International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

A Study of AdaBoost and Bagging Approaches on Student Dataset

G. T. Prasanna Kumari[1], Dr. M. Usha Rani[2]
[1]Associate Professor, Department of Computer Science and Engineering, S V Engineering College for Women, Tirupathi, AP, India-517501
[2]Professor, Dept. of Computer Science, Sri Padmavati Mahila Visva Vidyalayam, Tirupati, AP, India-517502

Abstract— *Data Mining otherwise called Information Mining, is the way towards examining data from different points of view and converting it into valuable data (information). Various essential and distinctive data mining techniques, for example, association rules, classification, clustering are performed utilizing WEKA tool. Classification is one of the Data Mining techniques that examines a given data set and incites a model for every class based on their attributes of the dataset. Particularly, Classification algorithms are very helpful in classifying the data. To order to increase the performance of individual classifier, ensemble is used. The ensemble is the combination of two or more classifiers. Different types of ensemble methods are available. Two most popular ensemble approaches are AdaBoost and Bagging, implemented on Educational Systems. These approaches are very successful in improving the accuracy, error rates and time required to build the model. So, these approaches are implemented on Student Data. The experiment is carried out using WEKA tool. Currently there is an increasing interest in application of data mining techniques in educational systems. This paper surveys on application of AdaBoost and Bagging approaches to Educational System using student dataset*.

Keywords— *Ensemble, Bagging, AdaBoost, Preprocessing, Classification, Educational data mining.*

I. Introduction

Data mining, also known as information disclosure in databases, will be the procedure of finding intriguing more suitable designs starting with huge add up of information. Classification assumes imperative part in the field of Data Mining. Classification may be a type of information examination that extracts models describing every last one of major information classes. Such models, called classifiers, foresee unmitigated class labels. Classification may be an two-stage process, comprising of a preparation venture (where a classification model may be constructed from training set), also a testing step(where those model may be used to foresee class labels for given data set). The expanding rate from claiming information extent diminishing effectiveness of single classifier. In this way blending from claiming two or more classifiers to exceptional prediction. Furthermore voting for data(outputs) may be used, execution will be improved, such systems need aid known as Ensemble Methods.

In the most recent decade, those ensemble based frameworks need reveled in an developing consideration because of their large portions wanted properties, and the wide range from claiming requisitions that can profit from them. The more popular areas where ensemble systems become naturally useful, such as incremental learning, data fusion, feature selection and error correcting output codes.

Whereas there is no single ensemble generation algorithm or combination rule that is universally better than others, all of the approaches have been shown to be effective on a wide range of real world and benchmark datasets, provided that the classifiers can be made as diverse as possible. In the absence of any other prior information, the best ones are the simplest and least complicated ones that can learn the underlying data distribution.

Ensemble methodology search a few distinct slants preceding making a choice. Those guideline may be on weigh a few individual classifiers, and consolidate them so as to arrive at a arrangement that is superior to the one gotten by each for them independently. Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data sets by taking a (weighted) vote of their predictions.

II. Adaboost and Bagging Approaches

AdaBoost and Bagging are two of the most well-known ensemble learning methods due to their theoretical performance and experimental results. However, these algorithms have been used mainly in batch mode, i.e., they require the entire training set to be available at once and, in some cases, require random access to the data. Bagging is useful for weak and unstable classifiers with a non-decreasing learning curve and critical training sample sizes. Boosting is beneficial only for weak, simple classifiers, with a non-decreasing learning curve, constructed on large training sample sizes.

AdaBoost is an algorithm for constructing a strong classifier which is linear combination of weak classifiers. Weak classifiers misclassifies certain instances. The set of weak classifiers is built iteratively from the training data over hundreds or thousands of iterations. At each iteration, the instances in the training data are reweighted according to how well they are classified. Weights are computed for the weak classifiers based on their classification accuracy. The assigned weight is used to vote for each classifier. If there is less error rate of classifier then more weight assigned to its vote. This training process is repeated. The weight of classifiers which voted for an object of a class is added. The class which gains higher total weight is the final class and it will introduced as the predictive class for that object.

