Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparison of Performance of Various Data Classification Algorithms with Ensemble Methods Using RAPIDMINER

Thendral Puyalnithi, Madhu Viswanatham V, Ashmeet Singh
School of Computing Science and Engineering, VIT University, Vellore,
Tamilnadu, India

Abstract— Data Mining techniques are helpful in finding out patterns between data attributes and results in probabilistic prediction of the label attribute. The paper discusses different classification techniques on small and large datasets. The two datasets are example datasets used from repository sites depending upon the number of instances. These datasets were applied in different classifier like Random Forest, Naive Bayes and Decision Tree to identify the best classifier for small dataset and large dataset. This paper gives the study and analysis of various methodologies used for prediction Based on the study, Naïve Bayes is most suitable for small datasets and Decision Tree is suitable for large datasets based on the evaluation done in this paper using various methodologies driven by RapidMiner tool while equating precision, recall and accuracy. Bagging and boosting ensembling techniques are applied for improving classifications of testing dataset.

Keywords— Naive Bayes, Random Forest, Decision Tree, Bagging, Boosting, RapidMiner tool

I. INTRODUCTION

The main objective of paper is to study the impact of different classification algorithms in the prediction of unknown label attributes. The parameters for judging the algorithms are accuracy, recall and precision. These are helpful when training data is used instead of testing data i.e. finding out the value for known values and comparing them to know the accuracy, recall and precision of the particular algorithm. This paper is catalogued as follows. Section II lists a related work. Section III discusses the datasets used and presents the aspects of classification algorithm and respective datasets under the proposed methods. Section V elaborates Analysis and presents the aspects of classification algorithm and respective datasets under the proposed methods. Section VI finalizes the results produced by the algorithms. Section VII provides the conclusion and the Section VIII gives the references.

II. RELATED WORKS

Mrs. M.S. Mythili and Dr. A.R.MohamedShanavas used data mining methodologies, such as decision table, IB1, J48, Multilayer Perceptron and Random Forest, to study and analyze the performance of the school students. The conclusion came out was that the Random Forest is the best classifier for analyzing the school students' performance result. It consumes less time and has good accuracy in [3].The classification results of Jehad Ali shows that the Random Forest gives better results for large datasets keeping the same number of attributes while J48 is best and easy approach for small datasets i.e. less number of instances in [4]. AmitGupte and his team too found Random Forest at top of all the other algorithms on their dataset of sentiment analysis. Sentiment analysis is a task which involves extraction of information from customers' feedbacks and other authentic sources such as survey agencies. Considering sentiment analysis the Random Forest classifier again has high accuracy and performance, simplicity in understanding, and improvement in results over a period of time. This results the classifier to best fit for situations like sentiment analysis in [5].

III. DATASETS

Datasets discussed in the paper are purely judged on the number of instances. There are basically two datasets used to judge the potential of different algorithms. The number of instances of two datasets is 498 and 30161.

1) Dataset of 30161 instances (Large)

This dataset aimed at the case of customers default payments in Taiwan. The dataset enrol a binary variable, default payment ('<=50' or ' >50' i.e. boolean), as the response variable. This dataset used the 14 variables as regular attributes and one as a label attribute. The dataset has 32561 instances in total among which 2400 instances have missing values in [9].

2) Dataset of 498 instances (Small)

CM1/Software defect prediction creator was a NASA spacecraft instrument. It was a NASA metrics data program which was written in C language. These metrics were segment based or it may call as function or method. CM1 has 498 numbers of instances which is the prior focus of using this dataset. Unlike the large dataset, none of the instances in this dataset had missing value in [8].

IV. PROPOSED METHOD

A. Naive Bayes

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. The Bayes Theorem: P(h/D)= P(D/h) P(h) P(D);P(h) : Prior probability of hypothesis h; P(D) : Prior probability of training data D; P(h/D): Probability of h given D; P(D/h): Probability of D given h.Moving on to the design part, RAPIDMINERtool has testing and training sections. The design consists of two parts (i)Preprocessing and (ii) Applying Model. In preprocessing, the design is from input data and filter is applied if there is any missing value in the data.The 10 fold validation is done using X-Validation operator.Inside X-Validation operator the classification model i.e. Naive Bayes is applied using the 'Apply operator' and performance is measured using 'Performance operator '.[6]

B. Random Forest

The Random Forests algorithm is one of the best among classification algorithms - able to classify large amounts of data with accuracy.

