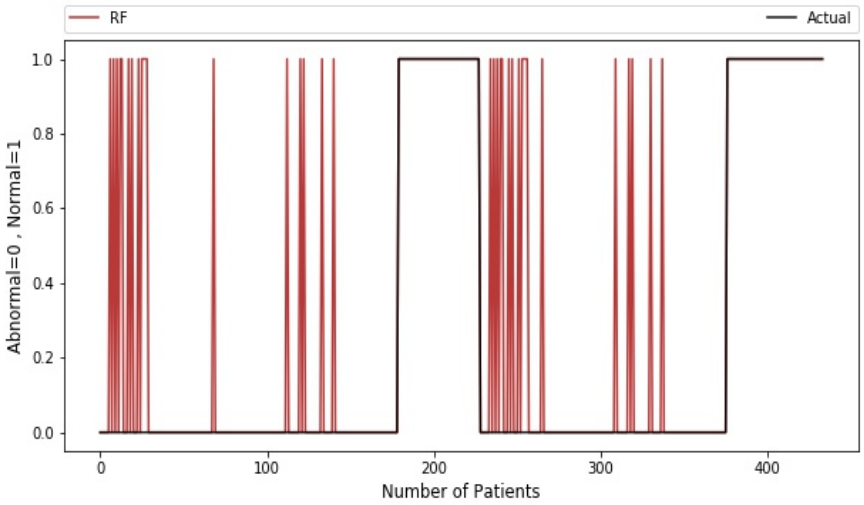


Figure 4.3.1. RF Comparison for first analysis

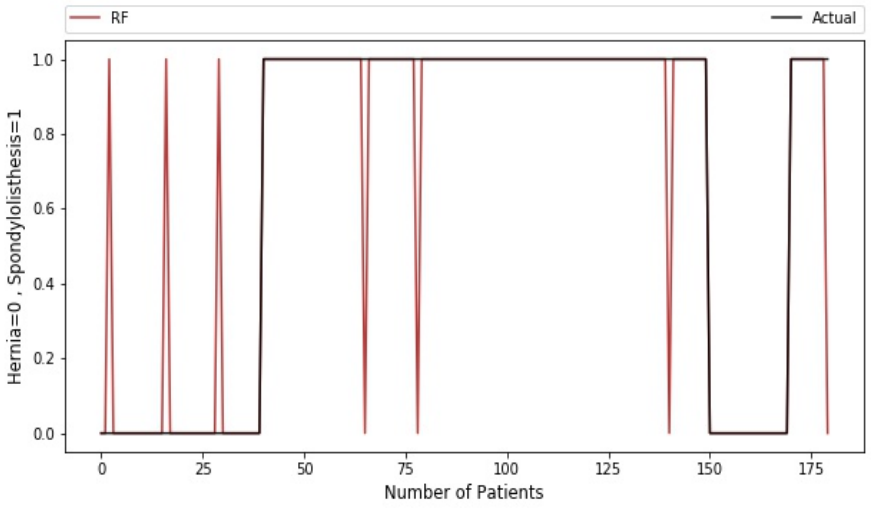


Figure 4.3.2. RF Comparison for second analysis

4.4 Support Vector Machine:

The comparison between actual result and the prediction of SVM is as follows,

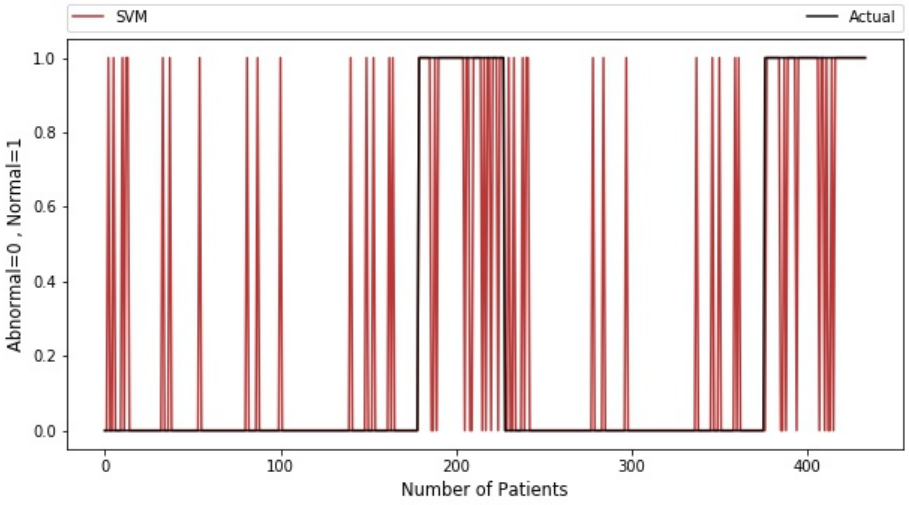


Figure 4.4.1. SVM Comparison for first analysis

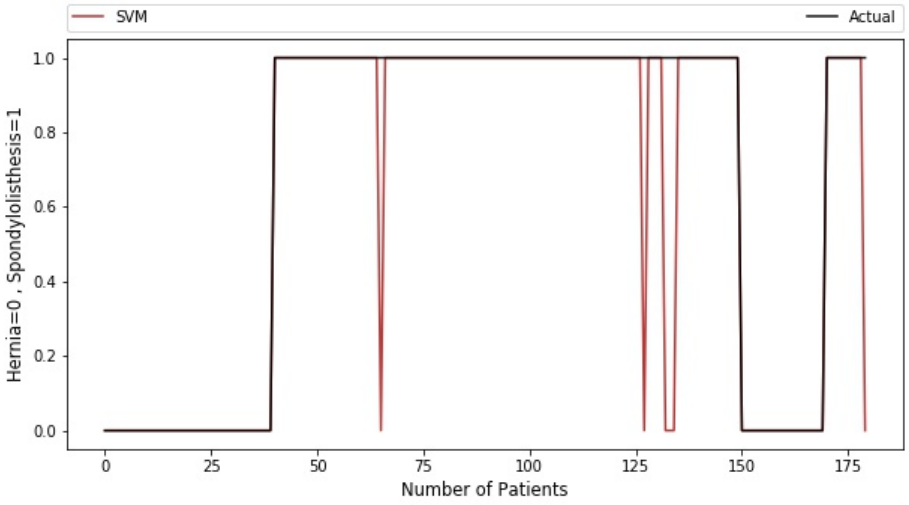


Figure 4.4.2. SVM Comparison for second analysis

4.5 Adaptive Boosting:

The comparison between actual result and the prediction of ADB is as follows,

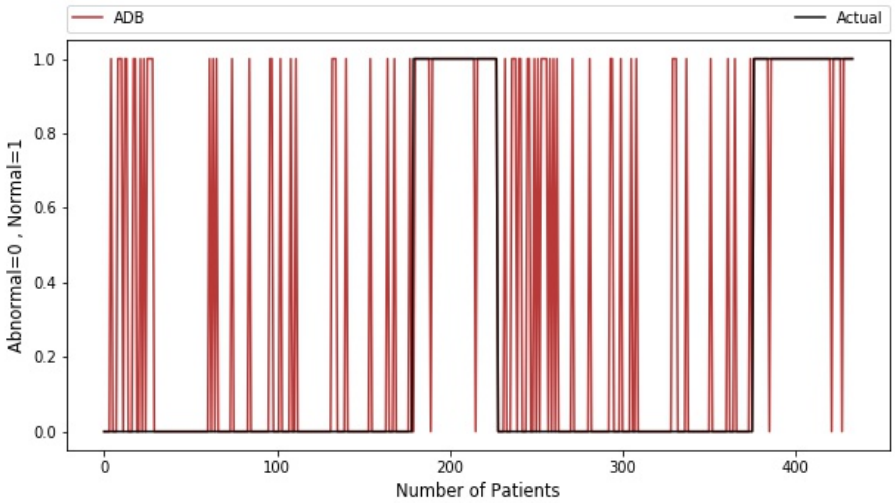


Figure 4.5.1. ADB Comparison for first analysis

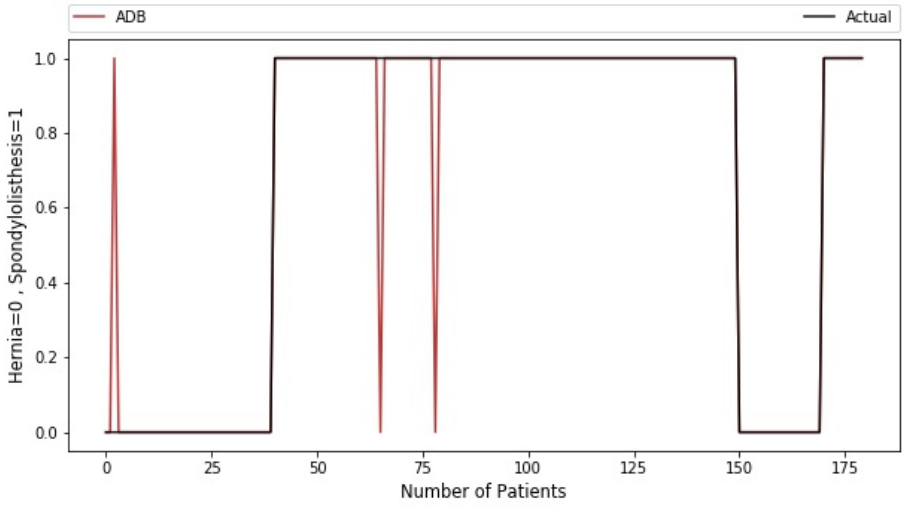


Figure 4.5.2. ADB Comparison for second analysis

4.6 Multilayer Perceptron:

The comparison between actual result and the prediction of MLP is as follows,

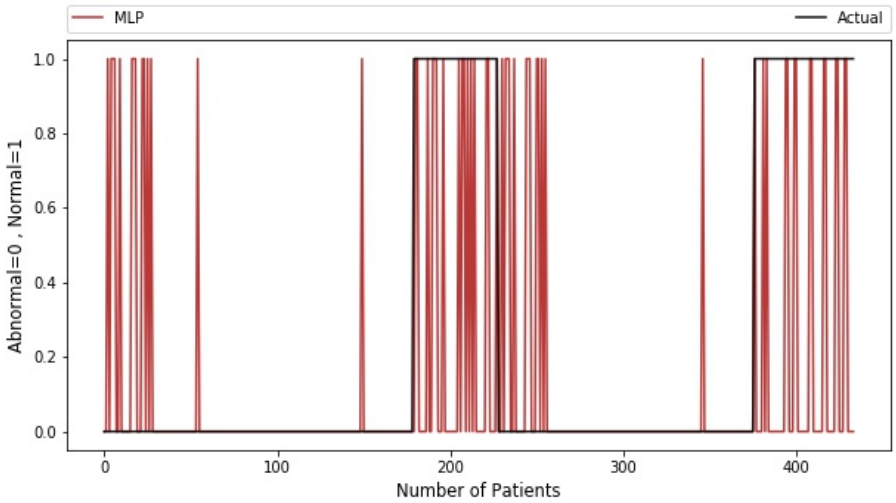


Figure 4.6.1. MLP Comparison for first analysis

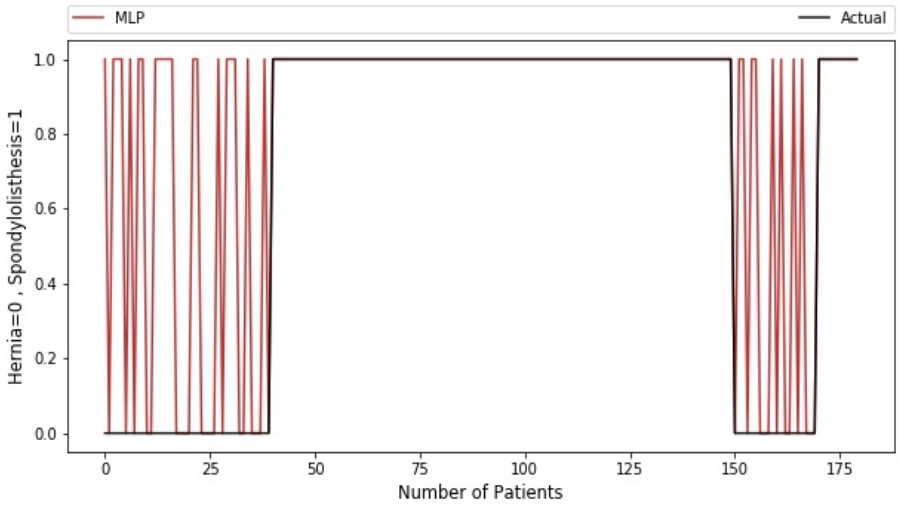


Figure 4.6.2. MLP Comparison for second analysis