AdaBoost and Bagging are two different approaches to improve the performance of the model:

1. AdaBoost is a approach to calculate the output using several different models and then average the result using a

weighted average approach. By combining the advantages and pitfalls of these approaches by varying your weighting formula we can come out with a good prediction for a large range of input data.

2. Bagging (stands for Bootstrap Aggregation) is the way decrease the variance in prediction by generating additional data for training from your original dataset using combinations with repetitions to produce multisets of the same cardinality/size as your original data. By increasing the size of your training set you can't improve the model prediction, but just decrease the variance, narrowly tuning the prediction to expected outcome.

A. *AdaBoost Algorithm*

AdaBoost combines multiple base classifiers whose combined performance is significantly better than that of any of the base classifiers.

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records.
- Initially, all N records are assigned equal weights.
- Records that are wrongly classified will have their weights increased.
- Records that are classified correctly will have their weights decreased.

B. *Bagging Algorithm*

Bagging is almost always more accurate than a single classifier, it is sometimes much less accurate than Boosting.
Bagging (for bootstrap aggregation) creates an ensemble by training individual classifiers on bootstrap samples of the training set. Bagging traditionally uses classifiers of the same type e.g., decision trees, and a simple combiner consisting of a majority vote across the ensemble.

Training phase –

- Initialize the parameters
  D = Ø, the ensemble L, the number of classifiers to train
- For k = 1,….,L
  Take a bootstrap sample S_k from Z
  Build a classifier D_k using S_k as the training set
  Add the classifier to the current ensemble, D = D U D_k
- Return D

Classification phase -

- Run D_1,…….,D_L on the input x.
- Class with major number of votes is chosen as a label for x

III. Educational Data Mining

There need aid expanding exploration in utilizing data mining strategies in educational framework. This developing field will be known as Educational Data Mining. It can be applied on the data related to the field of education. One of the educational system problems that are solved with data mining technique such as classification is the prediction of students' academic performances. Prediction of students' performance is for identifying the low academic performance students [1]. Students' academic performance is based upon various factors like personal, psychological, social, internet accessing time of the student, number of journals accessed by the student etc. A very promising tool to achieve this objective is the use of Data Mining technique such as classification in Weka [2].

Educational Data Mining (EDM) is a developing discipline, for creating strategies to exploring the exceptional sorts about student information that come from education systems, and utilizing routines to better concentrate on the students of all categories. Educational Data Mining concentrates on new tools and algorithms to discover data patterns. EDM develops strategies from statistics, machine learning, and data mining to examine student information gathered teaching and learning. EDM tests hypotheses furthermore informs training framework. Educational data mining is emerging as a research area for understanding how students learn. New computer-supported interactive learning methods and tools have led up opportunities to collect and analyze student data, to discover patterns(performance) in instances of student data, to discover the student's interest towards the subject (can be analyzed by going through the websites browsed by the student) and to make new discoveries about students learning skills.

The implementation of data mining methods and tools for analyzing data available at educational institutions, defined as Educational Data Mining (EDM) is a relatively new stream in the data mining research. In traditional educational systems, educators are able to obtain feedback on student learning experiences in face-to-face interactions with students, enabling a continual student's feedback of their teaching programs. Decision making of classroom teaching and learning processes involves observing a student's behavior, analyzing historical and personal student data, and estimating the effectiveness of pedagogical strategies.

The problems that are most often attracting the attention of researchers and becoming the reasons for applying data mining at education systems are focused mainly on retention of students and improving institutional effectiveness.

The challenge in the presented data mining project is to predict the student university performance based on the collection of attributes providing information about the students. The selected target variable in this case, or the concept to be learned by data mining algorithm, is the "Pedagogic Technique Class". A categorical target variable is constructed based on the original numeric parameter university percentage, performance at schooling, personal details and Students' academic performance.

