

Symptom Analysis of Parkinson Disease using SVM-SMO and Ada-Boost Classifiers

Muhtasim Billah

Student ID- 10101002

Supervisor

Rubel Biswas

Co-Supervisor

Md. Zahangir Alam

Department Of Computer Science & Engineering

January 2014



BRAC University, Dhaka, Bangladesh

Abstract

Parkinson Disease (PD) occurs due to the loss of dopamine in the brains thalamic region that results in involuntary or oscillatory movement in the body. Normally Doctors diagnosis the PD disease clinically with their expertise and experience. But most of the time wrong diagnosis and treatment are reported. For this, patients need to take number of tests for diagnosis, but most of the time, these all tests still not sufficient to diagnosis Parkinson Disease effectively. Firstly, this paper is proposed to apply some data mining technique to select the best attributes to increase the classification performance (according to the voice measurement datasets). In second step, Ada-Boost algorithm is applied to classify the Parkinson disease on the basis of Voice measurements data of PD patients. Then, Support vector Machine with Sequential Minimal Optimization classifier, is used to make the comparison with the result of Ada-Boost classifier to find out the best classifier. In addition, six other best classifiers ex: Naïve bayes, J48 Tree, LogitBoost, ADTree, BFTree, and Decision Stump Tree are used to make comparison with Parkinson dataset and to select the best classifier.

Keywords- Data Mining, Parkinson data, Classification algorithm, Association algorithm Weka tool

————— □ —————

ACKNOWLEDGEMENT

I want to express my gratitude to all the people who have given their heart whelming full support in making this compilation a magnificent experience.

First and foremost I offer my sincerest gratitude to my Supervisor, Mr. Rubel Biswas, who has supported me throughout my thesis with his patience and knowledge.

I also thank my Co-supervisor Mr. Zahangir Alam for his support and initiative to induce knowledge.

I can't forget to thank Almighty Allah and my family. Both of my supervisor and co-supervisor have inspired, encouraged and fully supported me for every endeavor of mine and backed me to surpass the hurdles that come my way.

Supervised By

.....

Submitted By

.....

Co-Supervised By

.....

TABLE OF CONTENTS

	Page
TITLE	I
ABSTRACT	II
ACKNOWLEDGEMENT	III
TABLE OF CONTENTS	IV
LIST OF DIAGRAM	VI
LIST OF TABLE	VI
1. INTRODUCTION	VII
1.1 Problem Description	VIII
1.2 Objective	IX
1.3 Thesis Outline.....	IX
2. BACKGROUND ANALYSIS	
2.1 Parkinson Symptoms	X
2.2 Data Mining.....	XII
2.3 Previous Work	XIII
3. SYSTEM DESIGN	
3.1 Dataset Information.....	XIV
3.2 Preprocessing the data.....	XVI
3.3 Algorithm Explanation	
3.3.1 Ada-Boost Classifier.....	XVII
3.3.1.1 Ada-Boost.M1.....	XVIII
3.3.1.2 Ada-Boost.M2.....	XIX

3.3.2 Support vector Machine with Sequential	XXI
Minimal Optimization classifier	
3.3.3 WEKA Set Up.....	XXIII
4. EXPERIMENT & ANALYSIS	
4.1 ARFF File Formation.....	XXV
4.2 Visualization of the attributes.....	XXVII
4.3 AdaBoost WEKA classification	XXVIII
4.4 SVM-SMO WEKA classification	XXXIII
4.5 Naïve Bayes, J48, LogitBoost, ADTree, BFTree and.....	XXXVI
Decision Stump Tree Classification in WEKA Tool	
4.6 Comparison Table of AdaBoostM1, SVM-SMO, Naïve Bayes	XLII
J48, LogitBoost, ADTree, BFTree and Decision Stump Tree	
4.7 Result Analysis	XLIII
5. CONCLUSION AND FUTURE WORK.....	XLV
REFERENCES.....	XLVI

LIST OF DIAGRAM

1. Figure 1 WEKA startup screen.....	XXIV
2. Figure 2 WEKA Explorer	XXIV
3. Figure 3 ARFF file format	XXV
4. Figure 4 ARFF file format	XXVI
5. Figure 5 Visualizing all the Attributes	XVII
6. Figure 6 parkinsons.arff file opening	XVIII
7. Figure 7 After opening arff file in WEKA tool	XXIX
8. Figure 8 Selecting AdaBoostM1 from meta classifiers	XXX
9. Figure 9 Setting cross validation and starting	XXXI
10. Figure 10 AdaBoostM1 Weka Classification result	XXXII
11. Figure 11 Selecting SMO classifiers from WEKA tool	XXXIII
12. Figure 12 Setting cross validation and starting	XXXIV
13. Figure 13 SVM-SMO classifier result	XXXV
14. Figure 14 Naïve Bayes Classification	XXXVI
15. Figure 15 J48 Tree Classification	XXXVII
16. Figure 16 LogitBoost Classification	XXXVIII
17. Figure 17 ADTree Classification	XXXIX
18. Figure 18 BFTree Classification	XL
19. Figure 19 Decision Stump Classification	XLI

LIST OF TABLE

1. Table 1 : Parkinson Dataset Attribute Information	XIV
2. Table 2 : Comparison Table.....	XLII

1. Introduction

Parkinson's is a progressive neurological condition, [1] which is characterized by both motor (movement) and non-motor symptoms. The condition was first described by Dr James Parkinson in his Essay on the Shaking Palsy (1817) in which he reported in detail the symptoms of six patients. His description of the motor symptoms remains accurate and unchallenged.

Parkinson's is a global phenomenon being recognized in all cultures and is estimated to affect approximately 6.3 million worldwide. It is the second most common neurodegenerative disorder and an Australian report (2011) estimate that 1 in 350 Australians now have the condition, and more than 30 people are diagnosed daily.

Increasing age is unequivocally associated with an increased risk of Parkinson's. Incidence is reported as 1:1000 for people over 65 and 1:100 over 75. Although the condition is age related, it is distinct from the natural aging process.

The average age of diagnosis is 55 - 65 years. The term 'young onset' is attributed to those diagnosed between 21 - 40 and prior to this the term 'juvenile onset' is used. Parkinson's is slightly more common in males than females (ratio 5:4).

Parkinson's may affect anyone at any time. Well known identities diagnosed with the condition include Muhammad Ali, Michael J Fox, Janet Reno, Billy Graham, Bob Hoskins and the late Pope John Paul II and Donald Chipp. There is a theory that Adolf Hitler may have had Parkinson's.

The underlying cause in approximately 95% of those diagnosed remains unknown, hence the term Idiopathic Parkinson's disease. In the 1960s it was discovered that the symptoms are primarily related to a lack of a neurotransmitter (dopamine) as a result of degeneration of dopamine producing neurons within the substantia nigra in the basal ganglia in the mid-brain. Approximately 70% of the dopamine producing neurons are lost prior to the time of diagnosis therefore most people affected by the condition can retrospectively describe a gradual development of symptoms. More recently a naturally occurring protein (alpha-synuclein) has been identified as misfolding and aggregating in the form of Lewy bodies found at post mortem in cases of Parkinson's.

The cause of Parkinson's is a longstanding topic for worldwide research and many theories exist. The most commonly explored are:

- Environmental

- Oxidative stress
- Genes
- Multi-factorial

1.1 Problem Description

There are many research works going on Parkinson Disease(PD) which seemed to be the second most common disease in the world and it still more increasing now every day's. This situation leads to build a decision support system for PD. Now ever day's computational tools have been designed for helping the doctors to make decision about their patients.

Artificial Intelligence techniques are one of the necessary of physical visits to the clinic for monitoring and treatments are difficult. Widening access to the Internet and advanced telecommunication systems bandwidth offers the possibility of remote monitoring of patients, with substantial opportunities for lowering the inconvenience and cost of physical visits. However, in order to exploit these opportunities, there is the need for reliable clinical monitoring tools.

Speech pathologists have been trying to get their patients with Parkinson's disease to raise their voices for years. Although the condition is primarily characterized by tremors and difficulty in walking, most patients also suffer from speech problems, particularly slurring and what's known in the field as weak voice. While 89% of people with PD experience some type of speech problems. So if the classification percentage of Parkinson disease is high then it's possible to predict Parkinson in early stage.

Typically, the diagnosis is based on medical summary and neurological examination conducted by interviewing and observing the patient in person using the Unified Parkinson's Disease Rating Scale (UPDRS). It is very difficult to predict PD based on UDPRS in early stages, only 75% of clinical diagnoses of PD are confirmed to be idiopathic PD at autopsy. Thus, automatic techniques based on Artificial Intelligence are needed to increase the diagnosis is accuracy and to help doctors to make better decisions.

1.2 Objective

THE main focus of this paper is the classification of different types of datasets that can be performed to determine if a person is Parkinson affected or not. My work is an attempt to introduce a classification approach making comparison between Support Vector Machine-Sequential Minimal Optimization (SVM-SMO) and AdaBoost (M1 and M2) classifiers. The main motivation for this work is that Parkinson affects majority of the people in the world and it's a hard disease to diagnosis.

1.3 Thesis Outline

Since Parkinson is a very hard disease to diagnose clinically, so for my thesis paper, in chapter two I will go through all kinds of symptoms of Parkinson Disease. I will describe all kinds of major symptoms which causes Parkinson Disease. To classify Parkinson symptoms with classifiers, I will require data mining idea. So in my chapter I will also go through data mining technology. In addition, I will summarize all kinds of thesis paper related to Parkinson disease and data mining technique which I studied during my thesis work. Chapter three will contain description about my system design. It includes my overall work explanation, Dataset information about patients. Not only this, I will also include my preprocessing criteria about dataset. Algorithms related to my thesis classification will be described in this chapter. Since for classification I will use WEKA tool, so I will include a description about this tool set up. Chapter four will include my experiment and analysis of my work. In this chapter I will show all of the work procedures for classification and comparison table will be given. According to the WEKA classification result, result will be analyzed properly here. Finally in chapter five, conclusion with future work will be added in this paper. In future work part, I will include my future ambition on this work.

Chapter 2

BACKGROUND ANALYSIS

2.1 Parkinson Symptoms

The presentation of symptoms varies greatly between individuals diagnosed and no two people will be affected in the same way. The provisional medical diagnosis is based on symptoms because there is no definitive medical test or radiological procedure which diagnoses Parkinson's. The diagnostic criterion is composed of four cardinal symptoms which are:

- Tremor
- Bradykinesia
- Muscle rigidity
- Postural instability
- Festination of speech

Tremor is related to an imbalance of neurotransmitters, dopamine and acetylcholine, for this reason, tremor may be the least responsive symptom to dopamine replacement therapy. A classic tremor presentation of Parkinson's involves the thumb and first finger and is referred to as 'pill rolling'.

Bradykinesia affects all activities of daily living, walking, talking, swallowing and speaking. In the eyes and face it presents as a decreased blink rate and a lack of facial expression. The term used to describe slowness of thought experienced by people with Parkinson's is bradyphrenia.

Muscle rigidity is commonly present in the wrist, shoulder and neck. It may also manifest as a slightly flexed elbow on the affected side. Early reports of a painful shoulder are associated with increased muscle rigidity and tone. The pain associated with Parkinson's is often underestimated and reported, and is usually associated with muscle rigidity.

Postural instability and gait disturbances often develop later in the progression of the condition. If a loss of postural reflexes and resulting falls occur early, it is not suggestive of typical Parkinson's. In early Parkinson's, the posture may show a slight flexion of the neck or trunk with a slight lean to one side. Gait changes include reduced arm swing (unilateral) and shortened stride height and length which may lead to shuffling.

In addition to these cardinal motor symptoms there are many others which are also considered in the diagnostic process. Often the non-motor symptoms are more challenging for the person living with Parkinson's.

- Anosmia
- Anxiety
- Constipation
- Depression
- Fatigue
- Micrographia
- Swallowing changes
- Sweating

Parkinson Disease can have a profound effect on speech and voice[2]. Although symptoms vary widely from patient to patient, the speech symptoms most commonly demonstrated by patients with PD are reduced vocal loudness, monopitch, disruptions of voice quality, and abnormally fast rate speech[2]. This cluster of speech symptoms is often termed Hypokinetic Dysarthria. The most common symptom of Hypokinetic Dysarthria is Hypophonia, or reduced vocal loudness. Patients demonstrating this symptom may be unaware of the volume at which they are speaking and may require frequent requests to speak louder.

