ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.1

Comparative Study of Meta Classification Algorithm: Bagging, AdaboostM1 and Stacking with Concept Drift based Synthetic Dataset Hyperplane1 and Hyperplane2

Sandeep Kumar Budhani[1], C. K. Jha[2], Amir Ahmad[3]
Assistant Professor[1], Professor & Head[2], Professor[3]
Department of Computer Science & Engineering[1], Department of Computer Science[2], Department of Information Technology[3]
Graphic Era Hill University, Bhimtal Campus, Nainital, India[1]
Banasthali Vidyapith, Banasthali, Rajasthan, India[2]
College of Information Technology, UAE University, Al- Ain, UAE[3]

Abstract:
Data mining has been a challenging area for researchers since long. And Classification is another barrier while working with data. Classification means to classify the data as per the defined attributes. The level of complexity increases if data belong to concept drift category. Concept drift data is a data which can change their nature frequently. So, working with such data is very challenging. In this paper, we have studied three Meta learning algorithms of classification as: Bagging, AdaboostM1, and Stacking. Dataset which we have used are Hyperplane1and Hyperplane2 which are synthetic datasets having 10 attributes with 100000 instances and 11 attributes with 100000 instances respectively. We have used the WEKA tool to test the performance of algorithms. The accuracy of algorithm has been taking as a performance parameter.

Keywords: Data mining, Concept drift, WEKA

1. INTRODUCTION

Data Mining is a process of Semi-Automatically examining large database to find the patterns and data [2], [12]. The major components of the architecture for a typical data mining system are shown in Fig1:

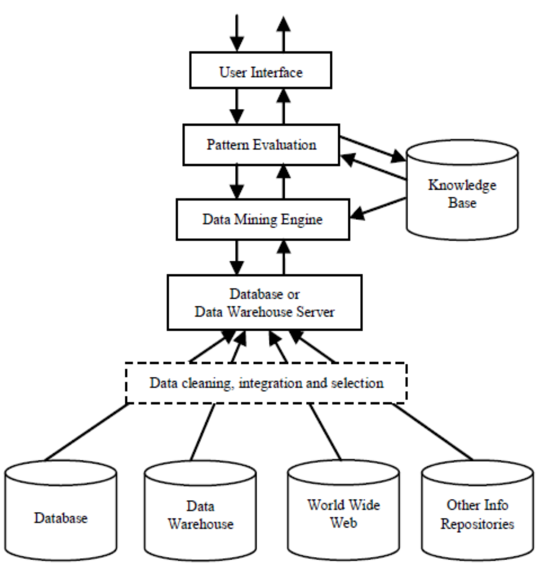


Figure.1.

Data mining also known as knowledge discovery in database (KDD), mainly data mining follows these steps; Data cleaning, Data integration, data transformation, data reduction, classification, clustering and association.

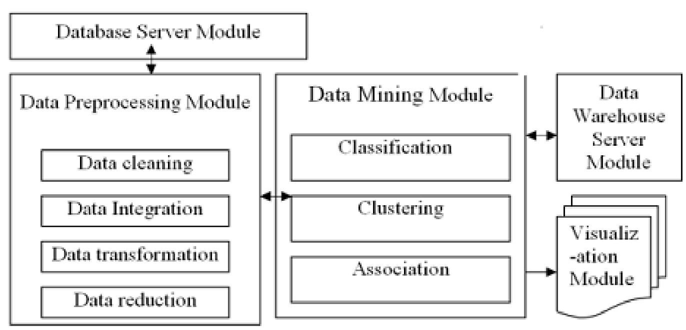


Figure. 2 [6]

Type of data is one essential factor while mining the data. The main objective of our work is to find the algorithm which produces the better accuracy. It becomes crucial when we want to predict the results with concept drift data. Prediction with static data can be done very easily because the values and nature of data are fixed. Concept drift is a data which is not fixed it can be changed very frequently. So, prediction with such data is more complex. For the comparison of different algorithms we have used the WEKA. WEKA (Waikato Environment for Knowledge Analysis) is a data mining tool developed by Waikato University in JAVA environment. It consists of various data mining algorithms.

What is Classification problem?