The final dataset used for the project implementation contains 100 instances. The attributes related to the student dataset can also include personal data like gender, age, students place, profile of the school, the final score at school, the successful admission exam, and the score and rank achieved at that exam.

Using AdaBoost and Bagging Algorithms, students has been classified with Pedagogic Techniques. Based on the performance corresponding Pedagogic Techniques will be implemented.

Adaboost was designed to use short decision tree models, each with a single decision point. Such short trees are often referred to as decision stumps.

TABLE 1. Student data.

S.NO.	Student Name	%AGE	PERFORMANCE	PEDAGOGIC TECHNIQUES
1	M BHARATHI	65.6	FIRST CLASS	BRAINSTROMING
2	M DHARMA TEJA	57.2	SECOND CLASS	ROLE PLAY
3	M KALPANA	72.53	DISTINCTION	SEMINARS OR CASE STUDY
4	M NIKHITHA	70.4	DISTINCTION	SEMINARS OR CASE STUDY
5	M RAKESH	59.2	SECOND CLASS	ROLE PLAY
6	MADAPURI VENKATADEEPIKA	56.27	SECOND CLASS	ROLE PLAY
7	MADDE SUSMITHA	73.2	DISTINCTION	SEMINARS OR CASE STUDY
8	MADGULA AMIT	46	PASS	DEMONSTRATION AND GROUP DISCUSSION
9	MADINENI HARI VAMSI KUMAR	30.93	FAIL	FALSE
10	MANGAPATNAM LAVANYA	57.6	SECOND CLASS	ROLE PLAY
11	MAVILLAPALLI PURUSHOTHAM	68.13	FIRST CLASS	BRAINSTROMING
12	MODEPALLI VISHNU VAMSI	73.2	DISTINCTION	SEMINARS OR CASE STUDY
13	MUCHHELI ANILKUMAR REDDY	61.33	FIRST CLASS	BRAINSTROMING
14	NANDALA NARAYANAMMA	70.53	DISTINCTION	SEMINARS OR CASE STUDY
15	O PAVAN KUMAR	62.27	FIRST CLASS	BRAINSTROMING
16	P NITHIN KUMAR REDDY	65.6	FIRST CLASS	BRAINSTROMING

IV. Implementing Ensemble Machine Learning Algorithms in Weka

Ensemble algorithms are a powerful class of machine learning algorithm that combine the predictions from multiple models.

Waikato Environment for Knowledge Analysis (Weka) is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. It is free software licensed under the GNU General Public License. Weka (pronounced to rhyme with Mecca) contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions. Java-based version (Weka 3), for which development started in 1997, is now used in many different application areas, in particular for educational purposes and research. WEKA is the collection or a suite of the tools for performing data mining with the implementation of the 'association rules' in it. Basically it is a collection of machine learning algorithm for the task of data mining, which is able to be applied directly to dataset or can call from your own java code.

A benefit of using Weka for applied machine learning is that makes available so many different ensemble machine learning algorithms. Each algorithm that we cover will be briefly described in terms of how it works, key algorithm parameters will be highlighted and the algorithm will be demonstrated in the Weka Explorer interface.