The symptoms can be very evident and is usually mild at the beginning and then get more complex and the functionality lost varies on several conditions. The list of signs and symptoms mentioned in various sources for Hypokinetic Dysarthria includes the 7 symptoms listed below[2]:

- Hoarse voice
- Breath voice
- Coarse voice
- Tremulous voice
- Single pitched voice
- Monotone voice
- Sudden pitch changes

Though the health care providers could diagnose the presence of Parkinson's disease based on the symptoms by the physical examination, the assess ability of the symptoms becomes difficult more particularly in case of elderly people [2]. As the illness progresses the signs like tremor, walking problem and speech variations becomes clearer. The main point that the diagnosis must concentrate on ruling out the other ailments that share the similar symptoms. The signs that need to be looked for are:

- Slow opening and inadequate closing of the vocal folds
- Slows down voluntary movements

2.2 Data Mining

Data mining, at its core, is the transformation of large amounts of data into meaningful patterns and rules. [3] Further, it could be broken down into two types: directed and undirected. In *directed* data mining, you are trying to predict a particular data point — the sales price of a house given information about other houses for sale in the neighborhood, for example.

In *undirected* data mining, you are trying to create groups of data, or find patterns in existing data — creating the "Soccer Mom" demographic group, for example. In effect, every U.S. census is data mining, as the government looks to gather data about everyone in the country and turn it into useful information.

For our purposes, modern data mining started in the mid-1990s, as the power of computing, and the cost of computing and storage finally reached a level where it was possible for companies to do it in-house, without having to look to outside computer powerhouses.

Additionally, the term data mining is all-encompassing, referring to dozens of techniques and procedures used to examine and transform data. Therefore, this series of articles will only scratch the surface of what is possible with data mining. Experts likely will have doctorates in statistics and have spent 10-30 years in the field. That may leave you with the impression that data mining is something only big companies can afford.

We hope to clear up many of these misconceptions about data mining, and we hope to make it clear that it is not as easy as simply running a function in a spreadsheet against a grid of data, yet it is not so difficult that everyone can't manage some of it themselves. This is the perfect example of the 80/20 paradigm — maybe even pushed further to the 90/10 paradigm. You can create a data-mining model with 90-percent effectiveness with only 10 percent of the expertise of one of these so-called data-mining experts. To bridge the remaining 10 percent of the model and create a perfect model would require 90-percent additional time and perhaps another 20 years. So unless you plan to make a career out of data mining, the "good enough" is likely all that you need. Looking at it another way, good enough is probably better than what you're doing right now anyway.

The ultimate goal of data mining is to create a model, a model that can improve the way you read and interpret your existing data and your future data. Since there are so many techniques with data mining, the major step to creating a good model is to determine what type of technique to use. That will come with practice and experience, and some guidance. From there, the model needs to be refined to make it even more useful. After reading these articles, you should be able to look at your data set, determine the right technique to use, then take steps to refine it. You'll be able to create a good-enough model for your own data.

2.3 Previous Work

For my thesis work, I had studied lots of thesis paper related to Parkinson disease and Data mining. According to the [2] paper written by Udaya Kumar, Magesh kumar, Parkinson's disease (PD) is degenerative illness whose cardinal symptoms include rigidity, tremor, and slowness of the movement. The speech symptoms most commonly demonstrated by patients with PD are reduced vocal loudness, monopitch, disruptions of voice quality. The aim of their paper was to predict the PD based on the audio files collected from various patients. Audio files were preprocessed in order to attain the features. The preprocessed data contains 23 attributes and 195 instances. On an average there were six voice recordings per person, By using data compression technique such as Discrete Cosine Transform (DCT) number of instances could be minimized, after data compression, attribute selection was done using several WEKA build in methods such as ChiSquared, GainRatio, Infogain after identifying the important attributes, they evaluated attributes one by one using stepwise regression. Based on the selected attributes they processed in WEKA by using cost sensitive classifier with various algorithms like MultiPass LVQ, Logistic Model Tree (LMT), K-Star. The classified result showed on an average 80%. By using those features 95% approximate classification of PD is achieved. This showed that using the audio dataset, PD could be predicted with a higher level of accuracy.

I have studied another thesis paper regarding Parkinson Disease [4] written by Anchana Khemphila and Veera Boonjing. According to the paper, the Multi-Layer Perceptron (MLP) with Back-Propagation learning algorithm was used to classify to effective diagnosis Parkinson disease (PD). It was a challenging problem for medical community. Typically characterized by tremor, PD occurs due to the loss of dopamine in the brains thalamic region that results in involuntary or oscillatory movement in the body. A feature selection algorithm along with biomedical test values to diagnose Parkinson disease. Clinical diagnosis was done mostly by doctor's expertise and experience. But still cases were reported of wrong diagnosis and treatment. Patients were asked to take number of tests for diagnosis. In many cases, not all the tests contribute towards effective diagnosis of a disease. Their work is to classify the presence of Parkinson disease with reduced number of attributes. Original, 22 attributes were involved in classify. They used Information Gain to determine the attributes which reduced the number of attributes which is need to be taken from patients. The Artificial neural networks were used to classify the diagnosis of patients. Twenty-Two attributes were reduced to sixteen attributes. The accuracy of training data set was 82.051% and in the validation data set is 83.333%.

Beside these two thesis papers, I have studied another thesis paper regarding WEKA classification technique using 10-fold cross validation. This thesis paper was for diabetic patients analyzed in WEKA tool [5] written by P.Yasodha and M. Kannan. According to the paper, Data mining is an important tool in many areas of research and industry. Companies and organizations are increasingly interested in applying data mining tools to increase the value added by their data collections systems. Nowhere is this potential more important than in the healthcare industry. As medical records systems become more standardized and commonplace, data quantity increases with much of it going unanalyzed. They aimed at finding out the characteristics that determine the presence of diabetes and to track the

maximum number of men and women suffering from diabetes with 249 population using WEKA tool. In this paper the data classification of diabetic patients data set was developed by collecting data from hospital repository consists of 249 instances with 7 different attributes. The instances in the Dataset were pertaining to the two categories of blood tests, urine tests. WEKA tool was used to classify the data and the data was evaluated using 10-fold cross validation and the results were compared.

CHAPTER 3

SYSTEM DESIGN

3.1 Dataset Information

The Parkinson database used in this study is taken from the University of California at Irvine (UCI) machine learning repository [6]. The dataset was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

Table1: Parkinson Dataset Attribute Information

Attribute	Type	Description
Name	Class	ASCII subject name and recording number
MDVP:Fo(Hz)	Numerical	Average vocal fundamental frequency
MDVP:Fhi(Hz)	Numerical	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Numerical	Minimum vocal fundamental frequency
MDVP:Jitter(%) MDVP:Jitter(Abs)	Numerical	Several measures of variation in fundamental frequency

MDVP:RAP MDVP:PPQ Jitter:DDP		
MDVP:Shimmer MDVP:Shimmer(dB) Shimmer:APQ3 Shimmer:APQ5 MDVP:APQ Shimmer:DDA	Numerical	Several measures of variation in amplitude
NHR HNR	Numerical	Two measures of ratio of noise to tonal components in the voice
RPDE D2	Numerical	Two nonlinear dynamical complexity measures
DFA	Numerical	Signal fractal scaling exponent
spread1 spread2 PPE	Numerical	Three nonlinear measures of fundamental frequency variation
Status	Numerical	Health status of the subject (one) - Parkinson's, (zero) - healthy

3.2 Preprocessing the dataset

Practical exploitation of the information in the measures calculated [7] requires constructing feature vectors from these measures, which can then be subsequently used to discriminate healthy from PWP. SVM classification performance is greatly enhanced by pre-processing of the values of each measure with an appropriate rescaling [8]. Here scaling each measure is such that, over all signals, the measure occupies the numerical range $[-1, 1]$.

Also in this stage, wishing to filter the number of measures down to a manageable size, such that a full search of all possible combinations can be conducted [9] in order to determine the optimal set for classification. It is noted that many of the measures will be highly correlated with other measures. This is because they will be measuring very similar aspects of the speech signal; for example, Jitter(%) and Jitter(Abs) [10] are derived from pitch period sequences and measure the average absolute temporal differences in these periods. Because of this correlation, only one of this pair of measures will contribute useful information for the classification stage, and the other should be removed.

It is therefore systematically searched through all pairs of features. Of those that are highly correlated (with a correlation coefficient of greater than 0.95), it is removed one of the pair.

It is then constructed feature vectors with each possible combination of subsets of pre-processed, filtered measures. To each combination, it is applied SVM classification. This is a direct measure of the practical separability of the classes.

Prior visual inspection of the layout and clustering of pairs of measures indicate that the optimal decision boundaries separating healthy from PWP may not be simple lines or hyperplanes. Because of this, it is used the kernel-SVM formulation, with *Gaussian radial basis kernel* functions [8]. These are flexible kernels that allow smooth, curved decision boundaries. For each combination of features, the classification performance is assessed in terms of the overall number of subjects correctly classified as healthy or PD, the number of PWP correctly classified (the *true positive rate*), and the number of healthy subjects correctly classified (the *true negative rate*). Validation of the results to obtain an estimate of out-of-sample performance and confidence intervals is assessed using *bootstrap resampling* with 50 replicates [8]. The choice of optimal SVM penalty value and kernel bandwidth is determined by exhaustive search over a range of values.

The bootstrap classification produces a set of classification performance results for each bootstrap replicate. In order to determine the best performing subset of features, it is compared the sets of overall classification results using the two-sided Wilcoxon rank-sum test against the null hypothesis of equal medians, at a significance probability of 0.05.

3.3 Algorithms Explanation

3.3.1 Ada-Boost Classifier

Boosting refers to a general and provably effective method of producing a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb [11]. Boosting has its roots in a theoretical framework for studying machine learning called the “PAC” learning model, due to Valiant [12]; see Kearns and Vazirani [13] for a good introduction to this model.

The AdaBoost algorithm, introduced in 1995 by Freund and Schapire, which solved many of the practical difficulties of the earlier boosting algorithms (which came up with the first provable polynomial-time in 1989).

For example, if we want to predict which person has Parkinson disease or not based on the symptoms, we can get a good prediction using Ada-Boost classifier. The algorithm takes as input $(x_1, y_1), \dots, (x_m, y_m)$ where each x_i belongs to some domain or instance space X and each level y_i in some level set Y . In most cases, we assume $Y = \{-1, +1\}$. AdaBoost calls a given *weak* or base learning algorithm repeatedly in a series of rounds $t = 1 \dots T$. The algorithm will maintain a distribution or set of weights over the training set. The weight of this distribution on training example i on round t is denoted by $D_t(i)$. At first stage all weights set equally, but on each round, the weights of misclassified examples are increased so that the weak learners is forced to focus on hard examples in the training set.

The weak learner’s job is to find out the weak hypothesis

$h_t : X \rightarrow \{-1, +1\}$ appropriate for the distribution D_t .

The goodness of the weak hypothesis is measured by its error:

Here the error is measured with respect to the distribution D_t , on which the weak learner was trained.

Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialize $D_1(i) = 1/m$

For $t \dots T$:

-Train weak learners using distribution D_t

-Get weak hypothesis $h_t : X \rightarrow \{-1, +1\}$ with error

$\epsilon_t = P_{r_i \sim D_t} [h_t(x_i) \neq y_i]$

-Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$

-Update:

$$D_{t+1} = \frac{D_t(i)}{Z_t} \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output of the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

There are two versions of the algorithm which we denote AdaBoost.M1 and AdaBoost.M2. The two versions are equivalent for binary classification problems and differ only in their handling of problems with more than two classes.

3.3.1.1 Ada-Boost.M1

This boosting algorithm takes as input a training set of N examples $S = ((x_1, y_1), \dots, (x_m, y_m))$ where x_i is an instance drawn from some space X and represented in some manner (typically, a vector of attribute values), and $y_i \in Y$ is the class label associated with x_i . The set of possible labels Y is of finite cardinality k .

