CAN MULTIPLE MODELS IMPROVE BAYESIAN'S PERFORMANCE? AN INVESTIGATION USING MCNEMAR'S TEST

M. Tahir, A. Shaukat, and N. Kanwal

Department of Computer Science, LCWU, Lahore, Pakistan
Corresponding Author email: nadia.kanwal@lcwu.edu.pk

ABSTRACT: Machine learning algorithms have been widely used for classification purposes in a number of research domains; however, very few researches paid any attention to statistically validate the performance of these algorithms for different data. This paper attempted to study the Naïve Bayes algorithm's performance for dataset of different sizes. Furthermore, a known theory has also been investigated, that building multiple models such as Bagging, Boosting and Stacking tend to improve a classifier's performance. The analysis has been performed using McNemar's test; a well known non-parametric statistical test in the medical analysis domain. Results showed that not all ensemble methods work as expected and therefore, needs to be selected carefully. Moreover, the use of McNemar's test appeared to be simple, but gave statistically valid results.

Keywords: McNemar's Test, Naïve Bayes, Boosting, Bagging, Stacking, Performance Evaluation, Classification

(Received 25-10-2014 accepted 15-12-2015).

INTRODUCTION

Naïve Bayes is a widely used classifier for many classification tasks like Data Mining (Singh, 2014), Classification of Text (Rennie *et al*., 2003) and Classification of Internet Traffic (Zhang, 2013). It is considered more appropriate for small dataset because it needs less training data. However, just like other classification algorithms this is not a consistent behavior because of changing characteristics and dimensions of datasets; for example Naïve Bayes may perform well in Data Mining applications (Rosen *et al*., 2010 and Ting *et al*., 2011) but do not classify image data correctly such as in medical images (Ting *et al*., 2011). There are a number of reported results where ensemble methods are likely to improve a classifier's performance (Breiman, 1996; Schapire, 2002 and Dzeroski and Zenko, 2004) such as Bagging, Boosting and Stacking. This research looks into the effect of these ensemble methods in view of data size and dimensionality as compared to independent classifier's results. This investigation has been carried out using a non-parametric test called McNemar's test, commonly used in the medical field for identifying medicines' effects on patients (Frotingham, 2001, Durkalski *et al*., 2003 and Uemura, 2001).

WEKA, a machine learning software, described in section 4, is used to classify selected datasets using Naïve Bayes, Bagging, Boosting, and Stacking. Along with a number of data visualization tools it also provides multiple testing measures such as confusion matrix, Precision, Recall, Kappa statistics, etc. These classification results were then analyzed using McNemar's test (Hall *et al*., 2009).

There are a number of classification algorithms designed and tested for different types of data, such as Decision Trees, Neural Networks, Bayesian classifiers, Support Vector Machines and the combination of one or more classifiers (Alpaydin, 2004). Combining one or more classifiers' output for final predictions is known as an ensemble method that may improve the classification accuracy of individual algorithms. The trick is to build multiple models of the same data and use them to classify the new instances. Multiple predictions by the same classifier for the same data set are combined in the Bagging and average of the predictions is considered as the final predictions (Breiman, 1996). Boosting, significantly boosts the accuracy of a weak classifier through accuracy and confidence measures (Freund and Schapire, 1996). Another method is called stacking that used to combine more than one classifier by using them as a base level classifier and a Meta classifier to compile final results (Dzeroski and Zenko, 2004). In short Bagging, Boosting, and Stacking can be used to improve the performance of any classifier, but the question is, does this always happen? This analysis is an attempt to answer this question.

The data can be easily classified using these methods using WEKA's cross validation routine. For consistency in comparisons, WEKA's 10-fold cross validation is used for performance evaluation of classifiers. 10-fold cross validation is a model validation technique in which a randomly selected subset of data is periodically used for validation check.

The rest of the paper is organized as follows, section 2 details the materials and methods used, section 3 describes the results of the tests performed, and lastly section 4 concludes the findings.