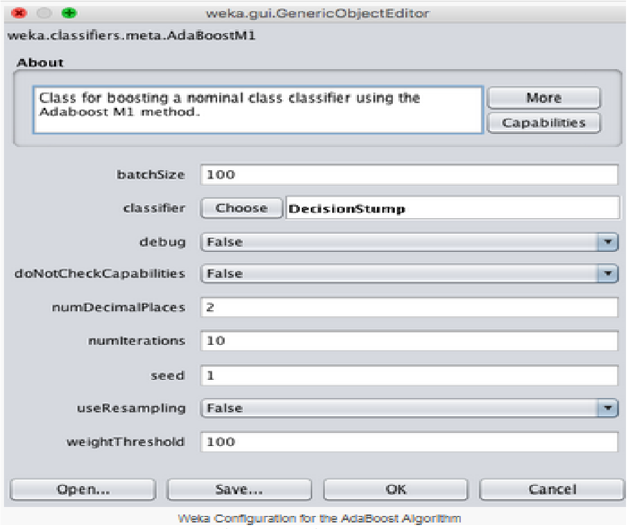
AdaBoost is an ensemble machine learning algorithm for classification problems. It is part of a group of ensemble methods called boosting, that add new machine learning models in a series where subsequent models attempt to fix the prediction errors made by prior models. AdaBoost was the first successful implementation of this type of model.

Adaboost was designed to use short decision tree models, each with a single decision point. Such short trees are often referred to as decision stumps.

The first model is constructed as per normal. Each instance in the training dataset is weighted and the weights are updated based on the overall accuracy of the model and whether an instance was classified correctly or not. Subsequent models are trained and added until a minimum accuracy is achieved or no further improvements are possible. Each model is weighted based on its skill and these weights are used when combining the predictions from all of the models on new data.

Choose the AdaBoost algorithm:

1. Click the "Choose" button and select "AdaBoostM1" under the "meta" group.
2. Click on the name of the algorithm to review the algorithm configuration.


weka.gui.GenericObjectEditor
weka.classifiers.meta.AdaBoostM1
About
More
Capabilities
Open...
Save...
OK
Cancel
Weka Configuration for the AdaBoost Algorithm

Class for boosting a nominal class classifier using the Adaboost M1 method.

batchSize 100
classifier Choose DecisionStump
debug False
doNotCheckCapabilities False
numDecimalPlaces 2
numIterations 10
seed 1
useResampling False
weightThreshold 100

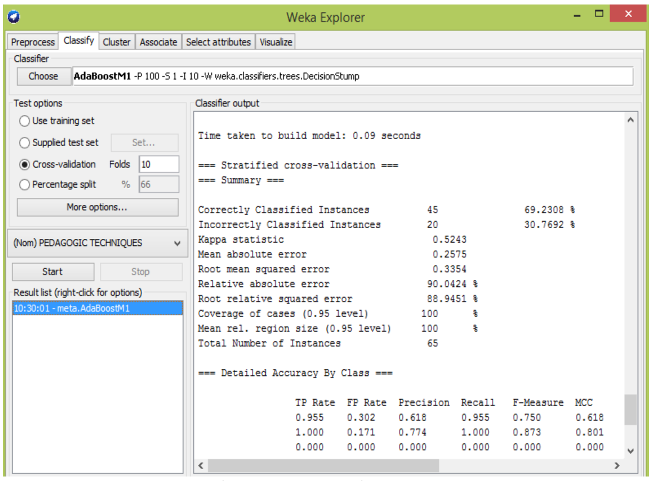
Fig. 1. Weka Configuration for the AdaBoost Algorithm.

The weak learner within the AdaBoost model can be specified by the classifier parameter.

The default is the decision stump algorithm, but other algorithms can be used, a key parameter in addition to the weak learner is the number of models to create and add in series. This can be specified in the numIterations parameter and defaults to 10.

1. Click "OK" to close the algorithm configuration.
2. Click the "Start" button to run the algorithm on the student dataset.

We can see that with the default configuration that AdaBoost achieves an accuracy of 69% and time taken to build model is 0.09 seconds.



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

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

(Nom) PEDAGOGIC TECHNIQUES

10:30:01 - meta.AdaBoostM1

```
Time taken to build model: 0.09 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          45               69.2308 %
Incorrectly Classified Instances        20               30.7692 %
Kappa statistic                          0.5243
Mean absolute error                      0.2575
Root mean squared error                  0.3354
Relative absolute error                 90.0424 %
Root relative squared error             88.9451 %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)     100      %
Total Number of Instances               65

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.955    0.302    0.618      0.955    0.750      0.618
                 1.000    0.171    0.774      1.000    0.873      0.801
                 0.000    0.000    0.000      0.000    0.000      0.000
```

Fig. 2. Weka Classification Results for the AdaBoost Algorithm.