In addition, the boosting algorithm has access to another unspecified learning algorithm, called the weak learning algorithm, which is denoted generically as **WeakLearn**. The boosting algorithm calls **WeakLearn** repeatedly in a series of rounds. On round t , the booster provides **WeakLearn** with a distribution D_t over the training set S . In response, **WeakLearn** computes a classifier or *hypothesis* $h_t: X \rightarrow Y$ which should correctly classify a fraction of the training set that has large probability with respect to D_t . That is, the weak learner's goal is to find a hypothesis h_t which minimizes the (training) error $\epsilon_t = Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$. Note that this error is measured with respect to the distribution D_t that was provided to the weak learner. This process continues for T rounds, and, at last, the booster combines the weak hypotheses h_1, \dots, h_T into a single final hypothesis

h_{fin} .

Still unspecified are: (1) the manner in which D_t is computed on each round, and (2) how h_{fin} is computed. Different boosting schemes answer these two questions in different ways. The initial distribution D_1 is uniform over S so $D_1(i) = 1/m$ for all i . To compute distribution D_{t+1} from D_t and the last weak hypothesis h_t , we multiply the weight of example i by some number $\beta_t = [0,1)$ if h_t classifies correctly, and otherwise the weight is left unchanged. The weights are then renormalized by dividing by the normalization constant Z_t .

Effectively, “easy” examples that are correctly classified by many of the previous weak hypotheses get lower weight, and “hard” examples which tend often to be misclassified get higher weight. Thus, **AdaBoost** focuses the most weight on the examples which seem to be hardest for **WeakLearn**.

The *training error* of the final hypothesis generated by **AdaBoost.M1** is small. This does not necessarily imply that the *test error* is small. However, if the weak hypotheses are “simple” and T “not too large,” then the difference between the training and test errors can also be theoretically bounded. The experiments indicate that the theoretical bound on the training error is often weak, but generally correct qualitatively. However, the test error tends to be much better than the theory would suggest, indicating a clear defect in our theoretical understanding.

The main disadvantage of **AdaBoost.M1** is that it is unable to handle weak hypotheses with error greater than $1/2$. The expected error of a hypothesis which randomly guesses the label is $1 - 1/k$, where k is the number of possible labels. Thus, for $k=2$, the weak hypotheses need to be just slightly better than random guessing, but when $k > 2$, the requirement that the error be less than $1/2$ is quite strong and may often be hard to meet.

3.3.1.2 Ada-Boost.M2

The second version of **AdaBoost** attempts to overcome this difficulty by extending the communication between the boosting algorithm and the weak learner. First, it allows the weak learner to generate more expressive hypotheses, which, rather than identifying a single label in Y , instead choose a set of “plausible” labels. This may often be easier than choosing just one label. For instance, in an OCR setting, it may be hard to tell if a particular image is “7” or a “9”, but easy to eliminate all of the other possibilities. In this case, rather than choosing between 7 and 9, the hypothesis may output the set $\{7, 9\}$ indicating that both labels are plausible.

It also allows the weak learner to indicate a “degree of plausibility.” Thus, each weak hypothesis outputs a vector $[0,1]^k$, where the components with values close to 1 or 0 correspond to those labels considered to be plausible or implausible, respectively. Note that this vector of values is *not* a probability vector, i.e., the components need not sum to one.

While it gives the weak learning algorithm more expressive power, it also places a more complex requirement on the performance of the weak hypotheses. Rather than using the usual prediction error, it asks that the weak hypotheses do well with respect to a more sophisticated error measure that it calls the *pseudo-loss*. Unlike ordinary error which is computed with respect to a distribution over examples, pseudo-loss is computed with respect to a distribution

over the set of all pairs of examples and incorrect labels. By manipulating this distribution, the boosting algorithm can focus the weak learner not only on hard-to-classify examples, but more specifically, on the incorrect labels that are hardest to discriminate. It will be seen that the boosting algorithm AdaBoost.M2, which is based on these ideas, achieves boosting if each weak hypothesis has pseudo-loss slightly better than random guessing.

More formally, a mislabel is a pair (i, y) where i is the index of a training example and y is an incorrect label associated with example i . Let B be the set of all mislabels: $B = \{(i, y) : i \in \{1, \dots, m\}, y \neq y_i\}$. A mislabel distribution is a distribution defined over the set B of all mislabels.

On each round t of boosting, AdaBoost.M2 supplies the weak learner with a mislabel distribution D_t . In response, the weak learner computes a hypothesis h_t of the form $h_t: X \times Y \rightarrow [0, 1]$.

There is no restriction on $\sum_y h_t(x, y)$. In particular, the prediction vector does not have to define a probability distribution.

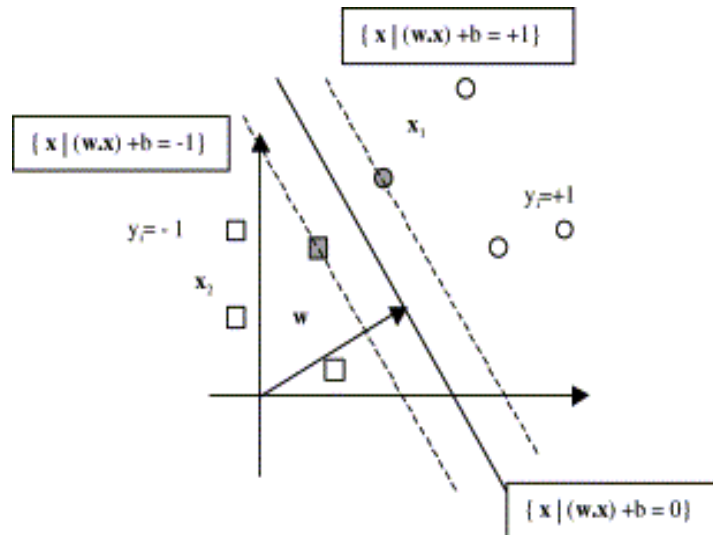
The interpretation leads to define the pseudo-loss of hypothesis h_t with respect to mislabel distribution D_t by the formula

$$\epsilon_t = \frac{1}{2} \sum_{(i, y) \in B} D_t(i, y) (1 - h_t(x_i, y_i) + h_t(x_i, y))$$

It can be verified though that the pseudo-loss is minimized when correct labels y_i are assigned the value 1 and incorrect labels $y \neq y_i$ assigned the value 0. Further, note that pseudo-loss $1/2$ is trivially achieved by any constant-valued hypothesis h_t . The weak learner's goal is to find a weak hypothesis h_t with small pseudo-loss. Thus, standard "off-the-shelf" learning algorithms may need some modification to be used in this manner, although this modification is often straight-forward. After receiving h_t , the mislabel distribution is updated using a rule similar to the one used in AdaBoost.M1.

3.3.2 Support vector Machine with Sequential Minimal Optimization classifier

SMO (Sequential Minimal Optimization) solves the SVM QP (support vector machine quadratic programming) problem by decomposing it into SVM QP Sub-problems and solving the smallest possible optimization problem, involving the two Lagrange multipliers, at each step. A Quadratic Problem is actually maximizing or minimizing a quadratic objective function subject to a set of linear constraints.

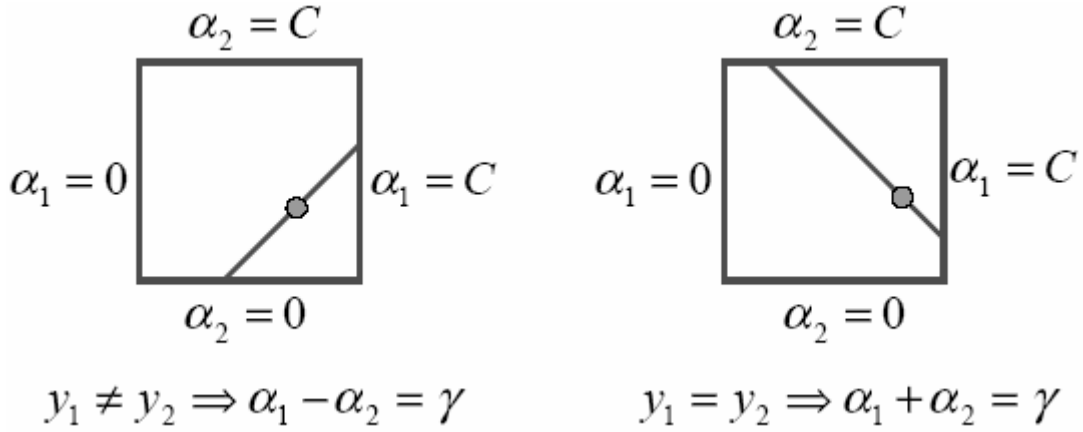


As above mentioned, SMO involves two Lagrange multipliers, so at each step, SMO chooses them to jointly optimize, finds the optimal values for these multipliers and updates the SVM to reflect the new optimal values.[14] The advantage of SMO is numerical QP optimization is avoided entirely here. Instead of this, solving for two Lagrange multipliers can be done analytically in SMO. Thus, the inner loop of the algorithm can be expressed in a short amount of C code, rather than invoking an entire QP library routine. Though more optimization sub-problems are solved through SMO, but each sub-problem is so fast that overall QP problem is solved quickly. Another advantage is that SMO doesn't require extra matrix at all. Since no matrix algorithms are used in SMO, so very large SVM training problems can fit inside of the memory of an ordinary personal computer or workstation.

There are two constraints for solving two Lagrange multipliers.

One is Bound Constraints $0 \leq \alpha_i \leq C$

Second one is Equality Constraints $\sum \alpha_i y_i = 0$



Equations:

$$L = \max(0, \alpha_2 - \alpha_1), \quad H = \min(C, C + \alpha_2 - \alpha_1).$$

$$L = \max(0, \alpha_2 + \alpha_1 - C), \quad H = \min(C, \alpha_2 + \alpha_1).$$

$$\eta = K(\vec{x}_1, \vec{x}_1) + K(\vec{x}_2, \vec{x}_2) - 2K(\vec{x}_1, \vec{x}_2).$$

$$\alpha_2^{new} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}.$$

$$\alpha_2^{new,clipped} = \begin{cases} H & \text{if } \alpha_2^{new} \geq H; \\ \alpha_2^{new} & \text{if } L < \alpha_2^{new} < H; \\ L & \text{if } \alpha_2^{new} \leq L. \end{cases}$$

$$\alpha_1^{new} = \alpha_1 + s(\alpha_2 - \alpha_2^{new,clipped}).$$

The threshold b is re-computed after each step, so that the KKT conditions are fulfilled for both optimized examples. The following threshold b_1 is valid when the new \mathbf{a}_1 is not at the bounds, because it forces the output of the SVM to be y_1 when the input is \mathbf{x}_1 :

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1)k(\vec{x}_1, \vec{x}_1) + y_2(\alpha_2^{new,clipped} - \alpha_2)k(\vec{x}_1, \vec{x}_2) + b.$$

The following threshold b_2 is valid when the new \mathbf{a}_2 is not at bounds, because it forces the output of the SVM to be y_2 when the input is \mathbf{x}_2 :

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1)k(\vec{x}_1, \vec{x}_2) + y_2(\alpha_2^{new,clipped} - \alpha_2)k(\vec{x}_2, \vec{x}_2) + b.$$

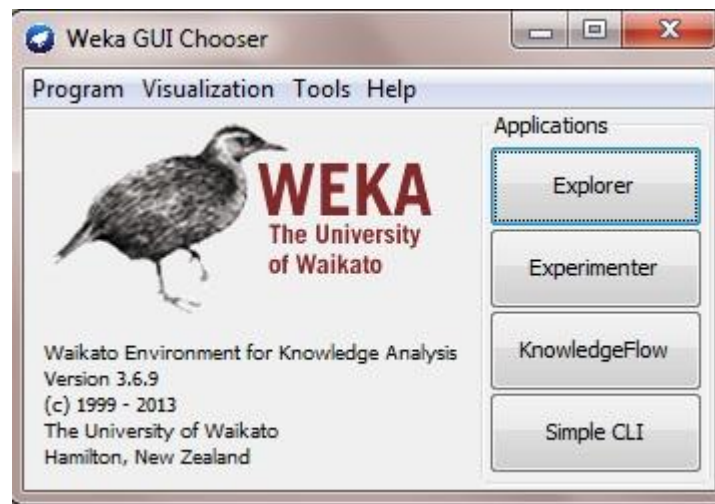
When both b_1 and b_2 are valid, they are equal. When both new Lagrange multipliers are at bound and if L is not equal to H , then the interval between b_1 and b_2 are all thresholds that are consistent with the KKT conditions. SMO chooses the threshold to be halfway in between b_1 and b_2 .