4.7 Naïve Bayes:

The comparison between actual result and the prediction of NB is as follows,

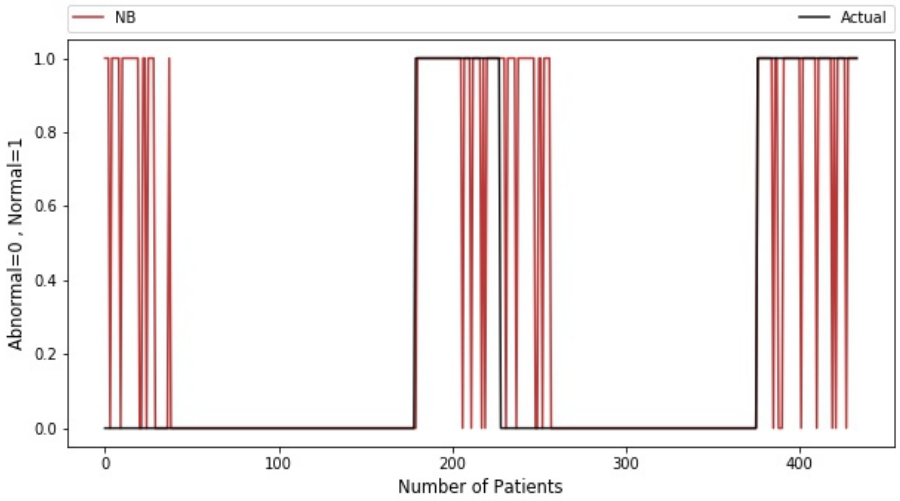


Figure 4.7.1. NB Comparison for first analysis

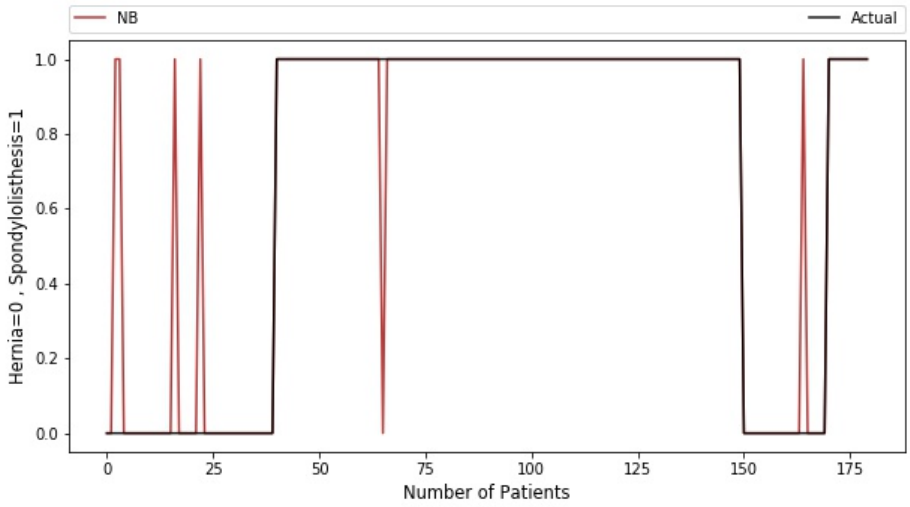


Figure 4.7.2. NB Comparison for second analysis

4.8 Quadratic Discriminant Analysis:

The comparison between actual result and the prediction of QDA is as follows,

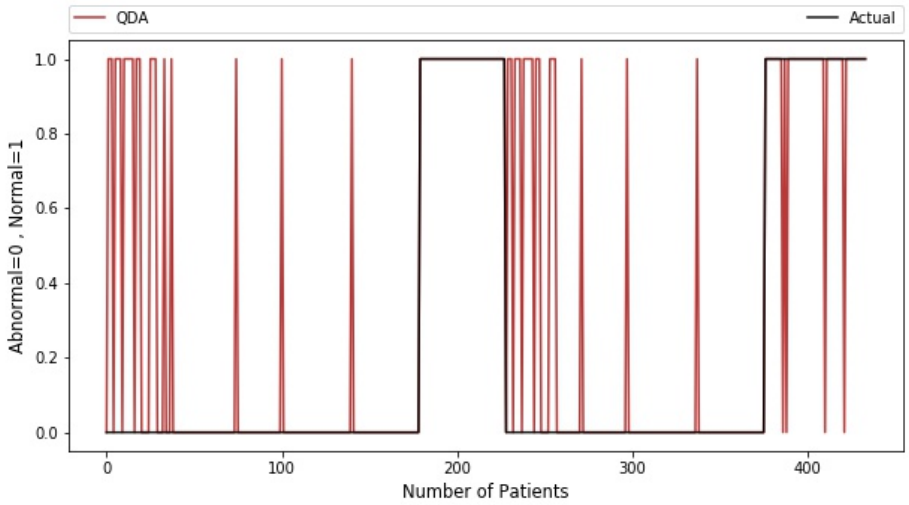


Figure 4.8.1. QDA Comparison for first analysis

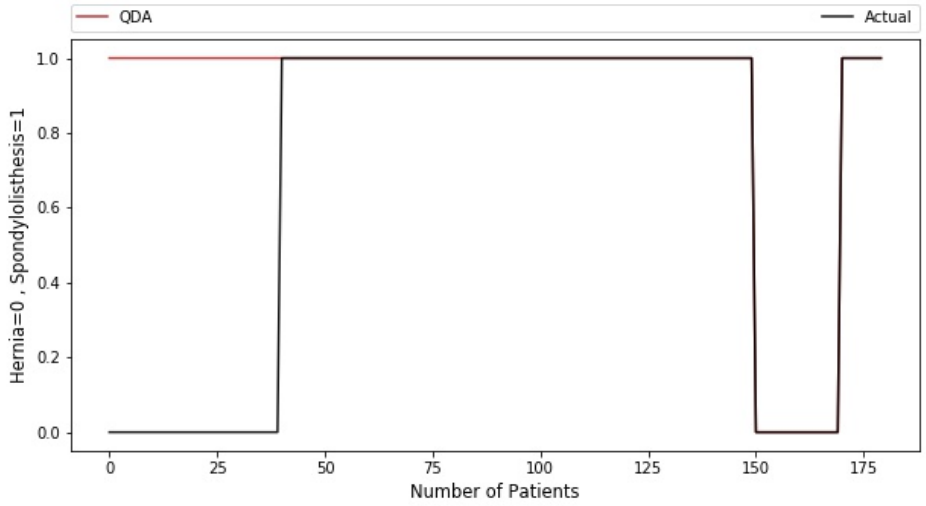