Bootstrap Aggregation or Bagging for short is an ensemble algorithm that can be used for classification or regression.

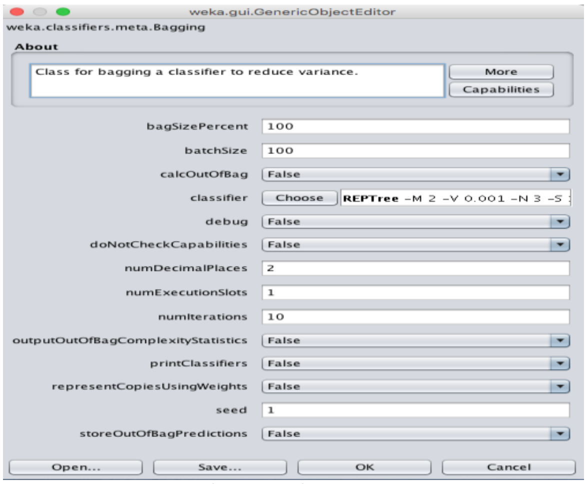
Bootstrap is a statistical estimation technique where a statistical quantity like a mean is estimated from multiple random samples of your data (with replacement). It is a useful technique when you have a limited amount of data and you are interested in a more robust estimate of a statistical quantity.

This sample principle can be used with machine learning models. Multiple random samples of your training data are drawn with replacement and used to train multiple different machine learning models. Each model is then used to make a prediction and the results are averaged to give a more robust prediction.

It is a technique that is best used with models that have a low bias and a high variance, meaning that the predictions they make are highly dependent on the specific data from which they were trained. The most used algorithm for bagging that fits this requirement of high variance are decision trees.

Choose the bagging algorithm:

1. Click the "Choose" button and select "Bagging" under the "meta" group.
2. Click on the name of the algorithm to review the algorithm configuration.



weka.gui.GenericObjectEditor

weka.classifiers.meta.Bagging

About

Class for bagging a classifier to reduce variance.

bagSizePercent	100
batchSize	100
calcOutOfBag	False
classifier	Choose REPTree -M 2 -V 0.001 -N 3 -S
debug	False
doNotCheckCapabilities	False
numDecimalPlaces	2
numExecutionSlots	1
numIterations	10
outputOutOfBagComplexityStatistics	False
printClassifiers	False
representCopiesUsingWeights	False
seed	1
storeOutOfBagPredictions	False

Fig. 3. Weka Configuration for the Bagging Algorithm.

A key configuration parameter in bagging is the type of model being bagged. The default is the REPTree which is the Weka implementation of a standard decision tree, also called a Classification and Regression Tree or CART for short. This is specified in the classifier parameter.

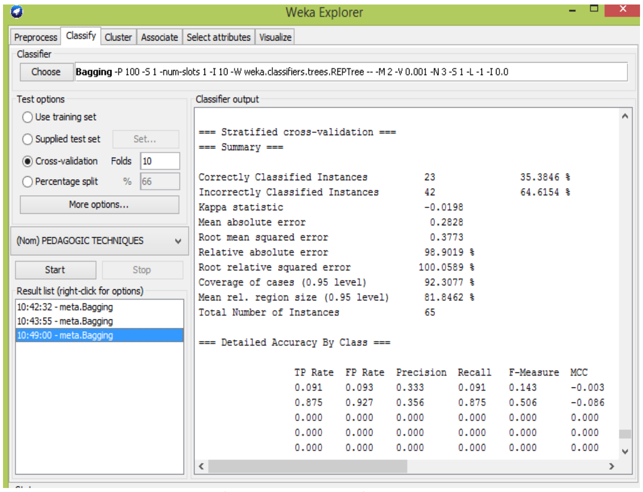
The size of each random sample is specified in the bagSizePercent, which is a size as a percentage of the raw training dataset. The default is 100% which will create a new random sample the same size as the training dataset, but will have a different composition.

This is because the random sample is drawn with replacement, which means that each time an instance is randomly drawn from the training dataset and added to the sample, it is also added back into the training dataset (replaced) meaning that it can be chosen again and added twice or more times to the sample.

Finally, the number of bags (and number of classifiers) can be specified in the numIterations parameter. The default is 10, although it is common to use values in the hundreds or thousands. Continue to increase the value of numIterations until you no longer see an improvement in the model, or you run out of memory.

1. Click "OK" to close the algorithm configuration.
2. Click the "Start" button to run the algorithm on the Student dataset.

We can see that with the default configuration that bagging achieves an accuracy of 35% and time taken to build model is 0.06%.



Weka Explorer

Classifier: Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.REPTree -- -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Test options

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

(Nom) PEDAGOGIC TECHNIQUES

Result list (right-click for options)

10:42:32 - meta.Bagging
10:43:55 - meta.Bagging
10:49:00 - meta.Bagging

Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          23               35.3846 %
Incorrectly Classified Instances        42               64.6154 %
Kappa statistic                         -0.0198
Mean absolute error                      0.2828
Root mean squared error                  0.3773
Relative absolute error                 98.9019 %
Root relative squared error            100.0589 %
Coverage of cases (0.95 level)          92.3077 %
Mean rel. region size (0.95 level)      81.8462 %
Total Number of Instances               65