Classification consists of predicting a certain outcome based on a given input. In order to predict the outcome, the algorithm processes a training set containing a set of attributes and the respective outcome, usually called goalor prediction attribute. The algorithm tries to discover relationships between the attributes that would make it possible to predict the outcome. Data analysis can be done in two ways to predict future data trends as Classification and prediction. The main objective of classification is to predict the target class (Yes/ No). There are two types of classification: Binary classification and Multiclass classification. If the trained model is for predicting any of two target classes, it is known as binary classification. E.g. prediction of student results whether the student will pass or fail. Similarly the behavior of customer can be predicted whether he will buy the product or not. These kinds of problems are comes under binary classification. Multiclass classification involves assigning an object to one of several classes. The problems can be defined by their attributes i.e. properties. So, on the basis of these attribute the problems can be classified and for such purposes the various type of classifiers can be used. There are various approaches to determine the performance of classifiers. The performance can be measured by various parameters. Since the researchers are making efforts to extend the capabilities of these approaches resulting meta-learning schemes or Meta classifiers have been build up. Instead of using a single classifier to make predictions an ensemble or Meta classifier can be used. The Meta classifiers are known to reduce the error during the classification process. This is the only reason that Meta classifiers classify the data better than other classifiers. The error can be classified in two ways:

- Since the partial amount of data is used to build and test a model, so, there may be an error because it depends on the properties of sample which is representing all data and has been taken for the development of a model.
- Since every model is taking different set of attributes for model development, so, there may be a probability that the classifier is classifying any data in better way and on the other hand other classifier is not classifying the same data not very accurately. There are various classifiers available in WEKA as: Bays, Function Lazy, Meta, Rule Tree etc. In our study, we have taken three Meta classifiers: Bagging, AdaboostM1, and Stacking. These are the three most prominent methods for constructing ensemble classifiers [BREIMAN]. Since these classifiers can increase the performance over a single classifier, which means that they are a very strong candidate for a general classification model. All the Meta classifiers vote on classifications using a weighted vote. Each model in the ensemble predicts a class and can also assign a confidence value to the prediction. These values are summed and the class with the largest value (most confidence) is chosen. The bagging and boosting Meta classifiers use one simple classification method, but create more than one module while the stacking one uses different classification methods [1]. Bagging generates more than one bootstrap training sets from the true training set using sampling with alternative and makes use of every of them to generate a classifier for inclusion in the ensemble. If the set of tuple is defined by D than bagging works over D as follows [5]:

1. The training set D_i of D tuple is sampled from the real set of tuple D for iteration I (i = 1, 2, 3, 4…..K).
2. Since the meaning of bagging is bootstrap aggregation so, here each training set is in bootstrap pattern.
3. The sampling and alternative method decides that which tuples of D will be included in D_i.
4. A classifier called as M_i is learned for every training set D_i.
5. After learning, every classifier returns their prediction value for every unidentified tuple which is named as a 'VOTE'.

By taking the average of every prediction, this method may be carried out to the prediction of continuous values. Research has proved that the classifiers based on bagging give better accuracy than a single classifier. Since the bagging method decreases the inconsistencies of single classifier that is why it gives better accuracy [4]. AdaboostM1 is an important boosting algorithm. This is used to increase the accuracy of learning algorithms e.g. if the class label of tuple x_i is represented by y_i of data of class d. All the data of class d is represented by D. This algorithm assigns the weight of 1/d to every training tuple. Since an ensemble is a collection of several classifiers, so, to create an ensemble of K classifiers, we need the K rounds of this algorithm. In every round a set D_i is formed by sampling of tuples from the set of data of class d i.e. D. Accordingly the tuples will be classified and then a weight will be assigned them. For incorrect classification the weight will be increased and for the correct classification its weight will be decreased. On the basis of analysis of weight value, one can understand that how many times a tuple has been misclassified. The weight value of such misclassified tuple will be used afterward. We need to concentrate on the misclassified tuple to develop a classifier. So, using this we develop a classifiers that balance each other. Than a weight normalization will be done after updating the weight of all tuples those have been classified correctly. As soon as the boosting will be completed, the class label of a tuple will be predicted by the ensemble of classifier. Every classifier is assigned a vote. Winner is declared on the basis of the highest value of sum of weight of every classifier.