3.3.3 WEKA Set Up

Data mining isn't solely the domain of big companies and expensive software. [3] In fact, there's a piece of software that does almost all the same things as these expensive pieces of software — the software is called WEKA. WEKA is the product of the University of Waikato (New Zealand) and was first implemented in its modern form in 1997. It uses the GNU General Public License (GPL). The software is written in the Java™ language and contains a GUI for interacting with data files and producing visual results (think tables and curves). It also has a general API, so you can embed WEKA, like any other library, in your own applications to such things as automated server-side data-mining tasks.

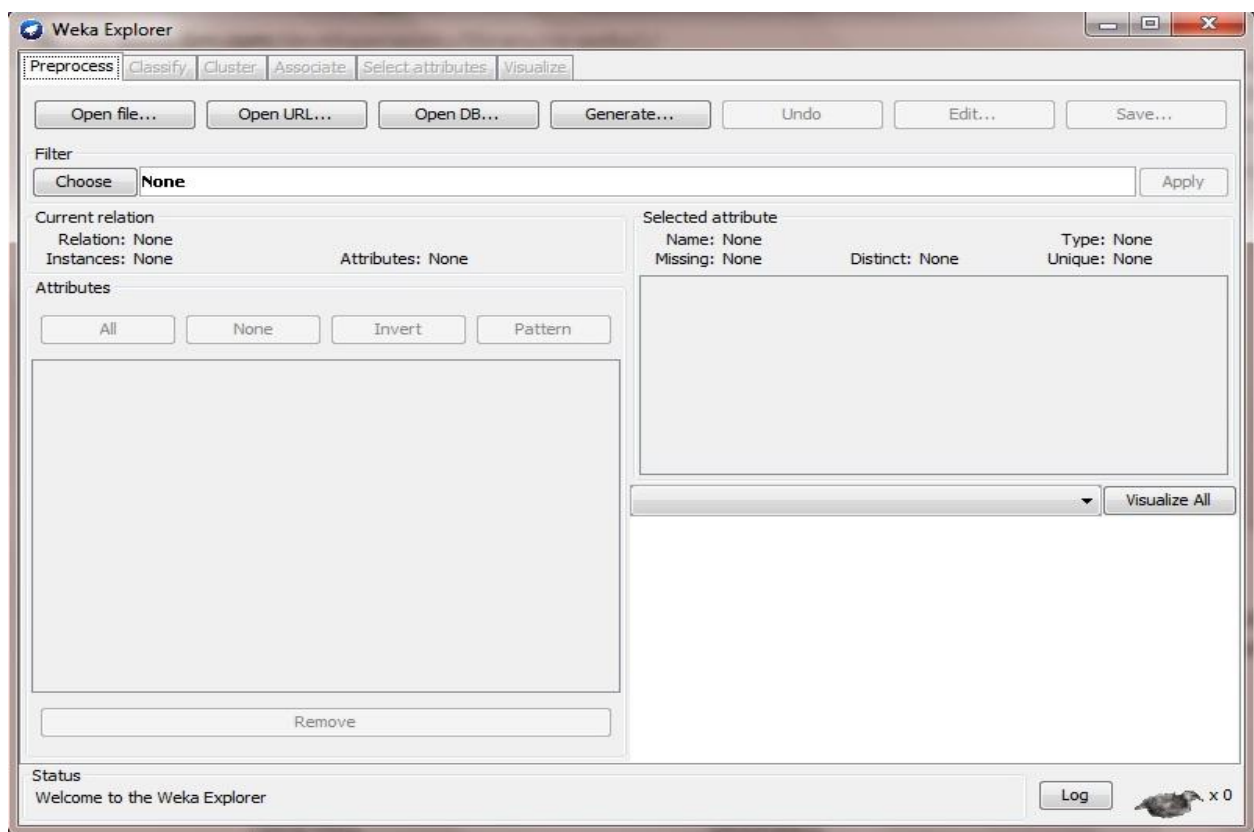
It's Java-based, so if one doesn't have a JRE installed on your computer, he should download the WEKA version that contains the JRE, as well.

Figure 1: WEKA startup screen



When one starts WEKA, the GUI chooser pops up and lets him choose four ways to work with WEKA and his data. For my work, I will choose only the **Explorer** option. This option is more than sufficient for everything.

Figure 2: WEKA Explorer



Chapter 4

EXPERIMENT & ANALYSIS

4.1 ARFF file formation

Since the dataset is already preprocessed, so for classification I have to make a arff file from the Parkinson training examples.

```
@relation parkinsons

@attribute class {phon_R01_S01_1,phon_R01_S01_2,phon_R01_S01_3,phon_R01_S01_4,phon_R01_S01_5,phon_R01_S01_6,phon_R01_S02_1,phon_R01_S02_2,phon_R01_S02_3,phon_R01_S02_4,phon_R01_S02_5,phon_R01_S02_6,phon_R01_S04_1,phon_R01_S04_2,phon_R01_S04_3,phon_R01_S04_4,phon_R01_S04_5,phon_R01_S04_6,phon_R01_S05_1,phon_R01_S05_2,phon_R01_S05_3,phon_R01_S05_4,phon_R01_S05_5,phon_R01_S05_6,phon_R01_S06_1,phon_R01_S06_2,phon_R01_S06_3,phon_R01_S06_4,phon_R01_S06_5,phon_R01_S06_6,phon_R01_S07_1,phon_R01_S07_2,phon_R01_S07_3,phon_R01_S07_4,phon_R01_S07_5,phon_R01_S07_6,phon_R01_S08_1,phon_R01_S08_2,phon_R01_S08_3,phon_R01_S08_4,phon_R01_S08_5,phon_R01_S08_6,phon_R01_S10_1,phon_R01_S10_2,phon_R01_S10_3,phon_R01_S10_4,phon_R01_S10_5,phon_R01_S10_6,phon_R01_S13_1,phon_R01_S13_2,phon_R01_S13_3,phon_R01_S13_4,phon_R01_S13_5,phon_R01_S13_6,phon_R01_S16_1,phon_R01_S16_2,phon_R01_S16_3,phon_R01_S16_4,phon_R01_S16_5,phon_R01_S16_6}
```

Figure 3: ARFF file format

Since my collected dataset is already preprocessed, so according to those data I need to make a arff file. There are some tags which have to be followed for making this. In Figure 3, I made a relation tag and I gave relation name “parkinsons”. Here one thing should be noticed that my saved arff file name and relation name should be the same. After the relation tag I gave attribute tag. All the attributes and their types I have to define under attribute tag. Since in the dataset, all the voice recording instances are numbered to identify each voice sample for example: phon_R01_s01, phon_R01_s02 etc, these data I kept in class attribute according to the arff file format.

```

@attribute RPDE numeric
@attribute D2 numeric
@attribute DFA numeric
@attribute spread1 numeric
@attribute spread2 numeric
@attribute PPE numeric

@attribute status {1,0}

@data

phon_R01_S01_
1,119.99200,157.30200,74.99700,0.00784,0.00007,0.00370,0.00554,0
.01109,0.04374,0.42600,0.02182,0.03130,0.02971,0.06545,0.02211,2
1.03300,0.414783,0.815285,-4.813031,0.266482,2.301442,0.284654,1

phon_R01_S01_
2,122.40000,148.65000,113.81900,0.00968,0.00008,0.00465,0.00696,
0.01394,0.06134,0.62600,0.03134,0.04518,0.04368,0.09403,0.01929,
19.08500,0.458359,0.819521,-
4.075192,0.335590,2.486855,0.368674,1

phon_R01_S01_
3,116.68200,131.11100,111.55500,0.01050,0.00009,0.00544,0.00781,

```




Figure 4: ARFF file format

Since I have 23 attributes in my dataset, first attribute was class type (figure 3) and the rest of the attributes are numerical type because they have numerical data values. For example, I gave MDVP_Fo, MDVP_Fhi, MDVP_Flo, MDVP_Jitter_1, MDVP_Jitter_2, MDVP_RAP, MDVP_PPQ, Jitter_DDP, MDVP_Shimmer_1, MDVP_Shimmer_2, Shimmer_APQ3, Shimmer_APQ5, MDVP_APQ, Shimmer_DDA, NHR, HNR, RPDE, D2, DFA, spread1, spread2, PPE all these attributes “Numerical” type. Attribute status will be numerical 0 or 1 because this attribute will determine whether the person is Parkinson affected or not according to the data values. In “data” tag section I organized all the attributes value according to the attribute sequence. Since my dataset contains 195 voice recording samples, I included all the instances in my arff file.

4.2 Visualization of the attributes

The below picture shows a histogram for the attribute distributions for a single selected attribute at a time, by default this is class attribute. Here, the individual colors indicate the individual classes.

X-axis denotes the feature values. The feature value ranges according to the respective feature range. These ranges are split using mean value. Y-axis denotes the number of people lie in- between that range.

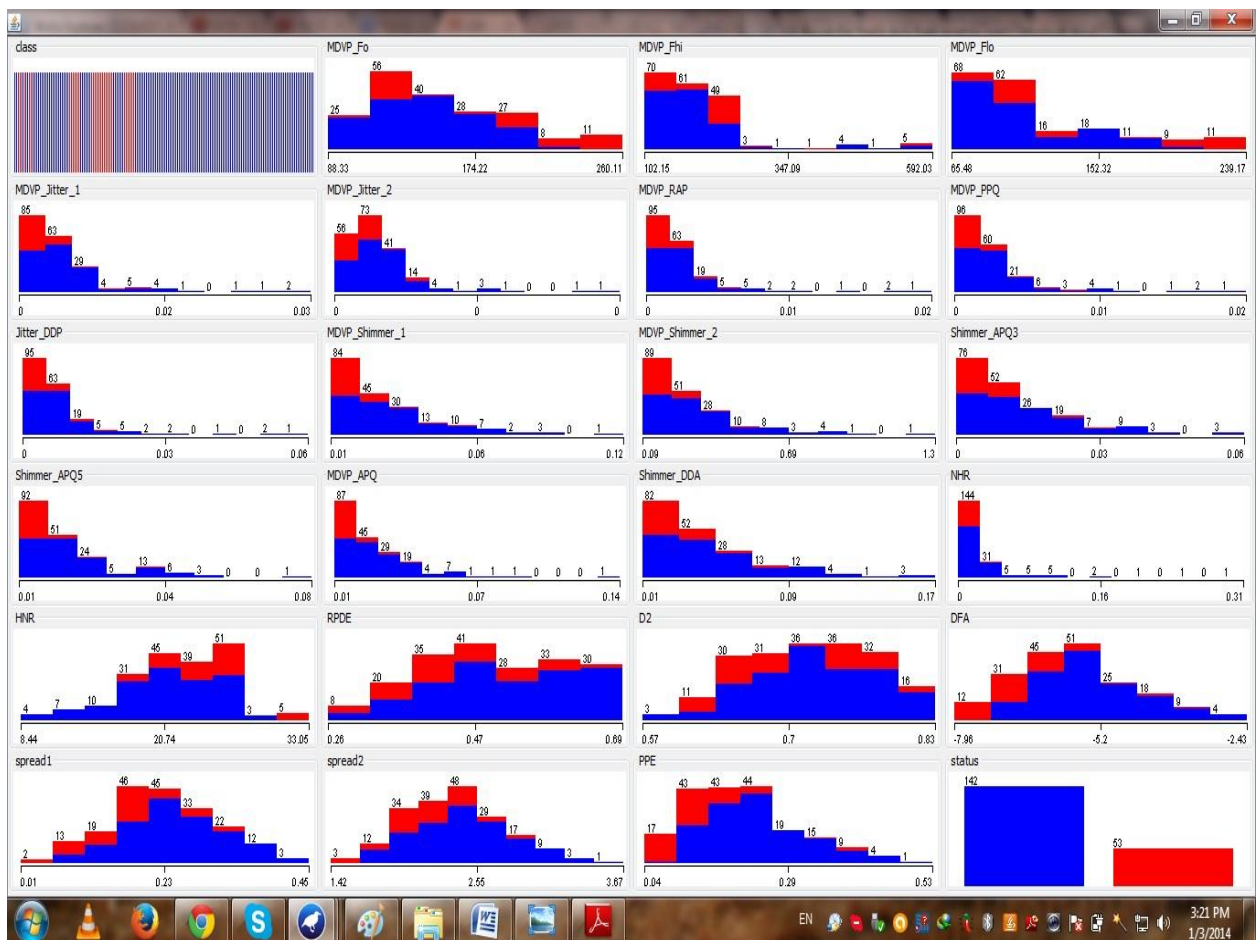


Figure 5: Visualizing all the Attributes

The table listed earlier is displayed diagrammatically here where the values of the attributes are shown as results of weka. The visualization clearly explains the value of each attribute along with its range of the value that the attribute gives. All the 23 attributes used are mentioned here.