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.091    0.093    0.333      0.091    0.143      -0.003
                 0.875    0.927    0.356      0.875    0.506      -0.086
                 0.000    0.000    0.000      0.000    0.000      0.000
                 0.000    0.000    0.000      0.000    0.000      0.000
                 0.000    0.000    0.000      0.000    0.000      0.000
```

Fig. 4. Weka Classification Results for the Bagging Algorithm.

V. Comparison of the AdaBoost and Bagging Approaches to the Student Data

The AdaBoost classifiers has improved the classification accuracy. The real Adaboost algorithm gives minor error rates than Bagging algorithm.

TABLE 2. Accuracy and Error.

Sl.No	Ensemble Method	Accuracy	Error	Time to build model
1	AdaBoost	69%	0.25	0.09s
2	Bagging	35%	0.28	0.06s

Table 2 shows Accuracies, Error rates and time taken to build the model of AdaBoost and Bagging methods. Figure 5 shows analysis of accuracies and error rates for AdaBoost and Bagging methods.

The AdaBoost classifiers can improve the classification accuracy. The real Adaboost algorithm gives minor error rates than Bagging algorithm.

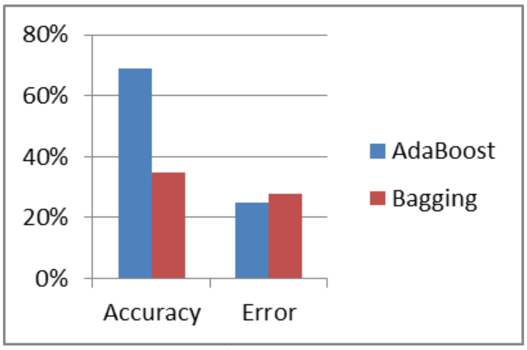


Fig. 5. Analysis of accuracies and error rates.

To improve performance with ensembles, we can combine the predictions from multiple models. In fact, we can often get good performance from combining the predictions from multiple good enough models rather than from multiple highly and fragile models. Obviously, ensembles takes more memory and processing time.

The study main objective is to find out if it is possible to predict the class pedagogic techniques using student attributes which are retained in the model. The WEKA Explorer application is used at this stage. 10 fold cross validation is used for the evaluation. The J48 is a powerful decision tree method that performs well on the student dataset. In this experiment we are going to investigate whether we can improve upon the result of the J48 algorithm using ensemble approaches. We are going to try two popular ensemble methods: AdaBoost and Bagging. For constructing the ensemble we are considering base classifiers such as AdaBoost and Bagging in combination with classifier such as J48. Accuracy is very important in the field of student domain, the performance measure accuracy of classification is considered in this study. Each classifier is applied for two testing options − cross validation (using 10 folds and applying the algorithm 10 times – each time 9 of the folds are used for training and 1 fold is used for testing) and percentage split (2/3 of the dataset used for training and 1/3 – for testing).For student data various data mining techniques are available. In the proposed, Adaboost and Bagging ensembles are constructed in WEKA using 10 fold cross validation. The results for AdaBoost show that decision tree shows good results. In all if considering average accuracy of student dataset shows better accuracy for bagging, whereas AdaBoost shows good accuracy.

The most popular and successful of all ensemble generation algorithms, AdaBoost (Adaptive Boosting) is an extension of the original boosting algorithm, that extends boosting to the multi-class problems. AdaBoost generates an ensemble of classifiers, the training data of each is drawn from a distribution that starts uniform and iteratively changes into one that provides more weight to those instances that are misclassified. Each classifier in AdaBoost focuses increasingly on the more difficult to classify instances. The classifiers are then combined through weighted majority voting.

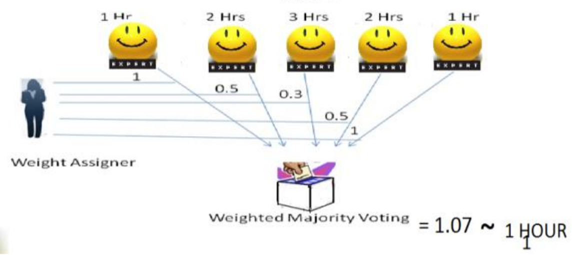


Fig. 6. Performance of Weighted Majority Voting.

Here is one model in figure 6 using adaboost algorithm giving the number of study hours to be assigned to our children (5-10 yrs) spend on homework. Weighted majority voting output is 1 Hour. Number of classifiers are 5, in order to improve the performance of output, we will increase the number of classifiers. Then, more accuracy for weighted majority voting.

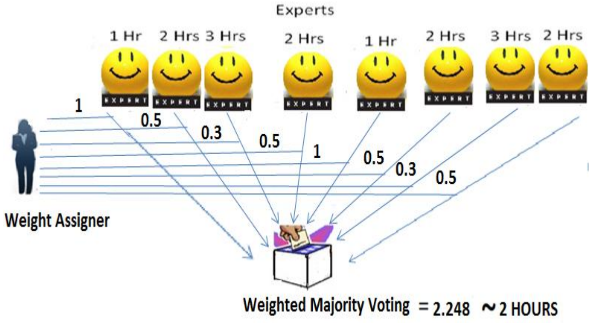


Fig. 7. Improve in performance of Weighted Majority Voting.

By increasing the number of classifiers to 8 in Figure 7, performance of Weighted Majority Voting have been improved i.e., 2 Hours in the above figure. Since the output depends on the majority, its better to increase the number of classifiers. Maximum number of classifiers giving the same output, weighted majority voting will be giving that output as the final output.