1. If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2. If there are M input variables, a number mM is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3. Each tree is grown to the largest extent possible. Pruning is restricted just to get more accuracy compromising increased execution time.

Similar to Naive Bayes design Random Forest design also consists of two parts (i)Preprocessing and (ii)Applying Model. All other operators are same except the replacement of naive bayes with the random forest.[7]

C. Decision Tree

A decision tree is a classifier which classifies an input sample into one of its possible classes.It is a tree structured classifier which makes decision rules from the large amount data to extract knowledge. A decision tree classifier uses a simple form which is concisely stored and that efficiently classifies new data.The advantages of decision tree in data mining 1) Its ability to handle different input data types such as, numerical, textual and nominal. 2) It can even take care of datasets whose instances have missing values and errors. 3) It is available in various packages of data mining and number of platforms.[2]

D. Bagging

Bagging is similar to the idea of subject expects in a work environment. When the president of the United States needs to make policy decisions, he relies on the expertise of his cabinet members to make the correct decision with respect to the policy. The expertise of the cabinet members compliment each other as opposed to being redundant and duplicative. Using this conceptual example of bagging, we can apply it to machine learning concepts familiar to us.Given some database of training data, we can take t samples from this database with replacement. Using samples taken from the training example database, we can train our machine learning algorithm independently on each of these datasets. After the training has completed, we are left with C_{t} classifiers, these are analogous to the cabinet members mentioned in the example. When presented with some unknown example, we make a prediction on it using each of the C_{t} classifiers. The final prediction is made by selecting the most common prediction from each of the classifier's C_{t}. The finial classification of the test example made from the target classifiers is called a voting scheme where the prediction of each target classifier is a "vote" towards the final prediction.

E. Boosting

There are many variants of the Boosting technique, here we discuss AdaBoost.M1. AdaBoost.M1 is similar to the bagging voting scheme however, votes are now weighted. The weights of these votes is increased fore more incorrectly classified examples (harder classifications) and decreased for more correctly classified examples (easier classifications).The AdaBoost.M1 algorithm can be performed in several steps similar to those of the bagging technique. First, weights are assigned to all examples in the training set database. For each training set sample generated, apply the machine learning algorithm of choice. Compute the classification error for each target classifier. If the classifier has an accuracy greater than 50% discontinue training, otherwise multiply the weight w of each object by e/(1-e) and then normalize the weights of all of the training examples. Now, on the testing set, apply a weight of q=0 to each class to be predicted. For each classifier, add -loge/(1-e) to the weight of the decision predicted by the current classifier. From all the classifier weights, output the decision with the highest classifier weight q.

V. ANALYSIS

The confusion matrix shows the relation between the actual and the predicted values. The three important factors decides the strength of the algorithm and its result - Accuracy, precision and recall. Class Precision: Percentage of correctly predicted values. Class Recall: Percentage of actual values were predicted correctly. Accuracy: It’s the overall correctly predicted values.Average of precision and recall of each class is taken to give overall precision and recall of the classifier. The accuracy of the classifiers are given by true positive rate, false positive rate, precision, recall and F-measures using Rapid Miner tool. Rapid Miner is a powerful software platform that gives an integrated environment for machine learning, data mining, text mining and other business and prediction analysis. The average of measures from all the classes has been taken to give the overall measure for classifiers. For example, to give the overall precision for a classifier for a given dataset, average of precisions of both (true/false) classes is calculated.

A. Accuracy

Accuracy is calculated as number of instances predicted positively divided by Total number of instances. This means accuracy is the percentage of the accurately predicted classes among the total classes. In the experiment the values of the accuracy posted into table in the basis of 0 to 100, not from 0 to 1.

Accuracy = ((True Positive + True Negative) / (P + N))*100

B. Precision

Precision is the preciseness or exactness of truly classified class, therefore known as positive predictive value. It is the proportion of instances which truly have class x / Total classified as class x. So basically high precision stated the accurate results and it takes all relevant data but returns only topmost results. In short, it is the number of chosen items which were related.