4.3 AdaBoost WEKA classification

AdaBoost is a binary/dichotomous/2-class classifier and designed to boost a weak learner that is just better than $1/2$ accuracy. There are two versions of the algorithm which is denoted by AdaBoost.M1 and AdaBoost.M2. The two versions are equivalent for binary classification problems and differ only in their handling of problems with more than two classes.

AdaBoostM1 is a M-class classifier but still requires the weak learner to be better than $1/2$ accuracy, when one would expect chance level to be around $1/M$. The training error of the final hypothesis generated by AdaBoost.M1 is small but the main disadvantage of AdaBoost.M1 is that it is unable to handle weak hypothesis with error greater than $1/2$. The second version of AdaBoost which is AdaBoost.M2 which attempts to overcome this difficulty by extending the communication between the boosting algorithm and the weak learner. In AdaBoost.M2 version, the weak hypothesis do well with respect to a more sophisticated error measure that we call the pseudo-loss. Unlike ordinary error which is computed with respect to a distribution over examples, pseudo-loss is computed with respect to a distribution over the set of all pair of examples and incorrect levels. By manipulating this distribution, the boosting algorithm can focus the weak learner not only on hard to classify examples, but more specifically, on the incorrect levels that are hardest to discriminate.

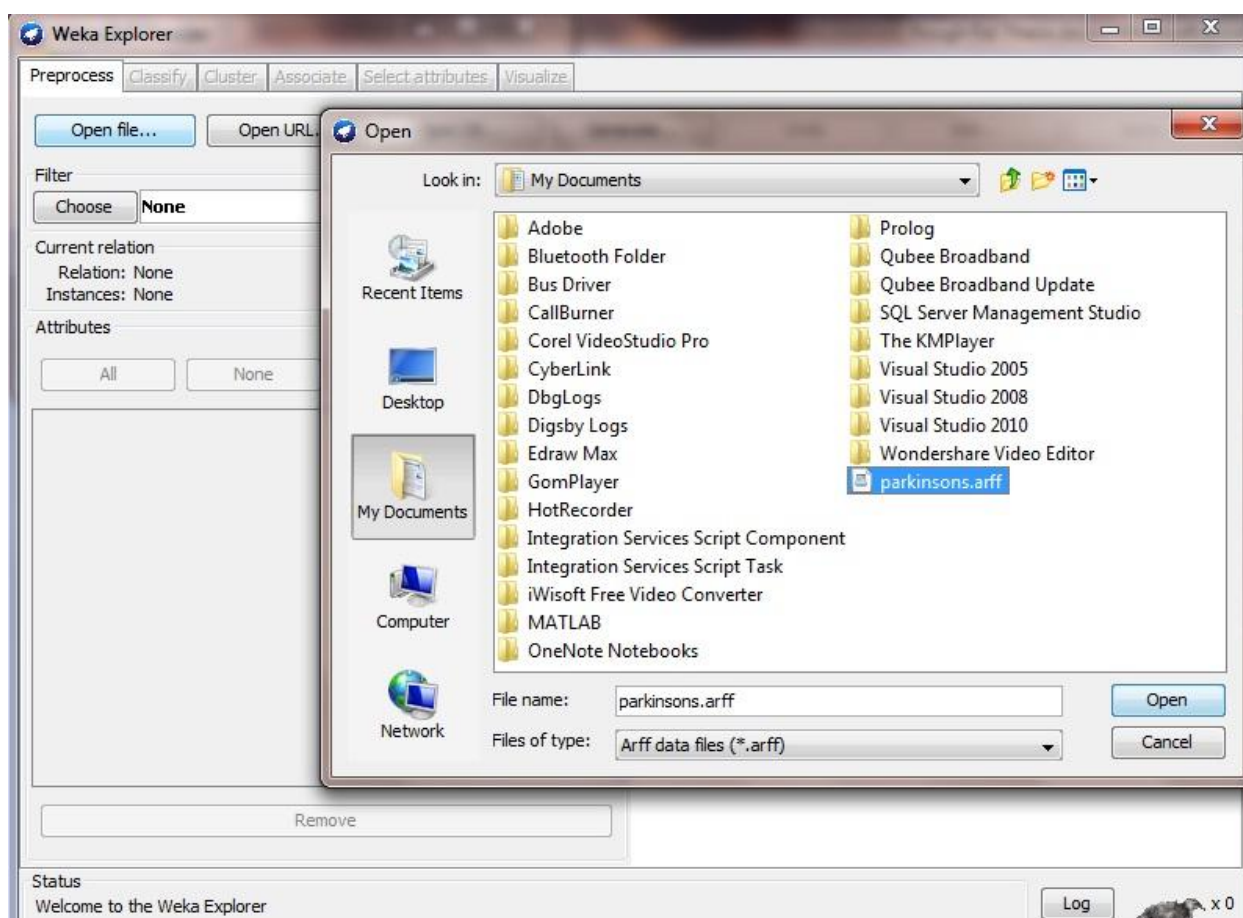


Figure 6 : parkinsons.arff file opening

At first, the arff file is opened in WEKA tool. According to the figure 5, I opened the parkinsons.arff file in WEKA.