The boosting method is an iterative process in which the weights of correctly classified instances are decreased and the weights of misclassified instances are increased. This produces classifiers that focus on classifying instances that were previously misclassified. The final prediction is determined by a weighted vote in which the predictions from well performing classifiers have greater influence in the voting process.

VI. Conclusion

In this paper we made an analysis of the accuracy, error rate and the processing time of student datasets with AdaBoost and Bagging algorithms. The experimental results show that, with the accuracy an error rate point of view, AdaBoost works better than Bagging algorithm. But the time taken to build the model for Bagging algorithm is less.

The performance of ensemble depends on the majority, its better to increase the number of classifiers. Maximum number of classifiers giving the same output, weighted majority voting will be giving efficient output.

References

[1] A. Goyal and R. Kaur, “A survey on ensemble model for loan prediction,” *International Journal of Engineering Trends and Applications (IJETA)*, vol. 3, issue 1, pp. 32-37, 2016.

[2] S. B. Meshram and S. M. Shinde, “A survey on ensemble methods for high dimensional data classification in biomedicine field,” *International Journal of Computer Applications*, vol. 111, no 11, pp. 5-7, 2015.

[3] Dr. A. AL-Malaise, Dr. A. Malibari, and M. Alkhozae, “Students’ performance prediction system using multiagent data mining technique,” *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 4, no. 5, pp. 1-20, 2014.

[4] M. Sukanya, S. Biruntha, S. Karthik and T. Kalaikumaran, “Data mining: Performance improvement in education sector using classification and clustering,” in *International Conference on Computing and Control Engineering (ICCCE)*, 2012.

[5] B. Bhardwaj and S. Pal, "Data mining: A prediction for performance improvement using classification," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 9, issue 4, 2011.

[6] P. Dhakate, S. Patil, K. Rajeswari, Dr. V. Vaithiyananthan, and D. Abin, "Preprocessing and classification in WEKA using different classifiers," *International Journal of Engineering Research and Applications*, vol. 4, Issue 8 (Version 1), pp. 91-93, 2014.

[7] I. Paris, L. Affendey, and N. Mustapha, "Improving academic performance prediction using voting technique in data mining," *World Academy of Science, Engineering and Technology*, vol. 4, issue 2, pp. 820-823, 2010.

[8] C. Wang, "New ensemble machine learning method for classification and prediction on gene expression data," *Conf Proc IEEE Eng Med Biol Soc*, pp. 3478-3481, 2006.

[9] Y. Liu, "Drug design by machine learning: Ensemble learning for QSAR modeling," in the *Fourth International Conference on Machine Learning and Applications (IC MLA'05)*, 2005.

[10] R. Polikar, "Ensemble learning," Scholarpedia, 2008.

[11] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21-45, 2006.

[12] L. Rokach, *Pattern Classification Using Ensemble Methods*, 2010.

[13] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," *Appearing in Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[14] H. Syed-Mohammed, J. Leander, M. Marbach, and R. Polikar, "Can AdaBoost.M1 learn incrementally? A comparison to Learn++ under different combination rules," *Int. Conf. on Artificial Neural Networks (ICANN2006), Lecture Notes in Computer Science (LNCS)*, vol. 4131, pp. 254-263, Athens, Greece. Berlin: Springer, 2006.

[15] D. Parikh and R. Polikar, "An ensemble based incremental learning approach to data fusion, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, no. 2, pp. 437-500, 2007.

Mrs G. T. Prasanna Kumari is an Associate Professor in the Department of Computer Science and Engineering, S.V.Engineering College for Women, Tirupati. She is pursuing Ph.D., in Computer Science in the area of Ensemble based Classifiers. She is in teaching since 1999. She presented papers on Data Mining and Networks at National and International Conferences and published articles in International journals.

Dr. M. Usha Rani is Professor in the Department of Computer Science, Sri Padmavati Mahila Viswavidyalayam (SPMVV Womens' University), Tirupati. She did her Ph.D. in Computer Science in the area of Artificial Intelligence and Expert Systems. She is in teaching since 1992. She presented many papers at National and Internal Conferences and published articles in national & international journals. She also written 4 books like Data Mining - Applications: Opportunities and Challenges, Superficial Overview of Data Mining Tools, Data Warehousing & Data Mining and Intelligent Systems & Communications. She is guiding M.Phil. and Ph.D. in the areas like Artificial Intelligence, Data Warehousing and Data Mining, Computer Networks and Network Security etc.