MATERIALS AND METHODS

To evaluate the performance of Naïve Bayes classifier and ensemble methods, three datasets were obtained as shown in Table 1. First dataset "Students" has been used for the classification of studnets' in groups using Naïve Bayes algorithm (Singh *et al.,* 2014), however, the results as reported by (Singh *et al.,* 2014), showed calculation errors. Therefore, it was considered valuable to investigate the behavior of this classifier for "Students" data*et al*ong with others. Moreover, the rationale behind the selection of these datasets was to include variety of datasets with different sizes and different number of classes.

Table 1. Data sets' description

Reference	Dataset	# of Instances	# of Attributes	# of Classes
Singh *et al.,* 2014	Students	50	7	4
Lichman, 2013	Student's Knowledge about Electrical DC Machine	403	5	4
Lichman, 2013	Tic-Tac-Toe	958	9	2
Lichman, 2013	Yeast	1484	8	10
Lichman, 2013	Car Evaluation	1728	6	4

Mcnemar's Test: McNemar's test introduced by (McNemar, 1947), is a non-parametric test that was used to identify statistically significant performance differences between the two algorithms for paired data (Bostanci and Bostanci, 2013 and Alpaydin, 2004). One can also refer to (McNemar, 1947 and Clark and Clark, 1999) for the detailed description of the test. The beauty of this test was that it used the cases where two algorithms performed differently instead of focusing on correct classification results. Table 2 explains the possible outcome of a classification task. These values could be Fail-Pass (*FP*), Pass-Fail (*PF*), Pass-Pass (*PP*), and Fail-Fail (*FF*).

Table 2. Possible outcomes of two classifiers with respect to ground-truth.

Classifier	Classifier 1 Failed	Classifier 1 Passed
Classifier 2 Failed	*FF*	*FP*
Classifier 2 Passed	*PF*	*PP*

***FP*:** number of correctly classified instances by the classifier 2 but misclassified by classifier 1.
***PF*:** number of correctly classified instances by the classifier 1 but misclassified by classifier 2.
***PP*:** number of correctly classified instances by both the classifiers.
***FF*:** number of incorrectly classified instances by both the classifiers.

The outcomes when both the classifiers had same results were not useful because they did not tell us about the difference between the performances of two classifiers (Clark and Clark, 1999). Therefore, *FF* and *PP* numbers in the table did not add any information in performance analysis; however, *FP* and *PF* actually depicted the performance differences of the two algorithms. Z scores were calculated to find statistically significant performance difference between the two classifiers. Z score was calculated as is shown in Eq. 2.

$$z = \frac{(|FP-PF|-1)}{\sqrt{FP+PF}} \qquad (2)$$

Table 3. Converting Z scores into confidences for α = 0.05.

Z score	Degree of Confidence Two-tailed Prediction	Degree of Confidence One-tailed Prediction
1.645	90%	95%
1.960	95%	97.5%
2.326	98%	99%
2.576	99%	99.5%

Z-Scores in Table 3 showed the significance level. A zero score showed that classifiers behave similarly. However, a greater value showed different behavior of classifiers and significant level if it was more than defined critical values. To identify critical Z-score, a significance level denoted by α was used. The value of α was set to 0.05 means probability of 1 in 20 for the rejection of hypothesis. In other words we have 95% confidence that the two algorithms were performing differently.

One could associate confidence limits with Z-scores given in Table 3. Two-tailed predictions were used when the purpose was to find the performance difference between two algorithms and one-tailed prediction if one had to find out which algorithm was better than the other. Furthermore, the maximum of *PF* and *FP* values were used to identify the classifier with better performance of the two. Table 2 showed McNemar's populate. For each pair of algorithms, WEKA's output predictions were enabled to calculate Z-scores from number of PF and FP.

In addition to this two more performance measures were used to find similarity in tests, Kappa statistics and Root Mean Squared Error. Kappa statistic was the measure of the accuracy between actual values and predicted values (Bostanci and Bostanci, 2013 and Othman and Yan, 2007). More the value of Kappa Statistics was closed to 1 it means accurate classification. Similarly, the error between the actual value and the predicted value was a very important parameter to judge an algorithm's performance this was known as RMSE (Bostanci and Bostanci, 2013). Zero or minimum value of RMSE means that classifier had a good performance.