Figure 4.8.2. QDA Comparison for second analysis

4.9 Logistic Regression:

The comparison between actual result and the prediction of LR is as follows,

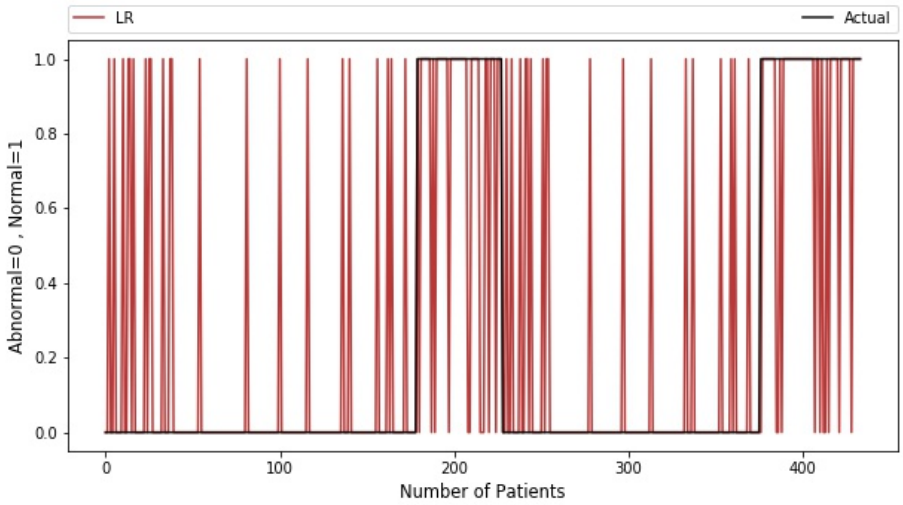


Figure 4.9.1. LR Comparison for first analysis

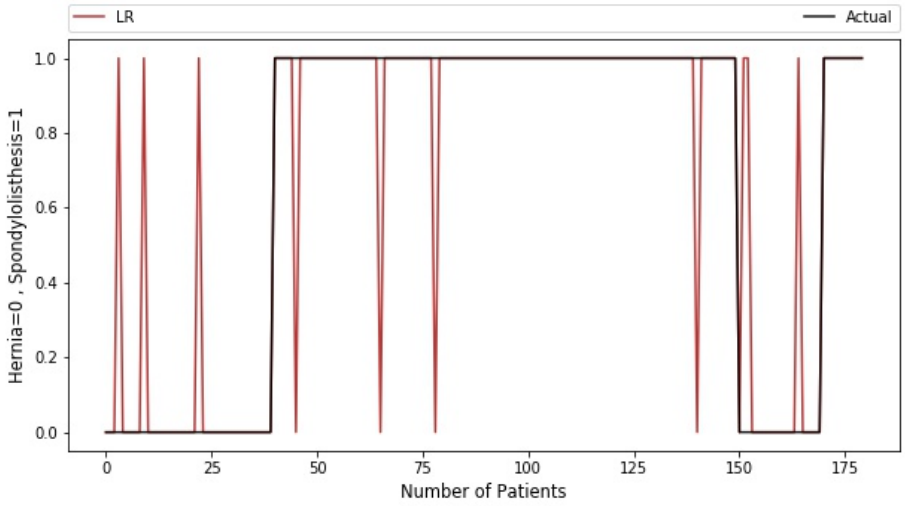


Figure 4.9.2. LR Comparison for second analysis

4.10 Gaussian Process:

The comparison between actual result and the prediction of GP is as follows,

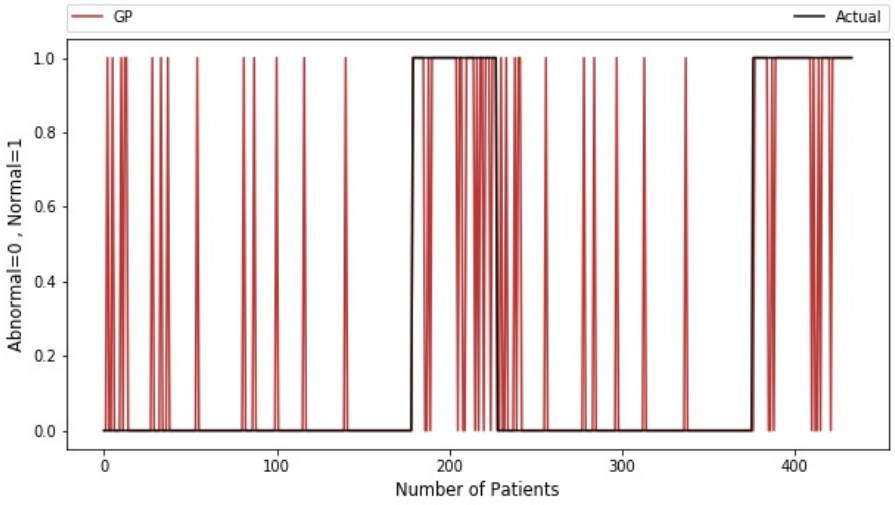
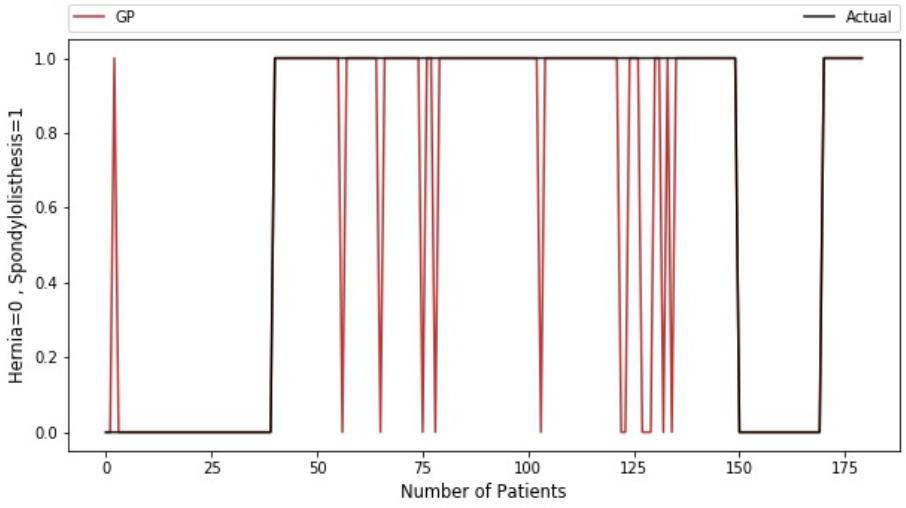


Figure 4.10.1. GP Comparison for first analysis



`

Figure 4.10.2. GP Comparison for second analysis

In this part, we are going to show the comparison between all the algorithms for both of our analysis. Table III shows the accuracy in percentage for each algorithm for both of our analysis. Fig. 4.1 & 4.2 shows the predictions of each algorithm for both of our analysis respectively. For fig. 4.1, blue and red block represents the number of patients predicted to be in normal and abnormal state respectively. In fig. 4.2, green and red block represents the number of patients predicted to be diagnosed with Hernia and Spondylolisthesis respectively. Actual represents the actual result.

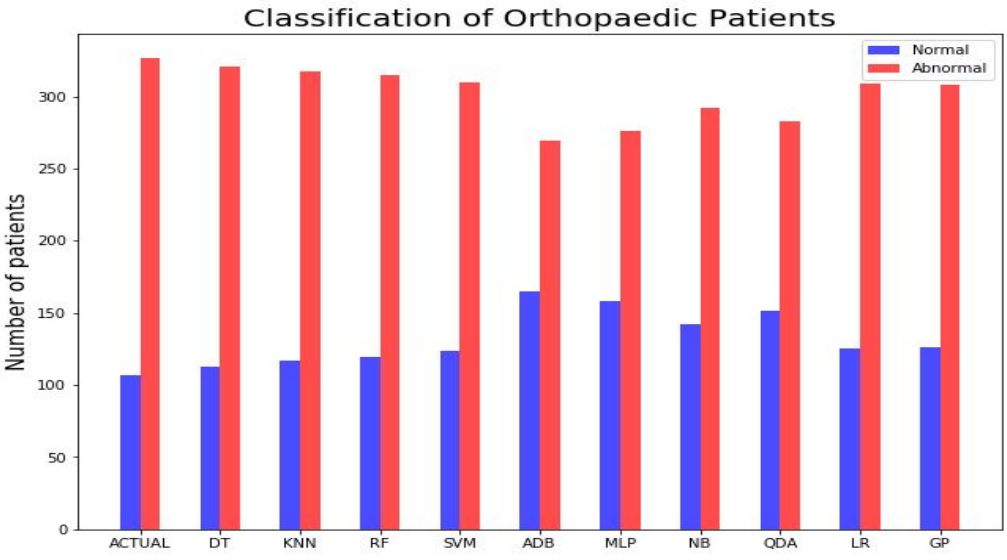


Figure 4.1. Comparative Output histogram for first analysis

From fig. 4.1, we can see that some of the algorithms prediction was close to the actual result, whereas decision tree was only few units away from the actual result, providing the best result among all the algorithms for our first analysis.

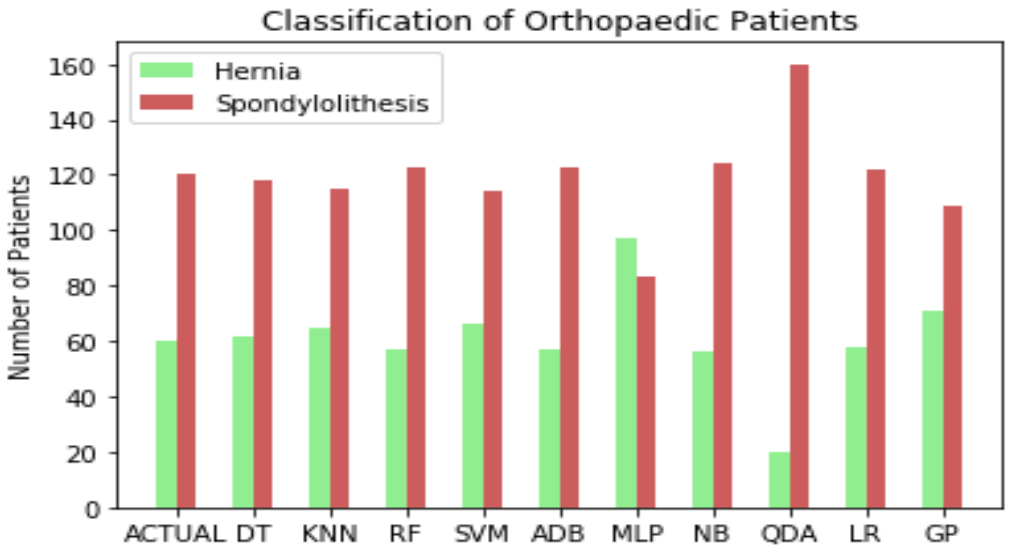


Figure 4.2. Comparative Output histogram for second analysis

From fig. 4.2, we can see that, unlike the result of first analysis, decision tree was only 2 units away from the actual result, providing the best result among all the algorithms for our second analysis as well. From the result of both of our analysis, we can state that Decision Tree algorithm provides the best result for this kind of analysis given the similar type of dataset.

Table 4.2. Accuracy Distribution

Algorithm	Normal/ Abnormal (%)	Hernia/ Spondylolisthesis (%)
Decision Tree	92	99
K-Nearest Neighbor	90	97
Random Forest	90	97
Support Vector Machine	89	96
Adaptive Boosting	84	98
Multi-Layer Perceptron	75	85
Naïve Bayes	85	97
Quadratic Discriminant Analysis	88	77
Logistic Regression	86	94
Gaussian Process	89	93

CHAPTER 5

CONCLUSION & FUTURE WORK

5.1 Conclusion

A person can be categorized based on his orthopedic condition and to do that we have used a number of machine learning techniques. Since there has not been enough application of machine learning in orthopedic field, we tried to implement a number of algorithms to analyze the comparative performance. Six biomechanical features have been used as parameters for the algorithms. Among all the ten algorithms, Decision Tree has provided the most accuracy for our dataset. It has given 92% accuracy for the first analysis where it detected whether the person is normal or not. And for the second analysis it has provided 99% to detect if the person has Disk Herniation or Spondylolisthesis. This analysis can assist doctors in identifying diseases in a faster and easier way.

5.2 Future Work

In future our work can be extended with the help of deep learning and neural network. We wish to test our system in other fields of medical science. Algorithms that have been used can perform even better with expanding data. We wish to implement our system in all possible medical fields so that work load on doctors can be reduced. We also want to implement our system in large datasets and check how the algorithms perform. In case of poor performance we would like to enhance the performance by merging or any other way possible. We look forward to ease the process of disease prediction and make lives easier for both patients and doctors. In addition we would like to add diet and exercise recommendation according to the result of our analysis.

REFERENCES

[1] D. Knudson, Fundamentals of Biomechanics, 2nd ed, New York: Springer, 2007, pp.3.

[2] K. Alawneh, M. Al-dwiekat, M. Alsmirat and M. Al-ayyoub, "Computer-Aided Diagnosis of Lumbar Disc Herniation," in 6th International Conference on Information and Communication System (ICICS 2013), 2013.

[3] S. Liao et al., "Automatic Lumbar Spondylolisthesis Measurement in CT Images," IEEE Trans. Med. Imag., vol. 35, no. 7, pp.1658-1669, July 2016.

[4] P. Shimpi, S. Shah, M. Shroff and A.Godbole, "A Machine Learning Approach for the Classification of Cardiac Arrhythmia," in Proceeding of the IEEE International Conference on Computing and Communication (ICCMC), pp. 603-607, 2017.

[5] Y. Wu, H. Wang and F. Wu, "Automatic Classification of pulmonary Tuberculosis and Sarcoidosis based on Random Forest," in 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017.

[6] P. Jagadev and H.G. Virani, "Detection of Leukemia and its Types using Image Processing and Machine Learning," in International Conference on Trends in Electronics and Informatics (ICEI), IEEE, 2017.

[7] D. H. Mantzaris, G. C. Anastassopoulos and D. K. Lymberopoulos, "Medical Disease Prediction Using Artificial Neural Networks,"

[8] L. Lanzarini, M. V. Camacho, A. Badran, and I. D. G. Armando, "Images compression for medical diagnosis using neural

networks," *J. Computer Science and Technology*, Vol.2, No. 1, 1999, pp. 78-80.

[9] A. Taktak, A. Fisher, and B. Damato, "Modelling survival after treatment of intraocular melanoma using artificial neural networks and Bayes theorem," *Phys. Med. Biol.*, vol. 49, pp. 87-98, 2004.

[10] D. H. Mantzaris, G. C. Anastassopoulos, A. D. Tsalkidis, and A. V. Adamopoulos, "Intelligent prediction of vesicoureteral reflux disease," *WSEAS Trans. Systems*, Issue 9, vol. 4, pp. 1440-1449, 2005.

[11] D. Tasoulis, P. Spyridonos, N. Pavlidis, D. Cavouras, P. Ravazoula, G. Nikiforidis, et all, "Urinary bladder tumor grade diagnosis using on-line trained neural networks," in *Proc. Knowledge Based Intelligent Information Eng. Systems Conf.*, Heidelberg, 2003, pp. 199-206.

[12] M. Tanthanuch and S. Tanthanuch, "Prediction of upper urinary tract calculi using an artificial neural network," *J. Medical Association of Thailand*, vol. 87, No. 5, pp. 515-518, 2004.

[13] F. Dieterlea, S. Muller-Hagedorn, H. Liebich, and G. Gauglitza, "Urinary nucleosides as potential tumor markers evaluated by learning vector quantization," *Artificial Intelligence in Medicine*, vol. 28, issue 3, pp. 265–279, 2003.

[14] R. Silipo and C. Marchesi, "Artificial neural networks for automatic ECG analysis," *IEEE Trans. Signal Processing*, vol. 46, pp. 1417-1425, 1998.

[15] K. Kambouri, and S. Gardikis, "Selective clinical estimation of childhood abdominal pain based on pruned artificial neural networks," in *Proc. 3rd WSEAS Int. Conf. on Cellular and Molecular Biology, Biophysics and Bioengineering*, Athens, 2007, pp. 50-55.

[16] D. Mantzaris, G. Anastassopoulos, A. Adamopouos, I. Stephanakis, K. Kambouri, and S. Gardikis, "Abdominal pain

estimation in childhood based on artificial neural network classification," in *Proc. the 10th Int. Conf. on Engineering Applications of Neural Networks*, Thessaloniki, 2007, pp. 129-134.

[17] K. Papik, B. Molnar, R. Schaefer, Z. Dombovari, Z. Tulassay, and J. Feher, "Application of neural networks in medicine - a review," *Medical Scince Monitor*, vol. 4, no. 3, pp.538-546, 1998.

[18] M. Lundina, J. Lundina, H. Burked, S. Toikkanenb, L. Pylkkanenc, and H. Joensuua, "Artificial neural networks applied to survival prediction in breast cancer," *Oncology*, vol.57 no.4 pp.281-286, 1999.

[19] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE:synthetic minority over-sampling technique", *Journal of artificial intelligence Research,* 2002, pp. 321-357.

[20] H. M. M. G. T. Herath, J. R. S. S. Kumara, M. A. R. M. Fernando, K. M. K. S. Bandara and I. Serina, "Comparison of supervised machine learning techniques for PD classification in generator insulation," *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, 2017, pp. 1-6.

[21] S. Pouriyeh, S. Vahid, G. Sannino, G. D. Pietro, H. Arabnia, J.Gutierrez, "A Comprehensive Investigation and Comparison of Machine Learning Techniques in the Domain of Heart Disease", *IEEE Symposium on Computers and Communications (ISCC)*, Heraklion, Greece, 2017.

[22] C. McCormic (2013,Dec.), "AdaBoost Tutorial"[online], http://mccormickml.com/2013/12/13/adaboost-tutorial