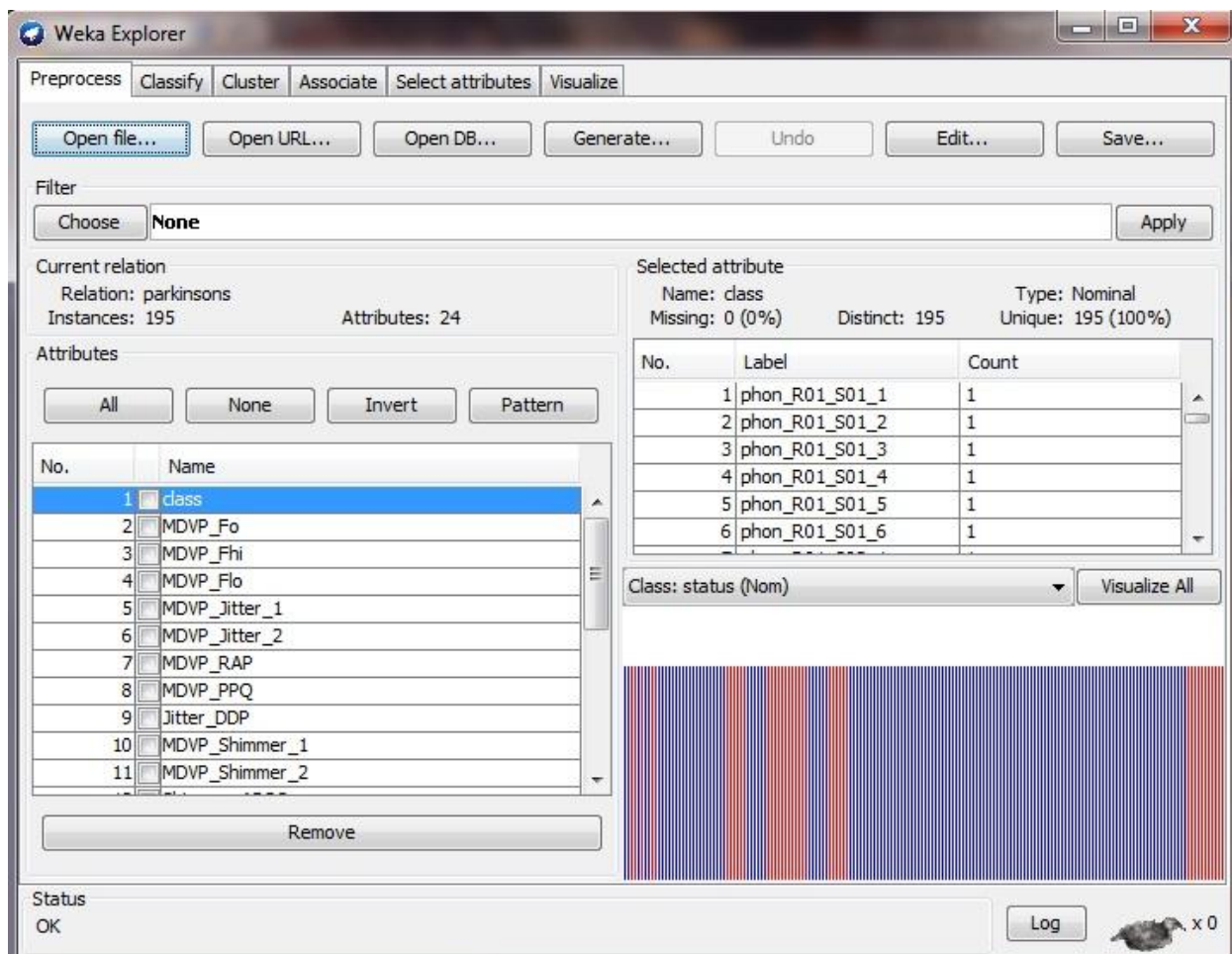


Figure 7: After opening arff file in WEKA tool

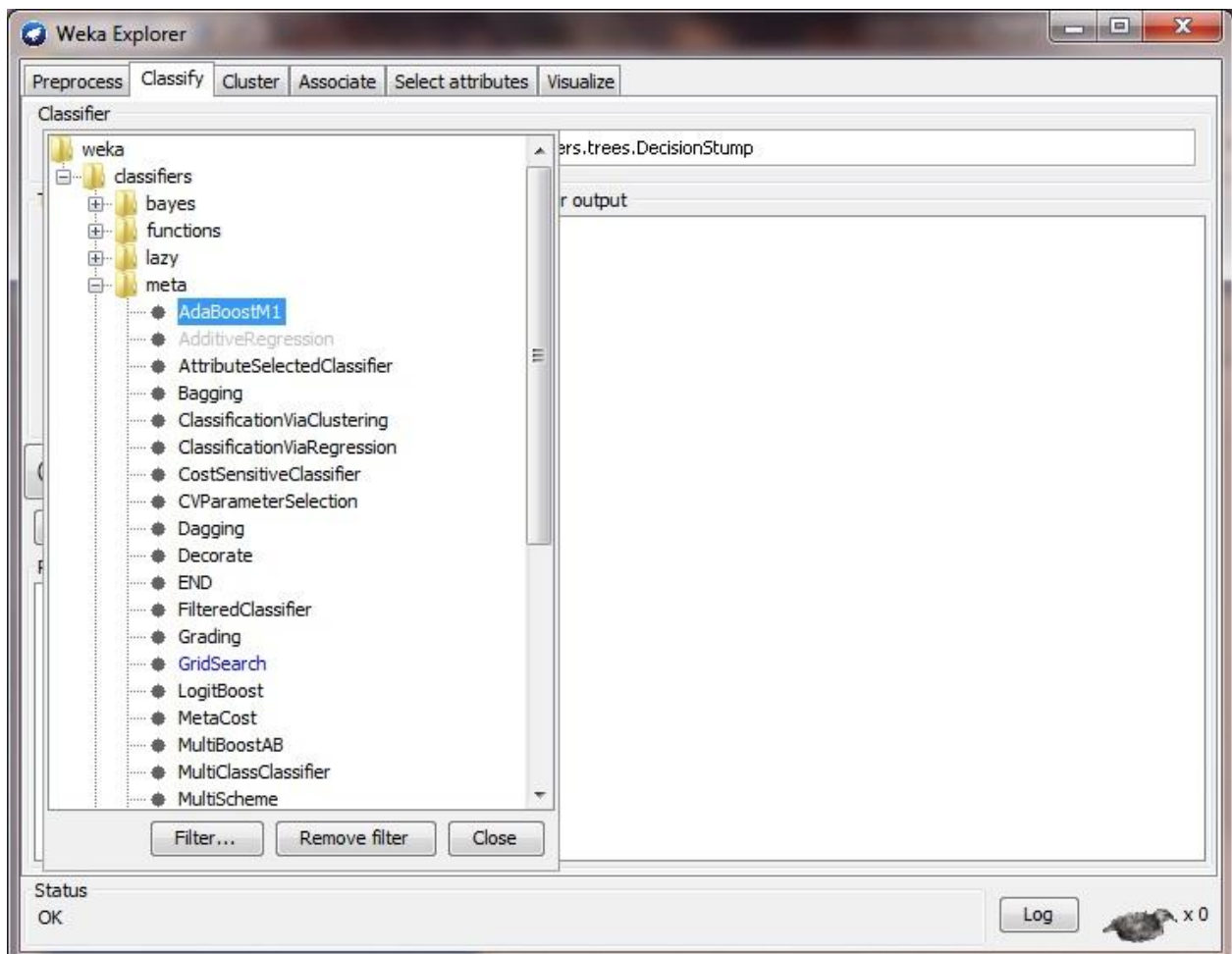


Figure 8: Selecting AdaBoostM1 from meta classifiers

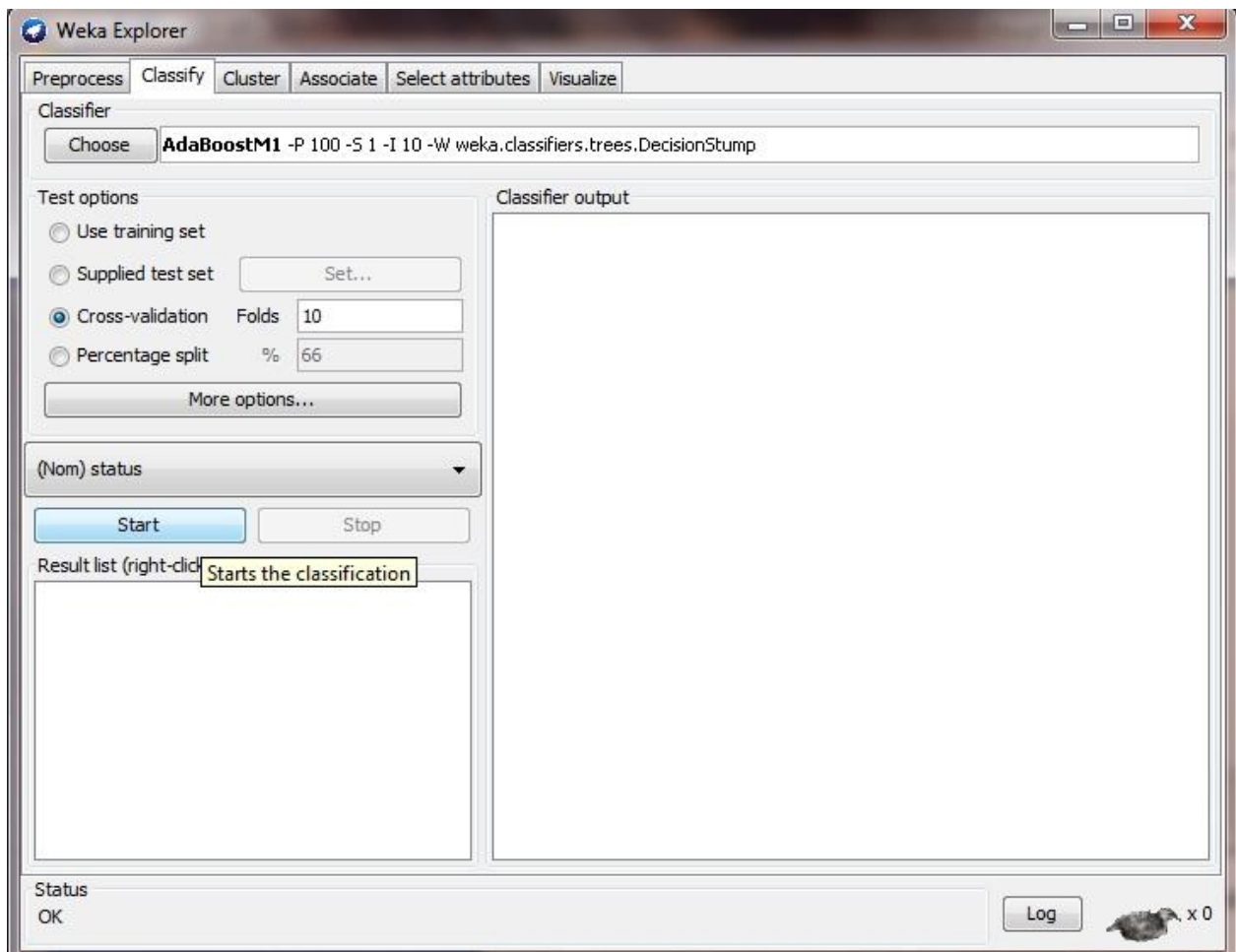


Figure 9: Setting cross validation and starting

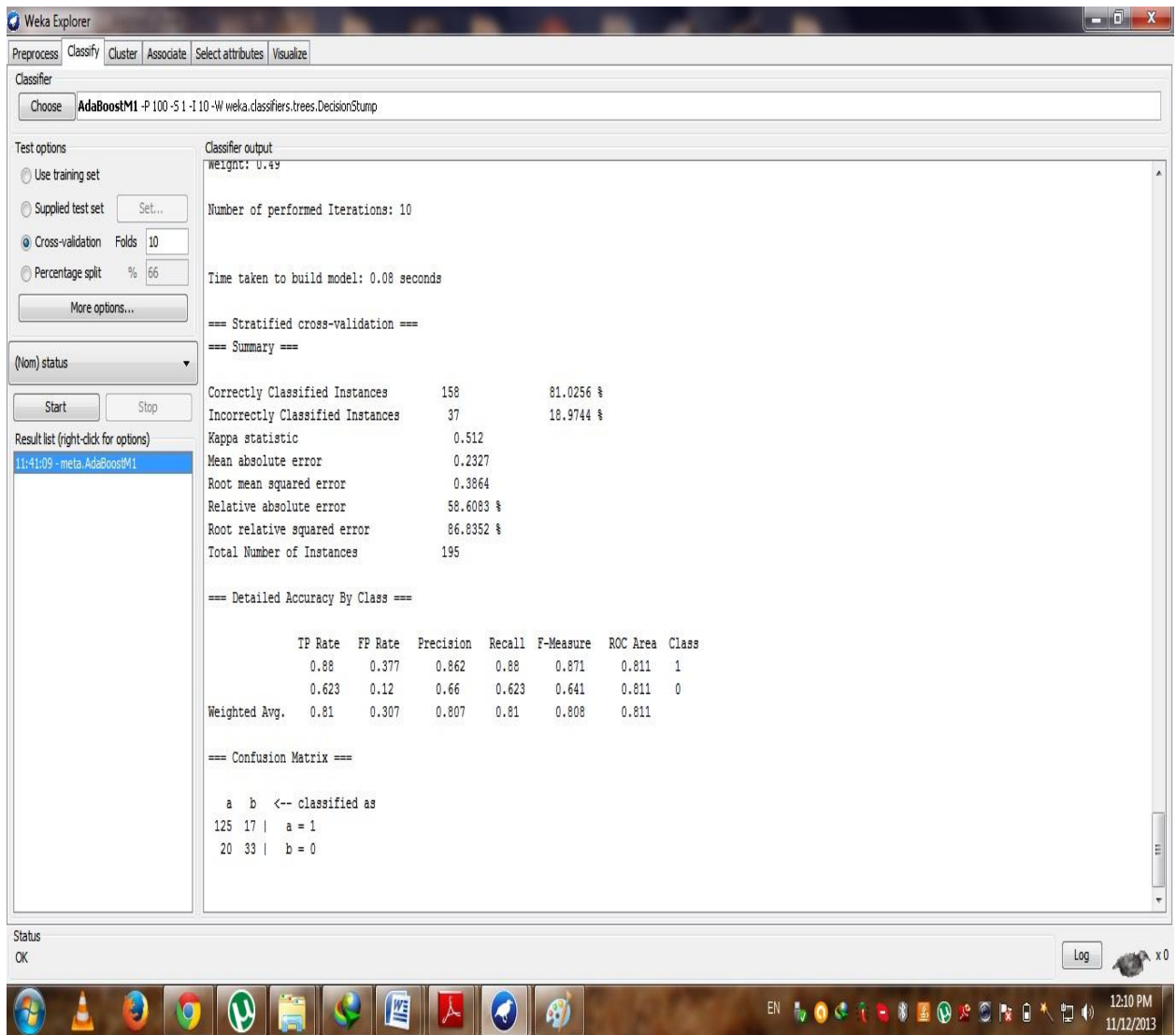


Figure 10: AdaBoostM1 Weka Classification result

4.4 SVM-SMO WEKA classification

Sequential Minimal Optimization (SMO) is one of the most popular algorithms for large-margin classification by Support Vector Machine (SVM). In WEKA, SMO implements John Platt's sequential minimal optimization (SMO) algorithm for training a support vector classifier and multi-class problems are solved using pairwise classification. To obtain proper probability estimates, SMO uses the option that fits logistic regression models to the outputs of the support vector machine.

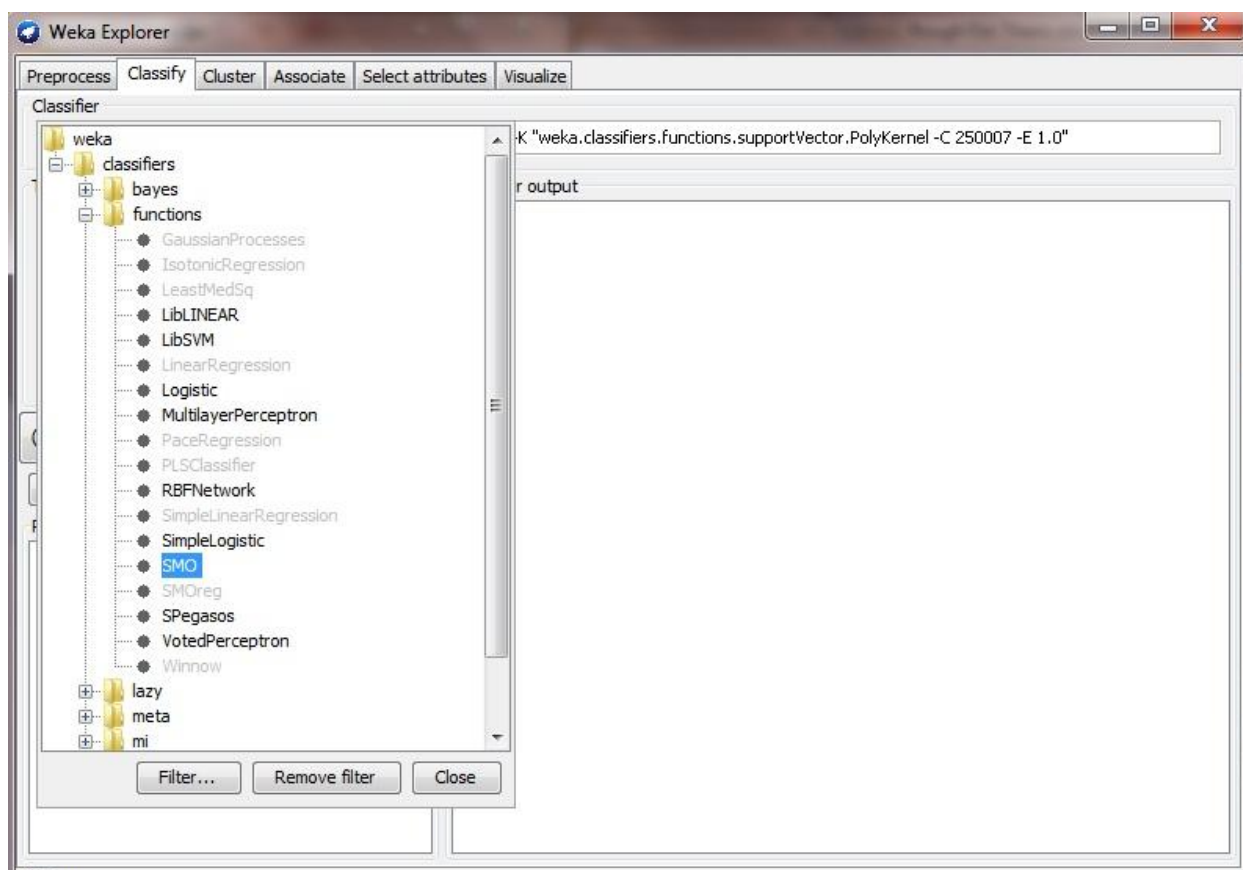


Figure 11 : Selecting SMO classifiers from WEKA tool

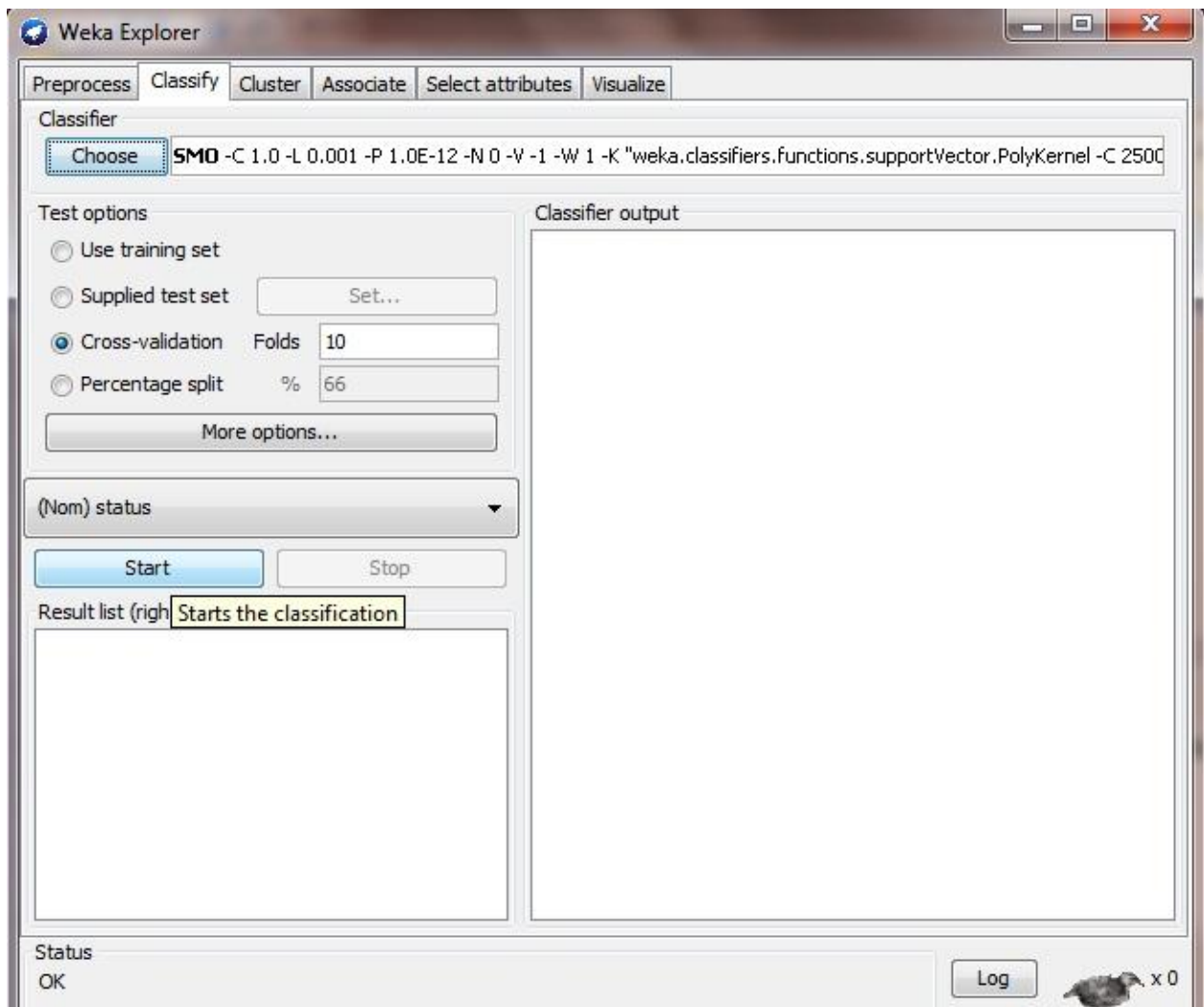


Figure 12 : Setting cross validation and starting

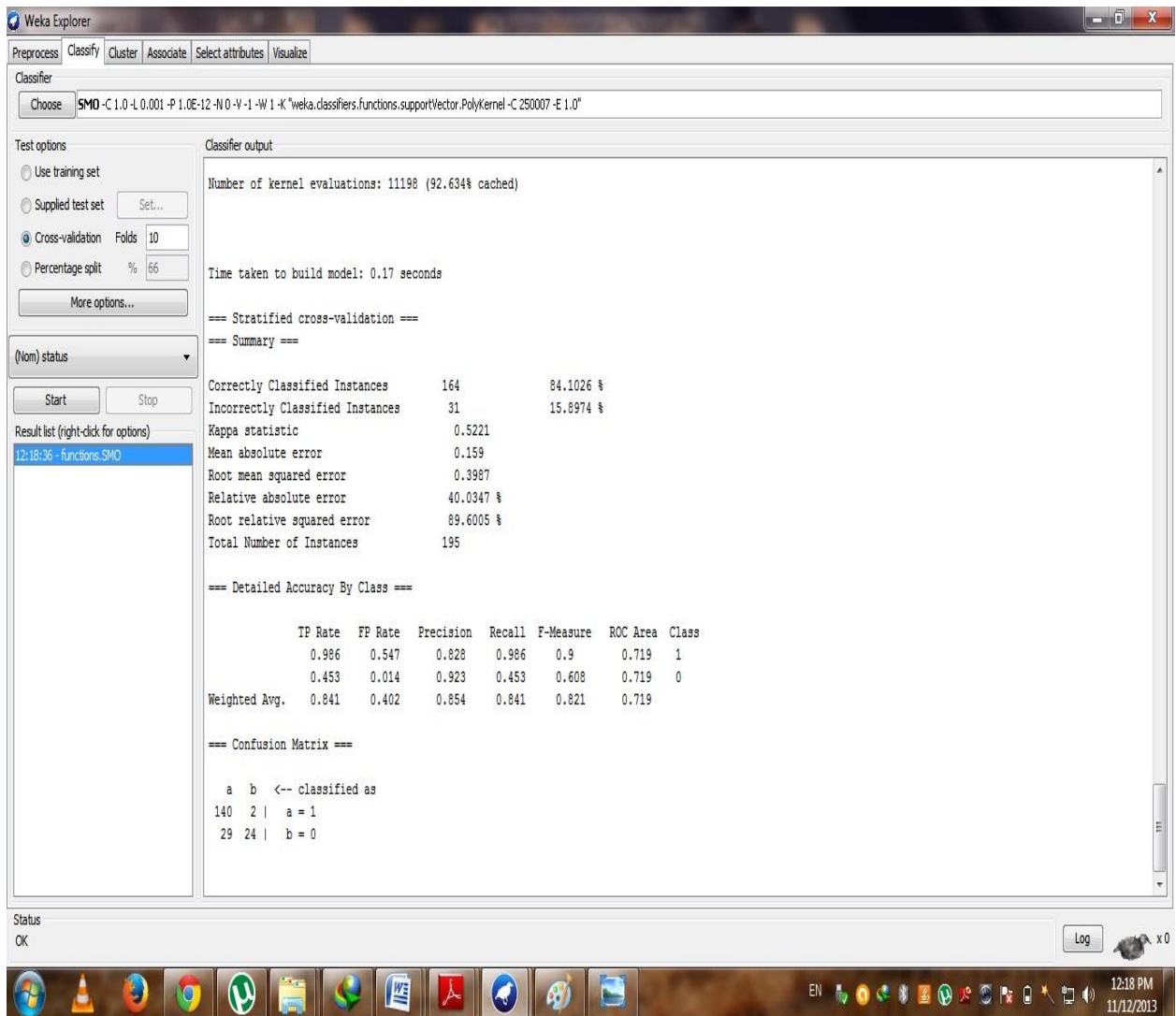


Figure 13 : SVM-SMO classifier result

4.5 Naïve Bayes, J48, LogitBoost, ADTree, BFTree and Decision Stump Tree Classification in WEKA Tool

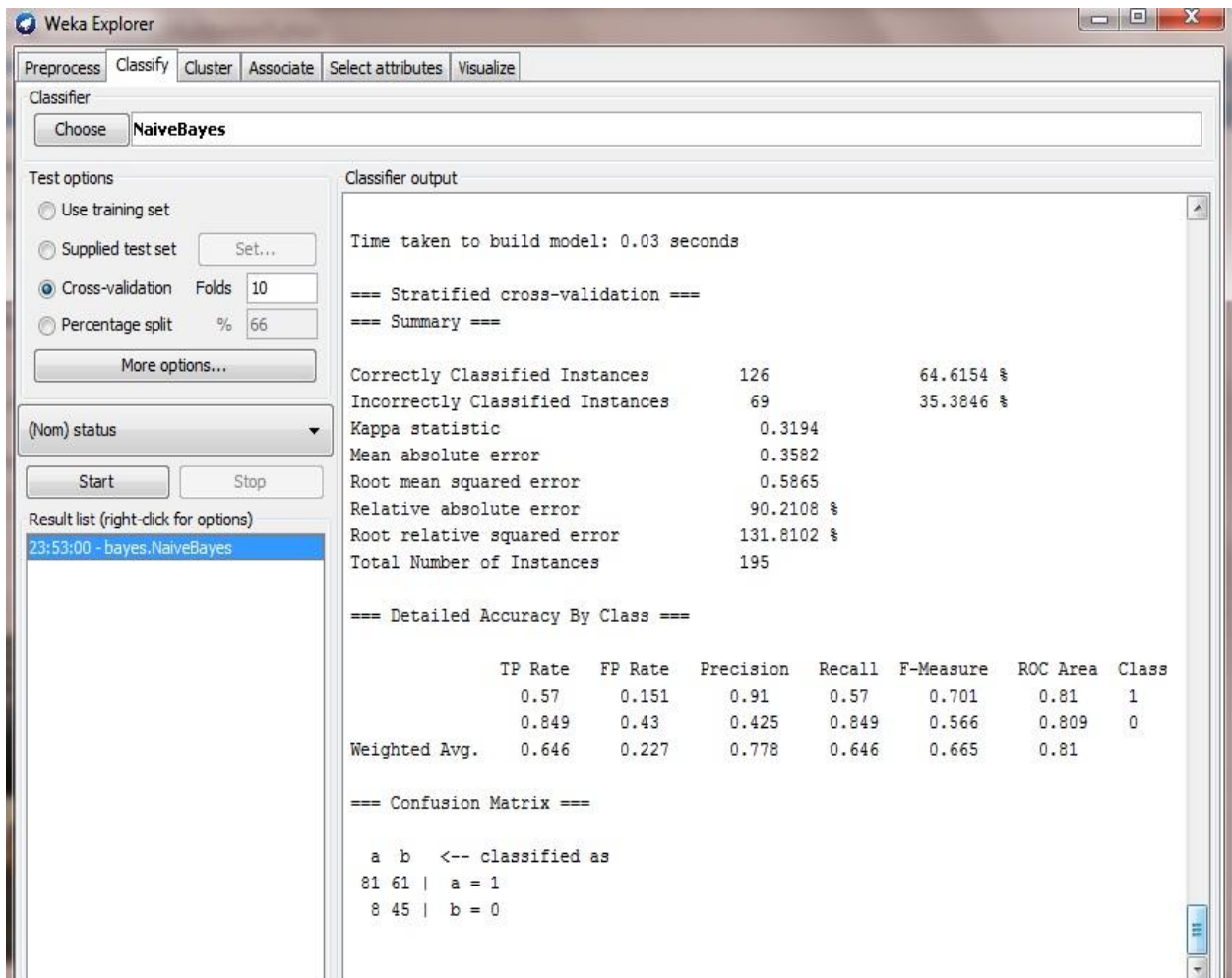


Figure 14: Naïve Bayes Classification

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier
Choose J48 -C 0.25 -M 2

Test options
 Use training set
 Supplied test set Set...
 Cross-validation Folds 10
 Percentage split % 66
 More options...

(Nom) status

Start Stop

Result list (right-click for options)
00:07:52 - trees.J48

Classifier output

=== Summary ===

Correctly Classified Instances	162	83.0769 %
Incorrectly Classified Instances	33	16.9231 %
Kappa statistic	0.554	
Mean absolute error	0.1884	
Root mean squared error	0.4059	
Relative absolute error	47.4484 %	
Root relative squared error	91.2254 %	
Total Number of Instances	195	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.908	0.377	0.866	0.908	0.887	0.748	1
	0.623	0.092	0.717	0.623	0.667	0.748	0
Weighted Avg.	0.831	0.3	0.825	0.831	0.827	0.748	

=== Confusion Matrix ===

```

a  b  <-- classified as
129 13 | a = 1
 20 33 | b = 0

```

Status

Figure 15: J48 Tree Classification

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose **LogitBoost** -P 100 -F 0 -R 1 -L -1.7976931348623157E308 -H 1.0 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump

Test options

Use training set

Supplied test set Set...

Cross-validation Folds 10

Percentage split % 66

More options...

(Nom) status

Start Stop

Result list (right-click for options)

00:10:10 - meta.LogitBoost

Classifier output

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	158	81.0256 %
Incorrectly Classified Instances	37	18.9744 %
Kappa statistic	0.5	
Mean absolute error	0.2193	
Root mean squared error	0.3571	
Relative absolute error	55.2196 %	
Root relative squared error	80.2376 %	
Total Number of Instances	195	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.894	0.415	0.852	0.894	0.873	0.855	1
	0.585	0.106	0.674	0.585	0.626	0.855	0
Weighted Avg.	0.81	0.331	0.804	0.81	0.806	0.855	

=== Confusion Matrix ===

a b <-- classified as

127	15	a = 1
22	31	b = 0

Status

Figure 16 : LogitBoost Classification

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose **ADTree -B 10 -E -3**

Test options:

- Use training set
- Supplied test set (Set...)
- Cross-validation (Folds: 10, %: 66)
- Percentage split (%: 66)

More options...

(Nom) status

Start Stop

Result list (right-click for options)

00:12:11 - trees.ADTree

Classifier output:

```

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      164      84.1026 %
Incorrectly Classified Instances    31      15.8974 %
Kappa statistic                     0.596
Mean absolute error                 0.2122
Root mean squared error             0.3386
Relative absolute error             53.4293 %
Root relative squared error         76.09 %
Total Number of Instances          195

=== Detailed Accuracy By Class ===
           TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
           0.894   0.302   0.888     0.894   0.891     0.881    1
           0.698   0.106   0.712     0.698   0.705     0.881    0
Weighted Avg.  0.841   0.249   0.84     0.841   0.841     0.881

=== Confusion Matrix ===

  a  b  <-- Classified as
127 15 |  a = 1
 16 37 |  b = 0

```

Status

Figure 17: ADTree Classification

Weka Explorer

Preprocess | **Classify** | Cluster | Associate | Select attributes | Visualize

Classifier: Choose **BFTree -S 1 -M 2 -N 5 -C 1.0 -P POSTPRUNED**

Test options:

- Use training set
- Supplied test set (Set...)
- Cross-validation (Folds: 10)
- Percentage split (%: 66)

(Nom) status

Start Stop

Result list (right-click for options)

00:15:28 - trees.BFTree

Classifier output:

```

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      142      72.8205 %
Incorrectly Classified Instances    53      27.1795 %
Kappa statistic                     0
Mean absolute error                  0.3959
Root mean squared error              0.445
Relative absolute error              99.7047 %
Root relative squared error          99.9987 %
Total Number of Instances           195

=== Detailed Accuracy By Class ===
          TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          1         1       0.728     1       0.843     0.475    1
          0         0         0         0         0         0.475    0
Weighted Avg.   0.728   0.728     0.53     0.728   0.614     0.475

=== Confusion Matrix ===
  a  b  <-- classified as
142  0  |  a = 1
 53  0  |  b = 0

```

Figure 18: BFTree Classification

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose DecisionStump

Test options:

- Use training set
- Supplied test set (Set...)
- Cross-validation (Folds: 10)
- Percentage split (%: 66)

More options...

(Nom) status

Start Stop

Result list (right-click for options): 00:17:17 - trees.DecisionStump

Classifier output:

```

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      150      76.9231 %
Incorrectly Classified Instances    45       23.0769 %
Kappa statistic                    0.3843
Mean absolute error                 0.3041
Root mean squared error             0.4161
Relative absolute error             76.5942 %
Root relative squared error         93.4974 %
Total Number of Instances          195

=== Detailed Accuracy By Class ===
           TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
           0.873   0.509   0.821     0.873   0.846     0.667    1
           0.491   0.127   0.591     0.491   0.536     0.667    0
Weighted Avg.  0.769   0.405   0.759     0.769   0.762     0.667

=== Confusion Matrix ===

  a  b  <-- classified as
124 18 | a = 1
 27 26 | b = 0

```

Status: OK

Log x 0

Figure 19 : Decision Stump Classification

4.6 Comparison Table of AdaBoostM1, SVM-SMO, Naïve Bayes, J48, LogitBoost, ADTree, BFTree and Decision Stump Tree

Table2: Comparison Table

Variables	AdaBoost.M1 Classification Result	SVM-SMO Classification Result	Naïve Bayes Result	J48 Tree Result	LogitBoost Result	ADTree Result	BFTree Result	Decision Stump Tree Result
Correctly classified instance	158 (81.0256 %)	164 (84.1026 %)	126 (64.6154 %)	162 (83.0769 %)	158 (81.0256 %)	164 (84.1026 %)	142 (72.8205 %)	150 (76.9231 %)
Incorrectly classified instance	37 (18.9744 %)	31 (15.8974 %)	69 (35.3846 %)	33 (16.9231 %)	37 (18.9744 %)	31 (15.8974 %)	53 (27.1795 %)	45 (23.0769 %)
Kappa statistic	0.512	0.5221	0.3194	0.554	0.5	0.596	0	0.3843
Mean absolute error	0.2327	0.159	0.3582	0.1884	0.2193	0.2122	0.3959	0.3041
Root mean squared error	0.3864	0.3987	0.5865	0.4059	0.3571	0.3386	0.445	0.4161
Relative absolute error	58.6083 %	40.0347 %	90.2108 %	47.4484 %	55.2196 %	53.4293 %	99.7047 %	76.5942 %
Root relative squared error	86.8352 %	89.6005 %	131.8102 %	91.2254 %	80.2376 %	76.09 %	99.9987 %	93.4974 %

Avg TP rate	0.81	0.841	0.646	0.831	0.81	0.841	0.728	0.769
Avg FP rate	0.307	0.402	0.227	0.3	0.331	0.249	0.728	0.405
Avg Precision	0.807	0.854	0.778	0.825	0.804	0.84	0.53	0.759
Avg Recall	0.81	0.841	0.646	0.831	0.81	0.841	0.728	0.769
Avg F-measure	0.808	0.821	0.665	0.827	0.806	0.841	0.614	0.762
Avg ROC area	0.811	0.811	0.81	0.748	0.855	0.881	0.475	0.667

4.7 Result Analysis

Kappa statistics: The kappa statistics is basically a measure of the agreement where it is normalized in cases of chance agreement. [2] Most commonly the kappa statistics is used in inter observer variability cases dealing with the point of 2 observers agreeing on a single interpretation. It is also used to assess performance in quality assurance schemes.

Mean absolute error: The mean absolute error is a value that calculates the closeness between the predictions or forecasts to the actual outcomes. This quantity is basically the average of the absolute errors.

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|$$

Where f_i is the prediction and y_i the true value.

Root mean squared error: The Root mean square error also sometimes known as the root mean squared deviation is a measure that is more usually used in order to calculate the difference between the values predicted by a model when compared to the actual observed values.

Relative absolute error: The Relative absolute error as the name suggests is the average of the actual values. The relative absolute error takes the total absolute error and normalizes dividing by the total absolute error.

Root relative squared error: The root relative squared error is relative to what it would have been if a simple predictor had been used. More specifically, this simple predictor is just the average of the actual values. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the simple predictor.

The root relative squared error is a measure of a simple predictor that just takes the average of the actual values. It takes the total of the squared error and normalizes it by dividing it by the total squared error by that of a predictor.

TP rate: The TP rate or the True Positive Rate is the ratio of number of PD patients predicted correctly to the total of positive cases. It is somewhat Equivalent to Recall.

FP rate: The FP rate or the False Positive Rate is the ratio of the number of healthy patients of incorrectly predicted as PD patient to the total number of healthy people.

Precision: As the name suggests Precision is the proportion of relevance of the input to the results that is obtained.

Recall: Recall is the ratio of relevant results found in the search result to the total of all relevant output. If the recall value is more it implies that relevant results are returned more quickly.

F-Measure: F-measure is a method in which we combine the recall and precision scores into a single measure of performance.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The primary motive of the thesis work is to classify patients with comparisons of different classifiers who have been affected by Parkinson's disease based on their speech. Parkinson Disease is a degenerative illness whose cardinal symptoms include rigidity, tremor and slowness of movement. The speech was chosen a testing factor as the disease has profound effect on voice and speech.

The main purpose of choosing this topic for the master thesis was that it is hard to diagnose the disease especially at its earlier stages. So building automatic techniques based on Artificial intelligence to detect the Parkinson's disease was a challenging and would be practically very useful. For the classification purpose a dataset was collected of 195 instances and 23 attributes.

This dataset was taken with 31 patients. Of these 31 patients 23 of them actually are affected by Parkinson's disease. The remaining 8 people are healthy. My system was made to test the 31 sample datasets and classify these people correctly as affected patients and healthy people. Classification is done with mainly 2 algorithms namely AdaBoost.M1 and SVM-SMO. Except these two, I also made a comparison between 6 algorithms namely Naïve Bayes, J48 tree, LogitBoost, ADTree, BFTree and Decision Stump Tree.

According to my classification result, I can conclude that SVM-SMO is the best classification approach to classify the Parkinson patients according to the dataset rather than AdaBoost.M1 classifiers since the error rate in SVM-SMO is less than that of AdaBoost.M1. In addition, in case of SVM-SMO, correctly classified instances is much more than of AdaBoost.M1. In SVM-SMO the percentage was **84.1026 %**, but in AdaBoost.M1 the percentage was **81.0256 %**.

In comparison between Naïve Bayes, J48 tree, LogitBoost, ADTree, BFTree and Decision Stump tree, according to the classification result, I can conclude that ADTree and J48 Tree is the best classification approach to minimize the error rate and increase the number of correctly classified instances. In ADTree and J48 Tree the correctly classified instances percentage was **84.1026 % and 83.0769 %** respectively.

I worked with AdaBoost.M1 classifier to classify Parkinson's patients. Since there is an upgraded version of AdaBoost Classifier which is AdaBoost.M2, So my future work would be to implement the dataset with the AdaBoost.M2 classifier. In addition, I also have a plan to make an Artificial Intelligence mobile application on Parkinson Disease Symptom Predictions.

REFERENCES

- [1] parkinson australia(<http://www.parkinsons.org.au/about-ps/about-pd.htm>)
- [2] Udaya kumar and Magesh kumar. “Classification of Parkinson’s disease using Multipass Lvq, Logistic Model Tree, K-star for Audio Data set, classification of Parkinson Disease using Audio Dataset”. Dalarna University, School of Technology and Business Studies, Computer Engineering, 2011.
- [3] <http://www.ibm.com/developerworks/library/os-weka1/>
- [4] Anchana Khemphila and Veera Boonjing. “Parkinson Disease Classification using Neural Network and Feature selection”.
- [5] P.Yasodha and M. Kannan. “Analysis of a population of a Diabetic Patients Databases in Weka Tool”.
- [6] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
URL <http://www.ics.uci.edu/~mlern/MLRepository.html>.
- [7] <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3051371/>
- [8] Michael J.Kearns and Umesh V.Vazirani. “An Introduction to Computational Learning Theory”. MIT Press, 1994.
- [9] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Machin Learn Res.2003;3:1157–1182.
- [10] <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3051371/table/T1/>
- [11] Yoav Freund, Robert E. Schapire, “ A Short Introduction to Boosting” in AT&T Labs-Research, Shannon Laboratory, 180 Park Avenue, Florham Park, NJ 07932 USA
- [12] L.G. Valiant. “A theory of the learnable” Communications of the ACM, 27(11):1134-1142, November 1984.
- [13] John C. Platt, “A Fast Algorithm for Training Support Vector Machines” . Microsoft Research, Technical Report MSR-TR-98-14, April 21, 1998
- [14] Hastie T, Tibshirani R, Friedman JH. The elements of statistical learning : data mining, inference, and prediction : with 200 full-color illustrations. Springer; New York: 2001.