ISSN 2347 - 3983

Volume 10. No.6, June 2022

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter081062022.pdf

https://doi.org/10.30534/ijeter/2022/081062022

Comparing Supervised Machine Learning Algorithms on Classification Efficiency of multiclass classifications problem

Workineh Menna Eligo[1], Chengcai Leng[2], Aklilu Elias Kurika[3], Anup Basu[4]

[1] School of Mathematics, Northwest University, Xi'an 710127, China
worke2003@gmail.com
[2]School of Mathematics, Northwest University, Xi'an, 710127, China
ccleng@nwu.edu.cn
[3]Department of Information Technology, Wolaita Sodo University, Ethiopia
akliluelias123@gmail.com
[4] Department of Computing Science, University of Alberta, Edmonton, AB T6G 2H1, Canada
basu@ualberta.ca

Received Date: March 05, 2022 Accepted Date : May 30, 2022 Published Date : June 07, 2022

ABSTRACT

Multi-class classification is a fascinating field to study. However, evaluating the classification performance of classifiers is difficult. Class indices such as accuracy, precision, recall, and F-measure, Kappa and area under the curve of receiver operating characteristics (AUC), can be used to evaluate classification performance. These indices describe the classification results achieved on each modelled class. Several measures have been introduced in the literature to deal with this assessment, the most commonly used being accuracy. In general these metrics were proposed to address binary classification tasks, whereas multiclass classification is the more difficult and currently active research area in machine learning (ML). In this paper, we intended to compare classification performance of nine supervised machine learning algorithms based on three learner types: statistical learner, rule-based learner and neural-base learner by considering accuracy, precision, recall and F-measure and ROC area achieved on four different datasets from UCI machine repository. Among these, Random forest has been the best performance in both 10 fold cross validation and percentage split with overall average accuracy of predictive power of 92.20% and 91.76% respectively, with less variability, whereas Naïve Bayes has the worst also in both 10 fold cross validation and percentage split by average correct classification performance of 79.18% and 76.92% respectively, and also with higher variability next to Decision Table.

Key words: Classifier performance, Multiclass classification, neural learner, rule based learner, statistical learner, UCI repository dataset

1. INTRODUCTION

Currently, a large amount of information is widely available in different formats on various media channels. Machine learning, a branch of Artificial Intelligence (AI), is the most recent and powerful data mining and representation technique. Artificial Intelligence attempts to upgrade computer programs to achieve tasks that usually need human involvement, like decision-making. Having the right decision for a specific problem is a key factor for achieving what we need. Many machine learning techniques are used for both classification and regression problems. Classification is used when the prediction goal is a discrete value or a class label. When the prediction goal is continuous, regression is the appropriate method [1].

Classification plays an important role in areas of gene selection [2], image classification, medical diagnosis [3-4], economic analysis, risk analysis [5-6], bioinformatics analysis [7] and many others [8]. There are only a few extensive empirical studies comparing classification performance of learning algorithms. Some of these studies are [7 and 9-12] which was mainly focused on binary response variables.

Multiclass-classification is the emerging research area that a problems arise in a condition where there are more than two levels in the response variable [13]. Currently, there are many multi class-classification algorithms. Some of these approaches are multinomial logistic regression (MLR), Naïve Bayesian methods, neural networks, k-Nearest Neighbors, random forest, decision trees, and hierarchical classification schemes [14]. MLR is simple to implement, and is very effective in handling multi-classification problems in the modern era [15]. Support vector machine (SVM) were originally developed for binary classification [16-17]. However, currently many researches are trying to use SVM to solve multiclass classification problems [18].

In this paper, we focus on comparing the classification performance of the training dataset and the predictive power of the unseen dataset of different classifiers such as: J48 (tree based), Random Forest (tree based), Multilayer Perceptron (MLP) is a class of feed forward artificial neural network (ANN), IBK (k-nearest neighbor), sequential minimal optimization (SMO) works as of support vector machines, Naïve Bayes, PART, Decision Table, and Logistic (it works by using Multinomial logistic regression model with ridged estimator) were compared in different set of problems, and the second objective of our research is distinguishing which classifier is best to which type of dataset. To evaluate the classification performance and prediction power of the classifier we used different metrics such as: accuracy, weighted Sensitivities/Recall, weighted Specificities, weighted Precision, weighted F-Measures, weighted area under ROC and kappa statistic for 6 dataset from UCI machine learning repository using R statistical software and WEKA data mining tool.

The computational complexity considers both the model train time as well as the test set evaluation time, rather than placing emphasis on only one of these, since some of the algorithms need more time to classify the test set than training the model. The machine configuration was Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz and 4 GB RAM.

2. THE THEORETICAL BACKGROUND OF LEARNING ALGORITHMS AND THEIR PROCEDURE

This section briefly describes all the algorithms that we considered in the experimental design and their procedure. The algorithms are belonging to the category of supervised learning methods, but we classify them into statistical learning, rule-based and neural algorithms, as described in section below.

In this paper the researcher has been followed the analysis in three different categories; named *Statistical learner, Rule-based learner and Neural-based learner*

2.1 Statistical Learner

Statistical learning theory is based on the machine learning framework in the field of statistics and functional analysis [19-20] statistical learning theory deals with the problem of finding predictive functions based on data. Statistical learning theory has been successfully applied in the fields of computer vision, speech recognition and bioinformatics. There are some statistical learning algorithms working for multiclass classification such as: support vector machine, multinomial logistic regression, multilayer perceptron and linear discriminant analysis are few of them.

2.1.1 Support Vector Machine (SVM)

Recently, after Vapnik *et al.* introduced SVM in the mid-1990s, statistical learning theory has received more attention from the pattern recognition community. SVM is an advanced version of the generalized portrait algorithm, which was developed in Russia in the late 1960s [21]. The working principle of SVM is similar to NN and C4.5. We can assign three work phases to the SVM; the first is the input phase or the conversion phase, then the learning phase, and finally the decision-making phase. NN and C4.5 did not do any important work in the first stage. But SVM has completed its most important work, transforming data by mapping the kernel to a high-dimensional feature space. The kernel function can be a polynomial, a Gaussian function, or many other functions. Theoretically, high-dimensional space can be infinite and linear discrimination is almost possible. SVM begins to learn data in high-dimensional feature space. In the learning stage, it is freed up by minimizing the size of the separation-constrained weight vector (based on the optimal hyperplane) and by using multiplier parameters (such as Lagrange multipliers) problem. At this stage, SVM only extracts support vectors. Based on the information in the support vector, SVM generates the final output function at the decision-making stage. Unlike NN and C4.5, SVM does not consider all samples to construct the final decision function with. Also, unlike iterative methods or pruning, SVM always gets a unique solution for the function of decision. Another characteristic of SVM is that it minimizes the structural risk of, rather than the empirical risk considered by most classical learning algorithms of [22-24]. WEKA considers SVM and minimum sequential optimization (SMO) of the polynomial kernel as 1 degree as the default configuration [25]. Naive Bayes (NB) is a simple classifier based on the classical statistical theory "Bayes theorem", it calculates maximum posterior probabilities based on the assumption that the attributes in training samples are independent and there are no hidden or latent attributes in the prediction process [26]. IBK is an instance-based learning approach like the k-nearest neighbor method. The basic principle of this algorithm is that each invisible instance is always compared with the existing instance using the distance metric; the most common Euclidean distance and the nearest existing instance are used to assign classes to the test samples [25]. The default setting of WEKA is k = 1. Compared with other algorithms, it takes longer to predict the category of test samples.

Support vector machine, the statistical learning algorithm has some advantages over the decision tree and NN algorithm. SVM takes the dot product of feature vectors to construct the optimal hyper-plane instead of using surfaces, clusters or interpolation as NN or decision trees. Therefore, it is less likely to lose important information during the modeling process [27].

2.1.2 Support Vector Machine (SVM) Procedure for Multiclass Classifications

SVMs were originated to perform binary classification [28]. However, applications of binary classification are very limited especially for more than two classes like remote sensing, land use land cover classification and so on. A number of methods proposed by researchers to generate multiclass SVMs from binary SVMs are still a continuing research topic. There are two main approaches of SVMs for multiclass classification - one against one and one against all approach [29]. The researcher used pairwise or one – vs – one approach to handle multi-classification problem.

Alternatively, generating many binary classifiers to decide the class labels is a method that attempts to directly solve a multiclass problem [30-33]. This is attained by adapting the binary class objective function and adding a constraint to every class. The modified objective function allows simultaneous computation of multiclass classification as given by [31],

$$\min_{w,b,\xi}\left[\frac{1}{2}\sum_{i=1}^{M}\|w\|^2+C\sum_{i=1}^{k}\sum_{r\neq y_i}\xi_i^r\right] \tag{18}$$

subject to the constraints,

$W_{y_i}\cdot X_i + b_{y_i} \geq W_r\cdot X_i + b_r + 2 - \xi_i^r$ for and, $\xi_i^r \geq 0$

for $i = 1,\dots,k$

where y_i ∈{1, . . ., M} are the multiclass labels of the data vectors and r ∈{1, . . ., M}\ y_i are multiclass labels excluding y_i.

As [30, 32, 33] mentioned, the results from this method and the one-against -all are similar. But, in this method, the optimization algorithm has to consider all the support vectors at the same time. Therefore, it may be able to handle large data sets; but, the memory and resulting computational time may be very high. In general, it can be said that the choice of a multiclass method depends on the problem in hand. A user should consider the accuracy requirement, computational time, resources available and nature of the task. For example, the multiclass objective function approach may not be suitable for a problem that contains a large number of training samples and classes due to the requirement of large memory and extremely long computational time.

2.1.3 Multinomial Logistic Regression Model

Multinomial logistic regression is used to predict categorical placement in or the probability of category membership on a dependent variable. The independent variables can be dichotomous (i.e., binary) or continuous (i.e., interval or ratio in scale) [34]. It is a binary logistic regression extension that includes native support for multiclass classification issues. By default, logistic regression is limited to two-class classification tasks. Some extensions, such as one-vs-rest, can be used to solve multi-class classification issues with logistic regression, but they require the classification problem to be split into many binary classification problems first. The multinomial logistic regression algorithm, on the other hand, is an extension of the logistic regression model that involves changing the loss function to cross-entropy loss and the predict probability distribution to a multinomial probability distribution to support multi-class classification problems natively [35]. Compared to logistic regression, it is more general since the response variable is not restricted to only two categories. As [36] its name is logistic in WEKA to build a model it uses multinomial logistic regression model with a ridge estimator.

2.2 Rule-Based Learner

There are many Rule-based learning algorithms, one of them is decision trees, it is also known as classification trees or hierarchical classifiers, is a divide and conquer or top-down induction method, there are many research have been done about decision tree in the Machine learning community. C4.5 was eventually extended to J48, an open source Java implementation of the C4.5 algorithm based on WEKA, which was an ID3 extension. J48 also considers missing values, decision tree pruning, continuous attribute value range, rule derivation, and other functions. The WEKA tool provides a variety of tree-pruning choices. The trim can be utilized as a precise tool in the event of suspected over-fitting. Other algorithms repeat the classification process until each sheet is pure, which means the data classification must be as accurate as possible. The algorithm develops rules, and the data's unique identity is determined by those rules [37].

OneR is a very simple and fast single-level decision tree algorithm [37]. OneR is selects attribute one by one from the data set and generates a different set of rules based on the error rate of the training set. Ultimately, it chooses the attributes with the smallest error rule and builds the final decision tree [38]. PART is a partial decision tree algorithm, which is the development version of the C4.5 and RIPPER algorithm. The main characteristic of the PART algorithm is that it does not require global optimization like C4.5 and RIPPER to generate adequate rules [39]. However, decision trees are sometimes more problematic because the size of the tree may be too large and may not work well for classification problems [40].

2.2.1 Random Forest

Breiman was the first to introduce Random Forests (Rf) [41] motivated by the previous work of Amit and Geman [42]. As Breiman mentioned, Random Forests can be used for either a categorical response variable for “classification,” or a continuous response, referred to as “regression.” Similarly, the predictor variables can be either categorical or continuous. From a computational point of view, Random Forests are appealing because they naturally handle both regression and (multiclass) classification. They are relatively fast to train and use for prediction, depend only on one or two tuning parameters, have a built in estimate of the generalization error, can be used directly for high-dimensional problems and can be easily implemented in parallel. Statistically, Random Forests are appealing because of the additional features they provide, such as: measures of variable importance, differential class weighting, missing value insertion, visualization, outlier detection and unsupervised learning.

In machine learning community, random forest is popular and it can be used in ecological classification [43], land-cover land usage [44], and medical data analysis [45], [46]. As Breiman

explanation random forest is an ensemble of tree structured classifiers. Each tree of the forest gives a unit vote, assigning each input to the most probable class label. It is a fast method, robust to noise and is a successful ensemble which can identify non-linear patterns in the data. It can easily handle both numerical and categorical data [46]. One of the major advantages of a random forest is that it does not suffer from over fitting, even if more trees are appended to the forest.
In this paper, we focus on comparing the classification performance of the random forest model and other selected models.

2.3 Neural-Based Learner

During the 1960s, Nilsson introduced pattern recognition artificial intelligence based on Neural, as a threshold unit called a neural network (NN). