A Survey of different classification techniques and their comparison using Mc Nemar's Test

Sandeep Kumar Patra
CH. Sharath Chandra Santosh Prasad

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela - 769 008, India

A Survey of different classification techniques and their comparison using Mc Nemar's Test

Dissertation submitted in

July 2013

To the department of

Computer Science and Engineering

Of

National Institute of Technology Rourkela

for the degree of Bachelors of Technology

By

Sandeep Kumar Patra
(109cs0172)
CH. Sharath Chandra Santosh Prasad
(109cs0489)

Under the supervision of
Prof. **Ramesh Kumar Mohapatra**

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela (769 008), India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008, India. (www.nitrkl.ac.in)

Ramesh Kumar Mohapatra
Professor

July 20, 2013

Certificate

This is to certify that the work in the thesis entitled "**A Survey of different classification techniques and their comparison using Mc Nemar's Test**", by **Sandeep Kumar Patra**(109CS0172) and **Ch. Sharath Chandra Santosh Prasad**(109CS0489), is a record of an original work carried out by them under my guidance and supervision for impartial fulfillment of the requirements for the award of the degree of Bachelors of Technology in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Ramesh Kumar Mohapatra

Acknowledgment

First of all, we would like to express my deep sense of respect and gratitude towards our supervisor Prof **Ramesh Kumar Mohapatra**, who has been the guiding force behind this work. We want to thank him for giving us the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for us to complete this thesis. We are greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. We consider it our good fortune to have got an opportunity to work with such a wonderful person. I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

CH. Sharath Chandra Santosh Prasad **Sandeep Kumar Patra**

Abstract

Five classification algorithms namely J48, Naive Bayes, K Nearest Neighbour, IBK and Decision Tree are evaluated using **Mc Nemar's** test over datasets including both nominal attributes and numeric attributes. It was found that K Nearest Neighbor performed better than the other classification methods for both nominal datasets and numerical datasets. It was also observed that the results of this evaluation confers with two other evaluation metrics used for evaluating classification algorithms or machine learning algorithms, Root Mean Squared Error and Kappa statistic.

Key words: Classifier Evaluation, Mc Nemar's test, Z score, Confidence level, Null Hypothesis.

Contents

Introduction

In today's world large quantities of data is being accumulated and seeking knowledge from massive data is one of the most fundamental attribute of Data Mining. It consists of more than just collecting and managing data but to analyze and predict also. Data could be large in two senses: in terms of size & in terms of dimensionality. Also there is a huge gap from the stored data to the knowledge that could be constructed from the data. Here comes the classification technique and its sub-mechanisms to arrange or place the data at its appropriate class for ease of identification and searching. Thus classification can be outlined as inevitable part of data mining and is gaining more popularity. Different types of classification techniques work differently for different datasets. Some techniques give better efficiency for a dataset of very large size but it might not be the optimal technique to use for a dataset with higher no. of attributes. Some techniques give better efficiency for a dataset with numeric attributes and some give better result with nominal attributes. Evaluating the performance of a machine learning method is as important as the algorithm itself because it identifies the strength and weakness of each classification algorithm or machine learning algorithm. This thesis enquires about the usage of Mc Nemar's test as an evaluation method for machine classification algorithms, systematically. Mc Nemar's test is used in different studies and other researches. Dietterich [1] examined 5 different statistical tests, one of them being Mc Nemar's test, to identify how these different tests differ in determining the performance of classification algorithms. A similar evaluation technique was performed on a very large database by Bouckaert [2]. Demsar [3] has evaluated decision tree, naive bayes and

k-nearest neighbor methods using several other non-parametric tests which include Analysis Of Variance method [4] and Friedman test [5, 6]. Other studies have shown classifiers using the same tests over a large sets but our method differs in this way that we use a different criterion that does a comparison between how the individual instances are classified and how this is reflected in the complete dataset. Five different machine learning algorithms namely J48 (Decision Tree), Naïve Bayes , K Star, IBK(KNN) and JRip were used in the experiment. WEKA 3.6.5 was used to get the classification results of the above mentioned algorithms. These classification methods are used to classify samples from different datasets. Then, an analysis of the classification results was done in order to identify how different pairs of learning methods differ from each other and which of the two algorithms perform better.

Motivation

Evaluation is necessary in order to compare different algorithms or classifiers, but which evaluation method should be used? Several evaluation methods of different origin have been proposed, e.g., from statistics, mathematics, and artificial intelligence. Which evaluation method to use is still an open research question, because evaluating a classifier often depends on the difficult task of measuring generalization performance, i.e., the performance on new, previously unseen data? For most real-world problems we can only estimate the generalization performance. Thus, it is important to continue investigating the existing evaluation methods theoretically and empirically in order to determine the characteristics of different estimations and when a certain method should be used. The idea to look at evaluation methods for classifier performance came from reading a theoretical paper on the topic of measure-based evaluation. However, the practical use of this particular method has not been investigated. To evaluate learning algorithms, we usually evaluate some of the classifiers it can generate. Methods that divide the available data into training and testing partitions a number of times (to generate classifiers from several training sets) have been investigated in many papers. However, learning algorithms like many computer programs can usually be configurated and we argue that it could be important to look at how well an algorithm can be tuned for a specific problem, and not only how well it performs on a particular problem. The fundamental question that we study is how to measure the performance of supervised learning algorithms and classifiers. Hence, the scope is narrowed from general learning problems to only include learning problems for which:

- The input consists of a sample of instances denied by a set of attributes,
- The correct output for each instance of the sample is available

Chapter 1

Decision Tree [13]:

- A decision tree is a special case of a state-space graph.
- It is a tree in which each internal node corresponds to a decision, with a sub tree at these nodes for each possible outcome of the decision.
- Decision trees can be used to model problems in which a series of decisions leads to a solution.
- The possible solutions of the problem correspond to the paths from the root to the leaves of the decision tree.
- As the name implies, this technique recursively separates observations in branches to construct a tree.
- Most decision tree classifiers perform classification in two phases: tree-growing (or building) and tree-pruning.
- The tree building is done in top-down manner.
- During this phase the tree is recursively partitioned till all the data items belong to the same class label.
- In the tree pruning phase the full grown tree is cut back to prevent over fitting and improve the accuracy of the tree in bottom up fashion.
- It is used to improve the prediction and classification accuracy of the algorithm by minimizing the over-fitting. Compared to other data mining techniques, it is widely applied in various areas since it is robust to data scales or distributions

Naive Bayes[13]:

- Naive Bayes (NB) classifier is a probabilistic classifier based on the Bayes theorem.
- Rather than predictions, the Naïve Bayes classifier produces probability estimates.
- For each class value they estimate the probability that a given instance belongs to that class.
- Requiring a small amount of training data to estimate the parameters necessary for classification is the advantage of the Naive Bayes classifier.
- It assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence.

Product rule $P(A \wedge B) = P(A \mid B)P(B) = P(B \mid A)P(A)$

Sum rule $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

Bayes theorem $P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$

Theorem of total probability, if event *Ai* is mutually exclusive and probability sum to 1.

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i)P(A_i)$$

Given a hypothesis *h* and data *D* which bears on the hypothesis:

P(h): independent probability of *h*: *prior probability*

P(D): independent probability of *D*

P(D|h): conditional probability of *D* given h: *likelihood*

P(h|D): conditional probability of *h* given *D*: *posterior probability*

IBK or K-Nearest Neighbor (KNN) [13]:

- IBK or K-Nearest Neighbor classification classifies instances based on their similarity.
- It is one of the most popular algorithms for pattern recognition.
- It is a type of Lazy learning where the function is only approximated locally and all computation is deferred until classification.
- An object is classified by a majority of its neighbors. K is always a positive integer.
- The neighbors are selected from a set of objects for which the correct classification is known. In WEKA this classifier is called IBK.
- The k-NN algorithm for continuous-valued target functions
- Calculate the mean values of the *k* nearest neighbors
- Distance-weighted nearest neighbor algorithm
- Weight the contribution of each of the k neighbors according to their distance to the query point x_q giving greater weight to closer neighbors.
- Similarly, for real-valued target functions.
- Robust to noisy data by averaging k-nearest neighbors.
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes.
- To overcome it, axes stretch or elimination of the least relevant attributes.

J-Rip [13]:

The algorithm is briefly described as follows:

Initialize RS = {}, and for each class from the less prevalent one to the more frequent one, DO:

1. Building stage:
Repeat 1.1 and 1.2 until the discretion length (DL) of the rule set and examples is 64 bits greater than the smallest DL met so far, or there are no positive examples, or the error rate >= 50%.

1.1. Grow phase:
Grow one rule by greedily adding antecedents (or conditions) to the rule until the rule is perfect (i.e. 100% accurate). The procedure tries every possible value of each attribute and selects the condition with highest information gain: p(log(p/t)-log(P/T)).

1.2. Prune phase:
Incrementally prune each rule and allow the pruning of any final sequences of the antecedents; The pruning metric is (p-n)/(p+n) – but it's actually 2p/(p+n) -1, so in this implementation we simply use p/(p+n) (actually (p+1)/(p+n+2), thus if p+n is 0, it's 0.5).

2. Optimization stage:
after generating the initial rule set {Ri}, generate and prune two variants of each rule Ri from randomized data using procedure 1.1 and 1.2. But one variant is generated from an empty rule while the other is generated by greedily adding antecedents to the original rule. Moreover, the pruning metric used here is (TP+TN)/(P+N).Then the smallest possible DL for each variant and the original rule is computed. The variant with the minimal DL is selected as the final representative of Ri in the rule set. After all the rules in {Ri} have been examined and if there are still residual positives, more rules are generated based on the residual positives using Building Stage again.
3. Delete the rules from the rule set that would increase the DL of the whole rule set if it were in it. and add resultant rule set to RS.
ENDDO

Chapter 2

DATASETS

In order to perform a completely correct evaluation, a comparatively very large number of datasets were obtained from UCI Machine Learning Repository [8] and used. Different datasets were randomly selected having both, numeric data (Table 2) and nominal data (Table 1).

Table1: Nominal Dataset

Dataset	**Instances**	**Attributes**	**Class Labels**
Arrhythmia	452	279	2
Restaurant/Consumer	138	47	3
Cylinder Bands	512	39	5
Zoo	101	18	7

Table 2: Numeric Dataset

Dataset	**Instances**	**Attributes**	**Class Labels**
Abscisic Acid Signal	300	43	2
Australia Sign language	2565	22	7
Concrete Slump Testing	103	10	3
Communities /Crime	1994	128	4
Ecoli bacteria	336	8	3

Chapter 3

Mc NEMAR'S TEST

The Mc Nemar's test [12] is a different version of c2 test and is a non-parametric test used in analyzing matched pairs of data. According to Mc Nemar's test, two algorithms can have four possible outcomes which can be arranged in a 2×2 contingency matrix as given in Table 3.

Table 3: Contingency Table

	Algorithm A is failed	Algorithm A is succeeded
Algorithm B is failed	Nff	Nsf
Algorithm B is succeeded	Nfs	Nss

Nss denotes the number of times (instances) when both the algorithms succeeded and Nff denotes failure of both the algorithms. These two cases does not give much information about the given algorithm's performanc as these do not indicate how the algorithm's performance differ from the other. However, the other two parameters (Nsf and Nfs) show the cases where one of the given algorithm fails and the other succeeds indicating the performance difference. In order to quantify these discrepancies, Mc Nemar's test is employed by calculating Z score.

$$z = \frac{(|N_{sf} - N_{fs}| - 1)}{\sqrt{N_{sf} + N_{fs}}}$$

Z- scores are calculated as given below:

- When z = 0, then the two algorithms are assumed to show same or similar performance.
- When this value gets diverged from 0 in positive direction, it is an indication that their performance differs from each other significantly.
- z scores can also be transformed into confidence levels as given in Table 4.

Table 4:

Confidence levels conferring to z scores for one-tailed prediction and two-tailed prediction

Z Score	One-tailed Prediction	Two-tailed Prediction
1.745	94.5%	94%
1.980	97%	95%
2.236	98.5%	98%
2.756	99%	99%

Following the table, it should be known that One-tailed Prediction is used to determine when an algorithm is better than the other algorithm whereas Two-tailed Prediction gives how much the two algorithms are different.

Mc Nemar's test is known to have a low Type-1 error which occurs when an evaluation method finds a difference between two machine learning algorithms when there are no differences at all.

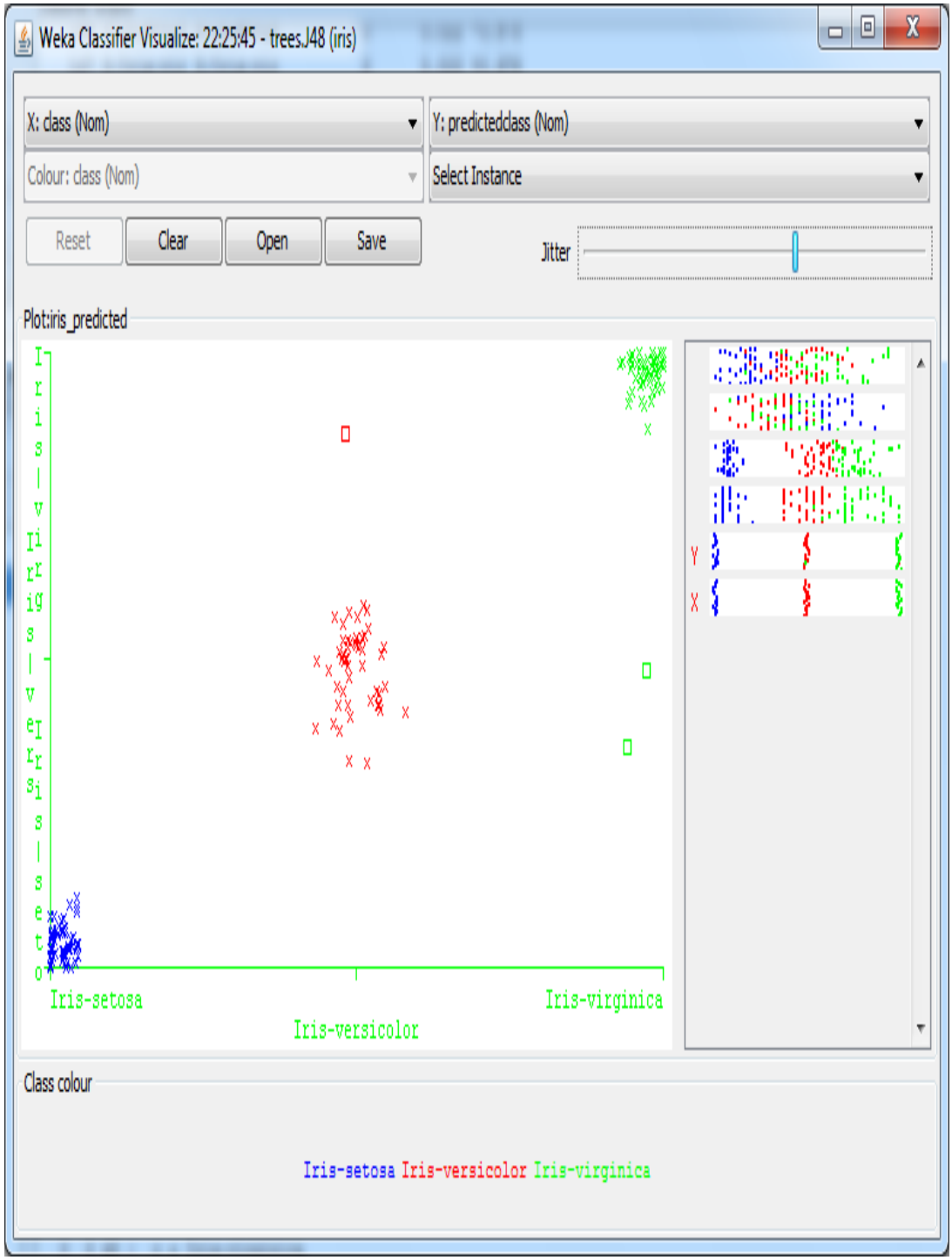
EVALUATION CRITERION

By applying the Mc Nemar's test for the evaluation of classification techniques, the following method is defined: An algorithm is said to be successful, if it can identify the class of an instance or variable correctly. Conversely, it is regarded as failed if it gives an incorrect class label for an instance. Using this criterion, using Mc Nemar's test, the z- scores were calculated for the 5 classification algorithms. All the above algorithms were used with their default parameter as parameter tuning may favor one or other algorithm to give better result. The null hypothesis (H0) for this experiment design says that different classifying techniques perform similarly whereas the alternative hypothesis (H1), otherwise suggests that at least one of the classifying techniques perform differently as shown in the following Equations.

H0: C1 =C2 =C3 =C4 =C5
H1: 9Ci, Ci 6=Cj, (i, j) 2 (1, 2, 3, 4, 5), i 6= j (2)

At the end of the experiment, the z scores indicate whether we should accept H1 and reject H0 or vice versa. In order to calculate the z scores, the classification results of the above classifiers must be identified for each individual data instance. This operation is performed for all instances in the obtained datasets. In WEKA 3.6.5, there are two options to see whether an instance is correctly classified or not. First option is the graphical one. The second option is to show the incorrect classification via the Output predictions option of the classifier. 10-fold cross-validation is used in the evaluation which works as: First the data is separated into ten sets of n/10 instances each. Then the training is performed using nine of these sets as training sets and one as testing set. This process is repeated ten times to

consider all the possible subsets divided and the final result for the class label is obtained by taking the average of all these iterations .The first option is used to see the result in a graphical manner, however, to calculate the number of incorrect and correct classifications by the method, one needs to feed these results into an Excel spreadsheet. For this reason, the second method is used to calculate number of instances where the classifiers succeeded or failed. Using these values, the z score was calculated using Equation 1. In order to decide which classifier performed better, Nfs and Nsf values for 2 classifiers were examined. For example, classifier X is said to perform better than classifier Y if Nsf is larger than Nfs according to Table 3 and vice versa.


Weka Classifier Visualize: 22:25:45 - trees.J48 (iris)
X: class (Nom)
Y: predictedclass (Nom)
Colour: class (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot:iris_predicted
Iris-setosa
Iris-versicolor
Iris-virginica
Class colour
Iris-setosa Iris-versicolor Iris-virginica

Chapter 4

McNemar's Tests' Results

In Tables 5 and Table 6, the arrows denote which classifier performed better in the particular datasets. z scores are given next to the arrows as a measure of how significant the results are. By looking at the Mc Nemar's test results for the nominal datasets (Table 5), one can arrive at the conclusion that K nearest Neighbor has produced significantly better results than J48 and Naive Bayes classifiers (H1 is accepted with a confidence level of more than 99.5%). J48 classifier performed better than the Naive Bayes for Abscisic Acid Signal and Ecoli bacteria datasets. For the Zoo dataset, Naive Bayes performed better than J48 and equally to the K Nearest Neighbor (H0 is not rejected.). The performance differences between IBK (K NN) and all other classifiers were not found to be statistically significant for the Zoo dataset but for the other nominal datasets, there were noticeable differences. JRip shows a poor performance overall except for the Zoo dataset where it performed better than J48

Table 5 Mc Nemar's Test Results for Nominal Datasets

	Naive Bayes	IBK(K NN)	J Rip	K Star
J 48	0	9.63	1.72	6.91
Naïve Bayes		9.63	1.72	6.91
IBK(K NN)			9.72	14.08
JRip				9.85

	Naive Bayes	IBK(K NN)	J Rip	K Star
J 48	25.56	17.22	34.79	24.62
Naïve Bayes		34.79	32.68	0
IBK(K NN)			12.08	34.86
JRip				31.73

	Naive Bayes	IBK(K NN)	J Rip	K Star
J 48	8.45	10.04	15.63	8.57
Naïve Bayes		15.63	0.65	0.65
IBK(K NN)			15.80	15.80
JRip				

	Naive Bayes	IBK(K NN)	J Rip	K Star
J 48	0.66	1.33	0	0.6
Naïve Bayes		0	0	0
IBK(K NN)			0	0
JRip				0

although the results were not significant. Many discrepancies in the classification are noticeable in the numeric datasets (Table 6). For the Abscisic Acid Signal and Ecoli bacteria datasets J48 has given better classification results than Naive Bayes. For the former dataset, the K Nearest Neighbor performed equally with J48 and Naive Bayes gave a poor classification performance than the other two. KNN and Naïve Bayes show better performance over J48; however there was no statistically significant

performance difference between these two classification methods. It is equally interesting to see that first 3 (J48, Naive Bayes, K NN) classifiers performed similarly on the Concrete slump dataset (H0 is not rejected). Some differences can be noticed between these classifiers, however the results are not significant (z = 0.75 for Naive Bayes and K NN) A similar result is also visible when the Arrhythmia dataset is considered because the values are very close to zero. For the Arrhythmia dataset, Naive Bayes showed better performance over J48 yet the difference was not very significant (with a confidence level less than 95%) whereas Naive Bayes performs significantly better than J48 for the Restaurant/Consumer dataset. We can also see that the IBK (K NN) did not produce good results for the Communities and crime dataset where a relatively large number of attributes are present. This lower performance can be due to an under fitting problem as the default parameter is used without any parameter tuning.

Table 6 McNemar's Test Results for Numeric Datasets

	Naive Bayes	IBK (K NN)	J Rip	K Star
J 48	1.63	0.98	0.54	0.24
Naïve Bayes		0.54	3.30	1.39
IBK(K NN)			2.94	0.56
JRip				2.18

	Naive Bayes	IBK (K NN)	J Rip	K Star
J 48	4.03	0	4.03	0.94
Naïve Bayes		4.03	5.08	5.22
IBK(K NN)			0.94	0.94
JRip				0

	Naive Bayes	IBK (K NN)	J Rip	K Star
J 48	0	0	0	1.04
Naïve Bayes		0	2.63	0.78
IBK(K NN)			2.63	0.78
JRip				1.36

	Naive Bayes	IBK (K NN)	J Rip	K Star
J 48	0	0.43	0.5	1.53
Naïve Bayes		0.5	0	1.53
IBK(K NN)			1.14	2.00
JRip				1.21

	Naive Bayes	IBK (K NN)	J Rip	K Star
J 48	12.93	1.66	14.27	6.54
Naïve Bayes		14.27	13.37	8.52
IBK(K NN)			0.89	7.96
JRip				7.04

CONCLUSION

This study employed Mc Nemar's test in order to evaluate machine learning algorithms namely J48, Naive Bayes and IBK (K NN) and JRip . By defining the failure and success criteria of Mc Nemar's test as correctly or incorrectly identifying the class label of an instance in a given dataset, the experiments presented the use of a non-parametric test as a new method to evaluate classification algorithms. The results showed that IBK (K NN) produced better results than the other methods for both nominal and numerical datasets. JRip was placed in the lowest ranks for both types of data. Another interesting result of the experiment was that the results of the Mc Nemar's test mostly conferred with Kappa statistic and Root Mean Squared Error as a justification of method's accuracy. The effect of parameter tuning is considered as a future research. In this case, the classifiers will be particularly tuned to achieve the optimally maximum results and then the same tests can will applied to see whether there will be any changes in the rankings.

References

1. T. G. Dietterch, Approximation of statistical test for comparison of classification learning techniques.

2. R. Bouckart and E. Franki, evaluating the significance test for comparison of classification techniques, in 8th Pacific-Asia Conference.

3. Demsar, Statistical comparisons of classification techniques on different datasets, Journal of Machine Learning 1 June, 2006.

4. M. Friedman, Comparison of alternative techniques for the problem of n rankings, The Annals of Mathematical Statistics, 1940.

5. P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Pearson Publications, 2009.

6. Friedman, Geiger-Goldszmit-Bayesian classifiers, Machine Learning, vol. 29, pp. 132–162.

7. M. Hall, E. Frank, G. Holmes, and I. H. Witten, The WEKA data mining software: An update, vol. 11, 2010.

8. "UCI Machine Learning Repository." http://archive.ics.uci.edu/ml/, 2013.

12. Q.McNemar, "Note of the sampling error in the difference between related proportion and percentage," Psychometrika.

13. Data Mining Concepts and Techniques- 2nd Edition by Jiawei Han and Micheline Kamber