RESULTS AND DISCUSSION

McNemar's test results and WEKA statistics were calculated and discussed below for each dataset from two different perspectives; first to find a machine learning algorithm which classified the data correctly, secondly the effect of data size on these algorithms' performance. Furthermore, it was also interesting to see that if McNemar's test results agreed to other statistics such as Kappa statistics and RMSE.

Pair-wise comparison between classifiers: In Table 4 the Z-scores representing a performance comparison of pair of algorithms were given in the intersecting cell of each row and column. As mentioned before a Z-score less than 1.96 depicted similar behavior of algorithms therefore, in the table a Z-score with no arrowhead showed that the algorithms were behaving similarly while in other cells, the direction of arrowhead indicated the one with significantly better performance.

For-example for Car Evaluation dataset the Z-score between Naïve Bayes and Bagging was *1.106* which implied that both algorithms classify this data similarly and there was no statistically significant difference between them for this specific dataset However, the Z-score of *6.168* between Naïve Byes and boosting showed a significant performance difference and the direction of the arrowhead indicates boosting to be performing better similar to (Schapire, 2002).

Table 4. McNemar's Test results

Classifier	Boosting	Bagging	Stacking
Car Evaluation			
Naïve Bayes	↑ 6.168	1.106	↑ 15.011
Boosting		← 6.595	↑ 12.212
Bagging			↑ 15.154
Students			
Naïve Bayes	0.288	0.707	1.032
Boosting		0.801	0.516
Bagging			1.549
Student's Knowledge about Electrical DC Machine			
Naïve Bayes	0.948	0	← 4.002
Boosting		0.516	← 3.323
Bagging			← 3.973
Yeast			
Naïve Bayes	← 3.902	↑ 2.285	0.267
Boosting		↑ 4.773	↑ 3.918
Bagging			0.725
Tic-Toc-Toe			
Naïve Bayes	↑ 11.217	0.5547	↑ 16.941
Boosting		↑ 10.859	↑ 12.247
Bagging			↑ 16.852

For Car Evaluation and Tic-Tac-Toe datasets the best algorithm appeared to be the Stacking method which outperformed all algorithms with the highest Z-scores. However, its performance was not significantly better for Student's Knowledge, Students and Yeast datasets as it either lost against every algorithm or performed similarly. Although analysis reported in (Dzeroski and Zenko, 2004) proved stacking to be better, but they ignored the effect of amount of data.

Analysis of Boosting vs. Bagging revealed that it performed better only for two datasets *i.e.* Yeast and Tic-Tac-Toe otherwise, mostly it performed similar to Naïve Bayes. Similarly, Bagging added no benefit to Naïve Bayes's performance for most of the datasets. Hence, in two datasets *i.e.* Students, and Student's Knowledge about Electrical DC Machine, performance of Naïve Bayes was not improved by the use of multiple models. In Car Evaluation dataset Boosting, and Stacking

enhanced the performance of Naïve Bayes Classifier but Bagging had no effect on its performance (Z score= 0) which means there was no significant difference between Naïve Bayes and Bagging. These results highlighted the importance of data characteristics before the selection of classifiers which has been overlooked in previous studies (Schapire 2002, Rennie *et al.* 2003, Zhang *et al.* 2013 and Singh *et al*. 2014).

Comparison with Kappa Statistics and RMSE: To verify the results discussed before following section presents WEKA statistics and its comparison with McNemar's test. McNemar's test showed that for larger datasets, multiple models improve the performance of Naive Bayes classifier. In this section these results were compared with Kappa Statistics and RMSE.

For Car Evaluation dataset Kappa Statistics showed that Stacking and Boosting improved the performance of Naïve Bayes but Bagging had no significant effect on it; an agreement with McNemar's test as shown in Table 5 and Fig 1. Similarly, Kappa Statistics for Student's Knowledge about Electrical DC Machine showed that there is no significant difference in the performance of classifiers and Stacking decreased the performance of Naïve Bayes rather than increasing. For Yeast dataset Kappa Statistics did not show any significant improvement in the performance of all classifiers. Same as for Students dataset Naïve Bayes and Bagging had no significant difference in their performance, whereas Boosting and Stacking reduced the performance of Naïve Bayes.

Table 5. Kappa statistics for all datasets

S. No.	Classifier	Data sets				
		Car Evaluation	Student's Knowledge	Yeast	Students	Tic-Tac-Toe
1	**Naïve Bayes**	0.6665	0.8471	0.4541	0.4337	0.2843
2	**Bagging**	0.6531	0.8416	0.4686	0.4859	0.289
3	**Boosting**	0.7838	0.8252	0.4029	0.3736	0.6321
4	**Stacking**	0.9811	0.6786	0.4491	0.3011	0.9954

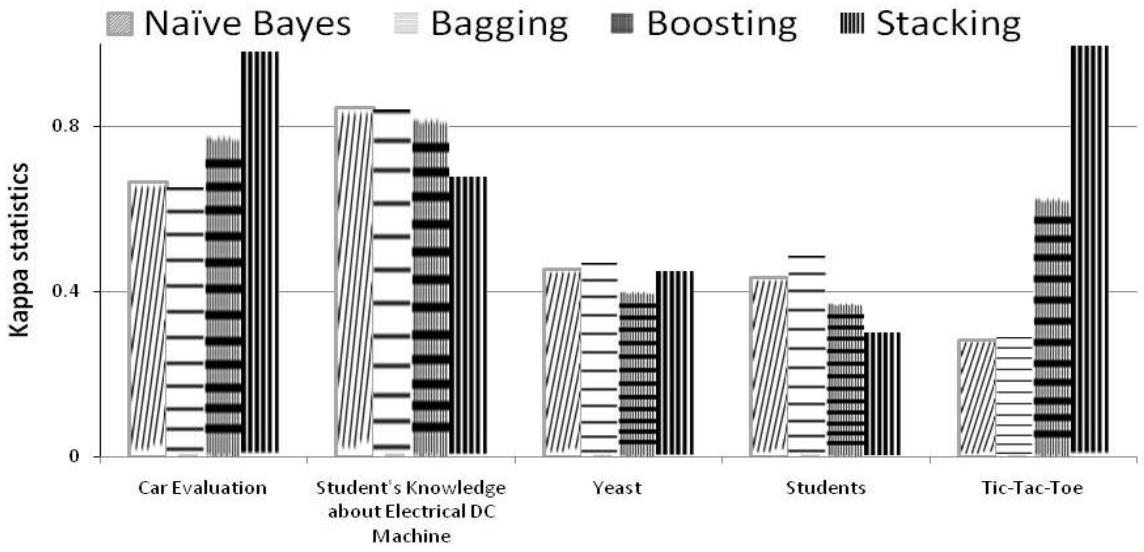

Fig 1. Graphical representation of Kappa Statistics for all datasets

Table 6. RMSE results for all datasets

S. No.	Classifier	Data sets				
		Car Evaluation	Student's Knowledge	Yeast	Students	Tic-Tac-Toe
1	**Naïve Bayes**	0.2262	0.2352	0.2391	0.3767	0.4319
2	**Bagging**	0.2275	0.2317	0.2378	0.3685	0.4304
3	**Boosting**	0.1837	0.2168	0.2844	0.4341	0.3367
4	**Stacking**	0.0588	0.3077	0.2371	0.4474	0.0491

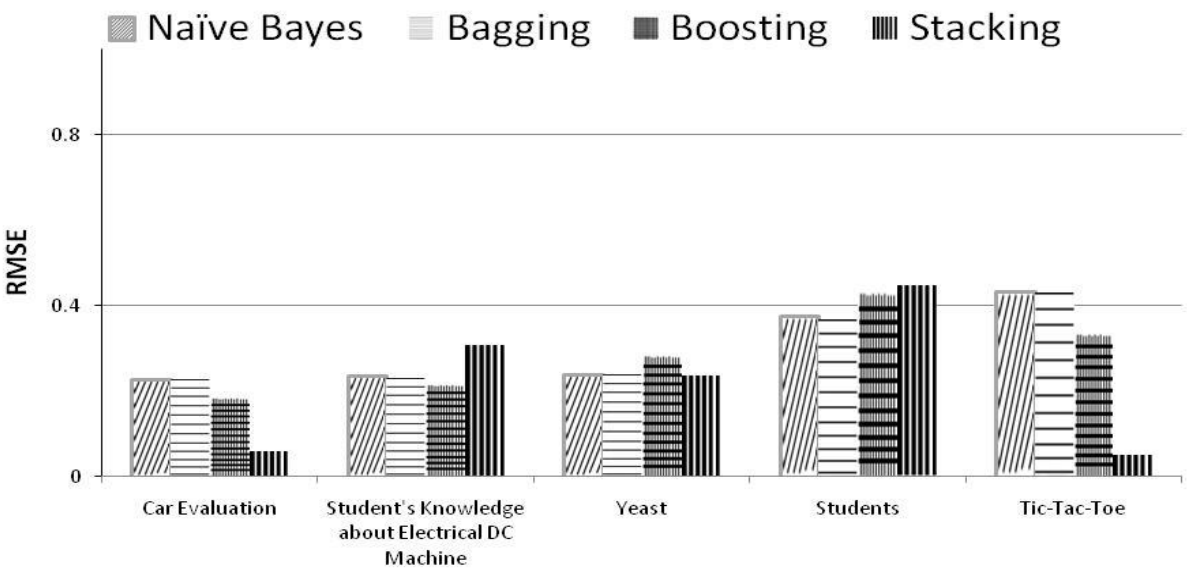

Fig 2. Graphical Representation of Root Mean Squared Error (RMSE) for all datasets

Kappa Statistics for Tic-Tac-Toe showed that Bagging did not improve the performance of Naïve Bayes but Boosting and Stacking had a very high impact on the performance of Naïve Bayes as they improved Naïve Bayes' performance significantly. The classification error of algorithms was consistent with the performance shown in the Kappa Statistics graph in Fig. 1.

RMSE for Car Evaluation showed that Stacking and Boosting reduced the classification error of Naïve Bayes, however, Bagging brought no significant difference in the performance. RMSE values obtained from Student's Knowledge about Electrical DC Machine and Students datasets showed that Bagging and Boosting did not contribute to performance improvement of Naïve Bayes and hence reduced the accuracy of original classifier. Yeast dataset's RMSE value showed that Bagging and Stacking performed similarly to Naïve Bayes but Boosting raised RMSE as shown in Table 6 and Fig 2. In general all of these results were compatible with McNemar's test results as found in (Bostanci and Bostanci, 2013), increasing the validity of this test for machine learning algorithms.

Effect of data size over algorithms' performance: It was also important to see if the dataset's size has any impact over the algorithm's performance because it could help us generalize the algorithms' ranking. Table 2 showed that Students dataset has only 50 instances and was the smallest dataset, however, this number was sufficiently large to apply the McNemar's test. According to the central limit theorem the minimum data should be greater than 30 to infer a statistically valid result (Clark and Clark, 1999). Therefore, McNemar's test results for Student's dataset were valid for comparison. Table 4 showed that all algorithms were performing similar for this dataset However, for Car Evaluation the performance differences were statistically significant. Hence, it was safe to conclude that if the data was small, using multiple models would increase the model construction time, however, it will not contribute to improve the classification accuracy, contrary to the results reported in (Schapire, 2002 and Ting *et al*., 2011)

Similarly, for Yeast dataset, there were more than 1400 instances as well as more number of classes to predict. It was interesting to see that for Yeast dataset the Z-scores were very low as compared to Car Evaluation dataset. This also highlighted the strength of McNemar's test that the test was capable of identifying statistically significant performance differences for large as well as small data. These results were found to be in line with (Clark and Clark, 1999 and Bostanci and Bostanci, 2013).

Conclusion: This study presented the performance comparison of Naïve Bayes algorithm with ensemble methods; where one expects significant improvement of classification results such as Bagging, Boosting and Stacking. McNemar's test, a relatively uncommon statistical test in machine learning domain was applied to see statistically significant differences in algorithms' performances. The results showed that for datasets with more number of instances, constructing multiple models improved the performance of the Naïve Bayes Classifier. However, Stacking should be used carefully because it can reduce the classification accuracy instead of improving it. The agreement of McNemar's test result with Kappa statistics and RMSE strengthened the confidence over these results. Both of these showed that the results obtained from Naïve Bayes can be improved by the use of multiple models. Furthermore, Boosting tends to help more in performance improvement than

Bagging and therefore, should be preferred. It was also useful to see that one simple test could be used to answer multiple questions such as whether two algorithms were different, and which one of them was better. Although this study was not adequate to generalize the algorithms' performances, but sufficient enough to highlight the differences which are statistically significant and should not be ignored.

REFERENCES

Alpaydin, E. (2004) Introduction to machine learning. MIT press, 360 p.

Bostanci, E., and B., Bostanci (2013). "An Evaluation of Classification Algorithms Using Mc Nemar's Test." In Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications: 15-26.

Breiman, L. (1996) "Bagging predictors." Machine Learning, 24(2): 123-140.

Clark, A. F., and C. Clark, (1999) Performance Characterization in Computer Vision. European Union's IST programme.

Durkalski, V. L., Palesch, Y. Y., Lipsitz, S. R., and F. P. Rust, (2003). Analysis of clustered matched-pair data. Statistics in medicine, 22(15):2417–2428.

Dzeroski, S., and B. Zenko. (2004) "Is combining classifiers with stacking better than selecting the best one?" Machine Learning, 54(3): 255-273.

Freund, Y., and R. .E Schapire. (1996) "Experiments with a new boosting algorithm." ICML 96: 148-156

Frotingham, R., (2001) "Rates of torsades de pointes associated with ciprofloxacin, ofloxacin, levofloxacin, gatifloxacin, and moxifloxacin." Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy, 21: 1468-1472.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, & H. I. Witten, (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

McNemar, Q. (1947) "Note on the sampling error of the difference between coorelated proportions or percentages." Psychometrika 12(2): 153-157.

Othman, M. F. bin, and T. M. S. Yan. (2006) "Comparison of different classification techniques using WEKA for breast cancer." Edited by Springer. 3rd Kuala Lumpur International Conference on Biomedical Engineering : 520-523.

Rennie, J. D., L. J. T Shih, and D. R. Karger. (2003) "Tackling the Poor Assumptions of Naive Bayes Text Classification." ICML 3: 616-623.

Rosen, G. L, E. R. Reichenberger, and A. M. Rosenfeld. (2010) "NBC: The Naive Bayes Classification tool Webservers for Taxonomic Classification of Metagenomic Reads." Bioinformatics (Oxford Press) 27(1): 127-129.

Schapire, R. E. (2002) "The boosting approach to machine learning: An overview." Nonlinear estimation and classification ,Springer : 149-171.

Singh Y., R., K A Soni, and S. Pal. (2014) "Implementation of Data Mining Techniques to Classify New Students into Their Classes: A Bayesian Approach." Journal of Computer Applications 85(11): 16-19 .

Ting, M. K., J. R Wells, C. S. Tan, W. S. Teng, and I. Geoffrey. (2011) "Feature-subspace aggregating: ensembles for stable and unstable learners." Machine Learning (Springer US) 82: 375-397.

Ting, S L, W. H. Ip, and H. C. T. Albert. (2011) "Is Naive Bayes a good classifier for document classification?" International Journal of Software Engineering and Its Applications ,5(3): 37.

Uemura N., S. Okamoto, S. Yamamoto, N. Matsumura, S. Yamaguchi, M. Yamakido, K. Taniyama, N. Sasaki, and S. J. Ronald, (2001) "Helicobacter pylori infection and the development of gastric cancer." New England Journal of Medicine, 345(11): 784-789.

Zhang J., C. Chen, Y. Xiang, W. Zhou, Y. Xiang, (2013) "Internet traffic classification by aggregating correlated naive bayes predictions." , IEEE Transactions on Information Forensics and Security, 8(1): 5-15.