Algorithmically it can be defined as:

Assign equal weight to all instances
For each of t iterations do
Apply a learning technique to build a model from the weighted instances and store the resulting model
Down-weight each instance correctly classified by the model
End

Stacking is an interesting method in which different classification types can be used in order to form a classifier by voting a mechanism. Stacking normally works in such a way that different classifiers create their module by using the training dataset and then their performance is tested on the test set like before, but the information attain then is used in order to provide weighting for each model in the final classification step. This final step can also use the confidence of the classifications of the models in order to provide a good final classification [3].

II. EXPERIMENTATION WORK:

Here we have analyzed the performance of bagging, adaboostM1 and stacking meta-algorithm Hyperplane1 and Hyperplane2 dataset. Various parameters have been taken as:

(i) TP Rate: True Positive rate is proportion of examples which were classified as a class X among all examples which truly have the X class. This shows how much the part of the class is truly captured

$$\mathrm{TP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

TN Rate

(ii) FP Rate: False Positive Rate is the proportion of the examples which were incorrectly identified and belong to another class.

$$FP = \frac{FP}{FP + TN}$$

(iii) Recall: Recall is the proportion of examples which were classified as class X among all examples which truly have class X. It is a measure of completeness. Negative predictive value is called as recall.

$$Recall = \frac{TP}{TP + FN}$$

(iv) F-measure: F measure is aggregate of precision and recall, The formula is defined as:

$$F - measure = \frac{2 * P * R}{P + R}$$

Where P = Precision and R = Recall

(v) ROC Area: ROC is comparison of two operating characteristics TPR and FPR. It is also known as a receivers operating characteristic curve. A receiver operating characteristic curve is a graphical measure which interprets the performance of a classified as its discrimination threshold is varied. It is an outcome of plotting the true positive rate vs. false positive rate at various threshold settings. True positive rate is fraction of true positives out of the total actual positives while the fraction of false positives out of the total actual negatives indicates false positive rate. The point in threshold curve record various statistics such as true positive, false positive etc. The curves are generated by sorting the prediction produced by the classifier in descending order of probability it assigns to the positive class. The formula is defined as:

ROC Area: TP Rate= TP/TP+FN*100, FP Rate= FP/FP+TN*100.

(vi) Time to Build: It defines the total time taken to build a model

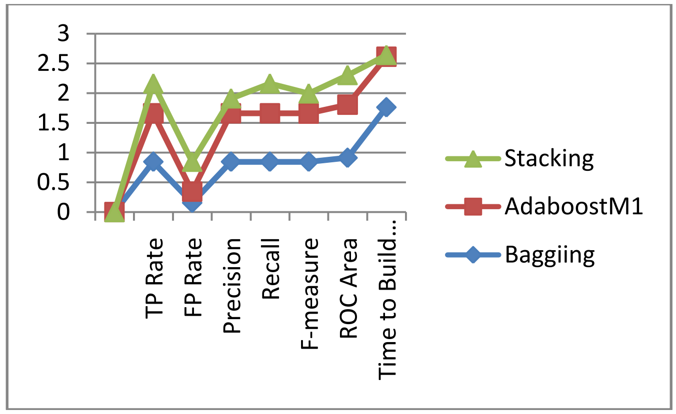
(vii) Accuracy: Accuracy defines percentage of correctly classified instances from the test set by the classifier.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

(viii)Precision or positive predictive value (PPV): Precision is the proportion of examples which truly have class X among all those which were classified as X. It is a measure of exactness. Positive predictive value is called as precision.

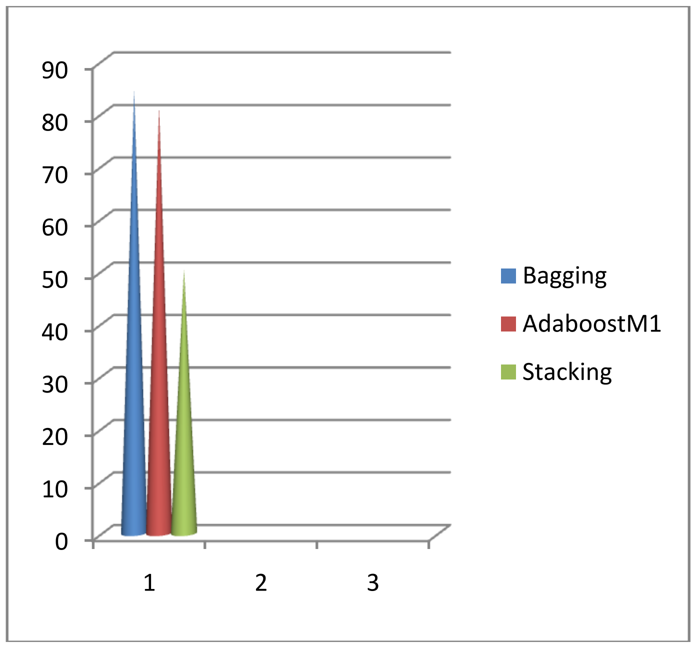
$$PPV = \frac{TP}{TP + FP}$$

Classification Algorithm Type: Meta Dataset: Hyperplane1			
Algorithm / Attribute	Bagging	AdaboostM1	Stacking
TP Rate	0.845	0.813	0.503
FP Rate	0.155	0.188	0.503
Precision	0.845	0.813	0.253
Recall	0.845	0.813	0.503
F-measure	0.845	0.812	0.337
ROC Area	0.911	0.892	0.5
Time to Build (in Sec)	1.76	0.85	0.03



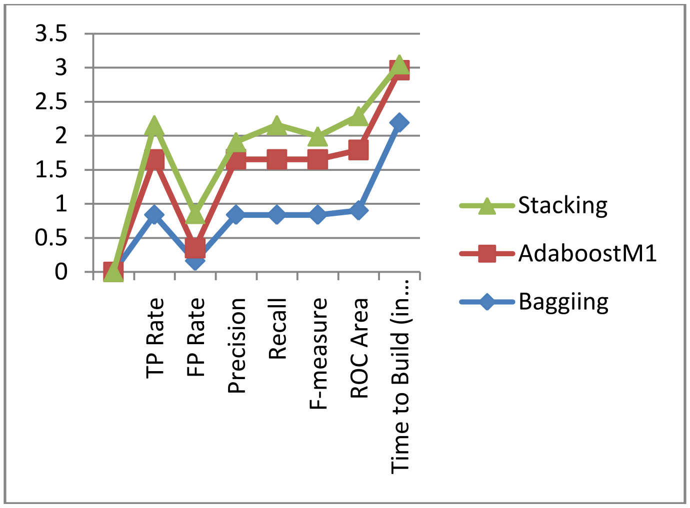
(Attribute Comparison using Dataset Hyperplane-1)

Classification Algorithm Type: Meta Dataset: Hyperplane1	
Algorithm	Accuracy
Bagging	84.54
AdaboostM1	81.25
Stacking	50.33



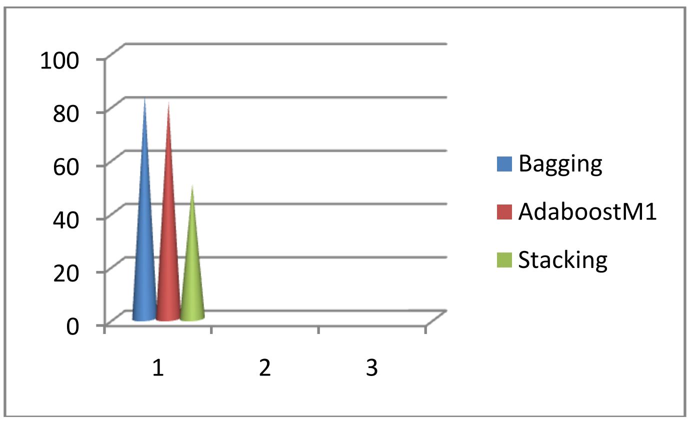
(Accuracy comparison using Dataset Hyperplane-1)

Classification Algorithm Type: Meta Dataset: Hyperplane2			
Algorithm / Attribute	Baggiing	AdaboostM1	Stacking
TP Rate	0.838	0.814	0.505
FP Rate	0.162	0.186	0.505
Precision	0.838	0.815	0.255
Recall	0.838	0.814	0.505
F-measure	0.838	0.814	0.339
ROC Area	0.901	0.889	0.5
Time to Build (in Sec)	2.19	0.77	0.09