Precision =(True Positive / (True Positive + False Positive))*100

C. Recall

Recall gives sensitivity of problem and it process values or product quantity or completeness. It returned most relevant and part of the documents that are relevant as result from the query. In other words, modules that are really recognize as difficult to maintain from the total number of modules. In short, it is the number of related objects that were chosen.

Recall = (True Positive / (True Positive + False Negative))*100

D. True Positive (TP)

True positive are the positive tuples which were correctly labeled by the classifier. It is the proportion categorized as class x / Actual total in class x. True positive projected by the modules that are predicted positively as the results specified at the end.

True Positive rate = (True Positive / (True Positive + False Negative))*100

E. False Positive (FP)

False positive, proportion incorrectly categorized as class x / Actual total of all classes, except x. It is incorrectly predicted compared to original results.

False Positive rate = (False Positive / (False Positive + True Negative))*100

F. F-Measure

F-Measure categorized as (2*Precision*Recall / (Precision+ Recall))*100. It is a combined measure for precision and recall.

VI. RESULTS

In the analysis of datasets, one attribute was taken as label attribute which was used for classification of instances. Using Rapidminer tool, CM1(of 498 instances) and E-commerce(of 30161 instances) datasets were applied to classifiers Naive Bayes, Random Forest and Decision Tree algorithms. The rapidminer has been used to classify the testing data which was done using 10-fold validation. In the case of random forest and decision tree, pruning and pre-pruning is applied for better results by compromising the execution time. The bootstrap validation is used in ensembling techniques such as baging and boosting. Fifty or ten , which is better, fold validation is used for ensembling.The Results of following analysis on two datasets are clearly given by the tables I,II,III and IV. Tables I and III have given the instances correctly classified and in-accurately classified with total number instances in dataset using different classifiers. Tables II and IV listed the accuracy, true positive rate, false positive rate, precision, recall and F-measures to analyse the classifiers. Also it provides best classifier by highlighted based on precision value. Tables V and VI are used to give comparison of all the three algorithms on both the datasets after applying bagging and boosting. The figures I,II,III and IV are drawn for representing the tables II, IV, V and VI.

Table I Of Smaller Dataset (498 Instances)

Method	**Appropriately Classified Instances**	**Appropriately Not Classified Instances**	**Total Instances**
Naive Bayes	420	78	498
Random Forest	445	53	498
Decision Tree	418	80	498

Table II Analysis on smaller Dataset (498 instances)

	Accu-racy	**Precision**	**Recall**	**TP Rate**	**FP Rate**	**F-Measu--re**
Naive Bayes	82.74	78.26	72.19	72.19	27.81	75.10
Random Forest	77.11	87.64	58.22	58.22	41.79	70.0
Decision Tree	81.42	81.21	75.55	75.55	24.46	78.27

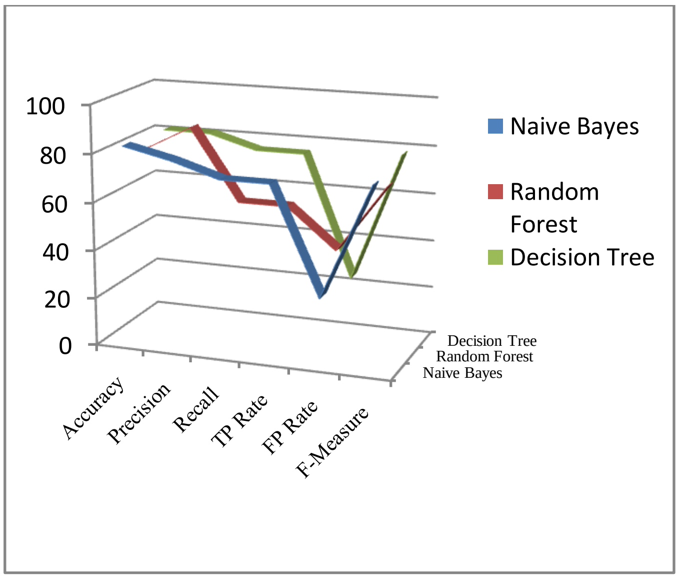

Fig. I. Describes the ratio of each classifier for CM1 dataset based on Table II.