(Attribute Comparison using Dataset Hyperplane-2)

Classification Algorithm Type: Meta Dataset: Hyperplane1	
Algorithm	**Accuracy**
Bagging	83.83
AdaboostM1	81.37
Stacking	50.51



(Accuracy comparison using Dataset Hyperplane-2)

The objective of data mining algorithm is to get best among available. In this paper different classification algorithms are used for performance evaluation. These algorithms are compared on the basis of execution time, TP Rate, FP Rate, Precision, Recall, Accuracy, and ROC. WEKA tool is used to evaluate and investigate the performance of three different categories of classifiers. The percentage of correctly classified instances is often called accuracy. We compare all the parameters and found that bagging classifier performs best among all three. According to the result the performance orders (highest to lowest) of the algorithms are: Bagging, AdaboostM1, and Stacking.

III. REFERENCES:

[1]. Breiman, L.: Stacked Regressions, Machine Learning 24: 1, pp. 49--64 (1996).

[2]. IwanSyarif, Ed Zaluska, Adam Prugel-Bennett, Gary Wills, Application of Bagging, Boosting and Stacking to Intrusion Detection, MLDM 2012: Machine Learning and Data Mining in Pattern Recognition pp. 593-602.

[3]. Magdalena Graczyk, TadeuszLasota, Bogdan Trawiński, Krzysztof Trawiński, Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal, ACIIDS 2010: Intelligent Information and Database Systems pp. 340-350.

[4]. P. Kalaiselvi and C. Nalini, "A Comparative Study of Meta Classifier Algorithms on Multiple Datasets", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, issue 3, pp. 654-659, 2013.

[5]. Breiman, L.: Bagging Predictors. Machine Learning 24:2, pp. 123-140 (1996).

[6]. Singal Swati, Jain Monika, "A study on WEKA tool for data preprocessing, classification and clustering", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013, pp. 250-253.

[7]. Smith, J., W., Everthart, J.,E., Dickson, W.,C., Knowler, W.,C. and Johannes, R.,S.,Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, Proceedings of the Symposium on Computer Applications and Medical Care, IEEE Computer Society Press, 1988, pp. 261-265.

[8]. A.Khemphila, V.Boojing, "Comparing Performance of logistic regression, decision tree and neural network for classifying heart disease patients", Proceeding of International conference on Com- puter Information System and Industrial Management Application, 2010, pp.193-198.

[9]. K.Srinivas, B.Kavitha Rani,A.Govrdhan, Applications of Data Mining Techniques in +health care and Prediction Heart Attacks, International Journal on Computer Science and Engineering (IJCSE), vol.II, 2010, pp.250-255.

[10]. Elma kolce(cela), NekiFrasheri "A Literature Review of Data Mining Techniques used in Healthcare Databases", ICT Innovations 2012 Web Proceedings – Poster Session ISSN 1857-7288.

[11]. Huy Nguyen Anh Pham and Evangelos Triantaphyllou, "Prediction of Diabetes by Employing a New Data Mining Approach Which Balances Fitting and Generalization" Computer and Information Science (Springer), pp 11-26.

[12]. Karthikeyini.V., Pervinbegum.I., "Comparison a performance of data mining algorithms (CPDMA) in prediction of Diabetes Disease", International journal of Computer Science and Engineer- ing, Vol.5, No. 03, March 2013, pp. 205-210.

[13]. Karthikeyini.V., Pervinbegum.I., Tajuddin.K., Shahina Begum, "Comparative of data mining classification algorithm (CDMCA) Diabetes Disease Prediction", International journal of Computer Applications, Vol.60, No. 12, Dec. 2012, pp. 26-31

[14]. K.R. Lakshmi and S. Prem Kumar, Utilization of Data Mining Techniques for Prediction of Diabetes Disease

Survivability. IJSER, Vol. 4, Issue 6, June 2013, ISSN 2229-5518, pp. 933-942.

Web References:

[1] http://www.sussex.ac.uk/Users/christ/crs/ml/lec03a.html

[2] http:// weka.sourceforge.net/doc.dev/weka/filters/supervised

[3] http://chem-eng.utoronto.ca/~datamining/dmc/zeror.htm

[4] http://chem-eng.utoronto.ca/~datamining/dmc/oner.htm