Table III Classified Instances of smaller Dataset (30161instances)

Method	**Appropriately Classified Instances**	**Appropriately Not Classified Instances**	**Total Instances**
Naive Bayes	24951	5210	30161
Random Forest	24069	6092	30161
Decision Tree	25558	4603	30161

Table IV Analysis on smaller Dataset (30000 instances)

	Accu-racy	**Preci-sion**	**Recall**	**TP Rate**	**FP Rate**	**F-measu-re**
Naive Bayes	84.34	58.84	60.41	60.41	39.60	59.61
Random Forest	89.96	61.82	50.80	50.80	47.50	55.77
Decision Tree	89.97	65.24	51.71	51.71	43.45	57.69

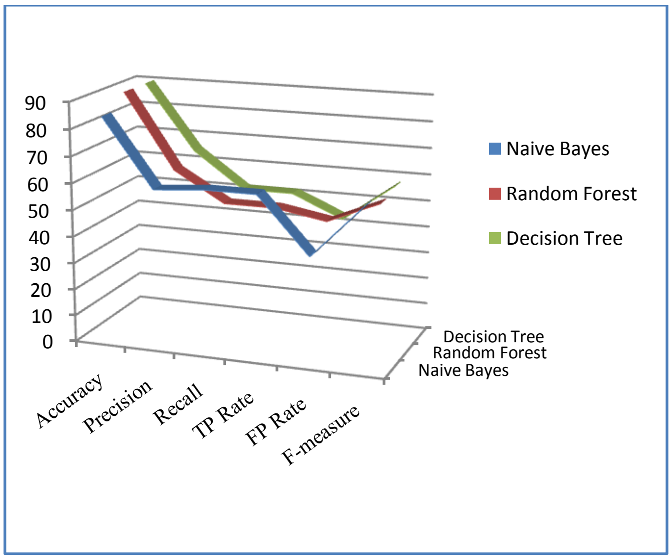

Fig.II. Describes the ratio of each classifier for credit-scanning dataset based on Table IV.

Table V Analysis on smaller Dataset-Boosting & Bagging (498 instances)

	Accuracy	**Precision**	**Recall**	**F-Measure**
Naive Bayes	84.34	58.84	60.41	59.61
Naive Bayes (boosting)	78.69	55.58	59.79	57.61
Naive Bayes (bagging)	82.61	57.94	70	63.40
Random Forest	89.96	61.82	50.8	55.78
Random Forest (boosting)	89.95	58.34	51.51	54.71
Random Forest (bagging)	91.11	69.08	51.94	59.30
Decision Tree	89.96	65.24	51.71	57.69
Decision Tree (boosting)	89.34	57.48	51.99	54.60
Decision Tree (bagging)	84.57	56.25	55.5	55.87

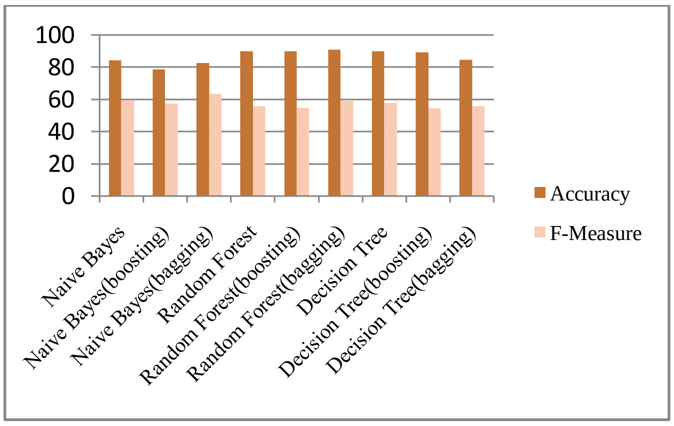

Fig.III.Describes the ratio of each classifier for CM1 dataset based on Table V.

Table VI Analysis on Larger Dataset-Boosting & Bagging (30161 instances)

	Accuracy	**Precision**	**Recall**	**F-Measure**
Naive Bayes	82.73	78.26	72.19	75.10
Naive Bayes (boosting)	57.93	52.33	52.88	52.60
Naive Bayes (bagging)	83.32	78.52	72.73	75.51
Random Forest	77.11	87.61	54.06	66.86
Random Forest (boosting)	80.52	86.18	60.14	70.84
Random Forest (bagging)	80.51	88.54	59.82	71.40
Decision Tree	81.42	86.62	54.06	66.57
Decision Tree (boosting)	82.16	87.13	63.7	73.59
Decision Tree (bagging)	85.1	81.35	75.50	78.31

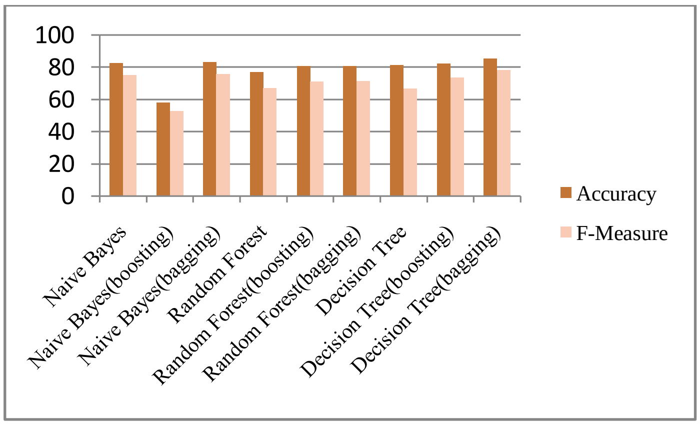

Fig.IV.Describes the ratio of each classifier for credit-scanning dataset based on Table VI.

VII. CONCLUSION

In this study, Naive Bayes results better results than other two in smaller dataset whereas Decision Tree is best for larger dataset. Therefore, Random forest acts as an average in both the cases. This happened because random forest takes large set of data to learn but the fails in these datasets as they have one thing in common. i.e. much lesser amount of data for true instances. Therefore, as the number of instances having 'True' value were less in number, it was easier for Naive

Bayes classifier to learn and respond better than others. But, in the case of adaboosting, Random Forests' performance remained constant whereas others accuracy and f-measure was decreased in CM1 dataset(smaller). In credit-scanning dataset (larger), Bossting effect can been seen in Random Forest and Decision Tree but performance of Naive bayes was decreased drastically. Adaboosting affected Random Forest the most but Decision Tree had highest overall accuracy after boosting. In bagging of classifiers considering small dataset, Naive bayes and random Forest classified better whereas performance of Decision Tree decreased by a bit. But in large dataset, naive bayes had little impact in the improvement of performance but the other two showed much better results. Decision tree succeeded the most in bagging of classifiers in large dataset.

Therefore, Random forest had a positive impact of bagging and boosting in which boosting had bi more impact on it. It works well in large datasets.

Decision Tree being topper in classifying accuracy results for large datasets but acted best bagging in done along. has major impact on smaller datasets but boosting was a big fail for it in every case. It had somewhat better results with bagging technique for large datasets but was negligible considering others result.

REFERENCES

[1] S.L. Ting, W.H. Ip, Albert H.C. Tsang "Is Naïve Bayes a Good Classifier for Document Classification?", International Journal of Software Engineering and Its Applications,Vol. 5, No. 3, July, 2011.

[2] ShahrukhTeli, PrashastiKanikar" A Survey on Decision Tree Based Approaches in Data Mining" , International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 4, 2015.

[3] Mrs. M.S. Mythili, Dr.A.R.MohamedShanavas, "An Analysis of students' performance using classification algorithms", IOSR Journal of Computer Engineering, 2278-8727Volume 16, Issue 1, Ver. III (Jan. 2014).

[4] Jehad Ali, Rehanullah Khan, NasirAhmad,ImranMaqsood," Random Forests and Decision Trees", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012.

[5] AmitGupte, Sourabh Joshi, Pratik Gadgul, AkshayKadam," Comparative Study of Classification Algorithms used in Sentiment Analysis", International Journal of Computer Science and Information Technologies, Vol. 5 (5), 2014.

[6] Khosrow-Pour, Mehdi,"Encyclopedia of Information Science and Technology", First Edition, January 31, 2005 IGI Global.

[7] Saurbhkumar, Dr. Manish mann ,"E-Mail Filtering For The Removal Of Misclassification Error",International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 2, Issue 12, December 2015.

[8] CM1 dataset, Promise Software Engineering Repository, "http://promise.site.uottawa.ca/SERepository/datasets/cm1.arff", December 2, 2004.

[9] Credit Screening, UCI Machine Learning Repository, "http://archive.ics.uci.edu/ml/machine-learning-databases/credit-screening/ ".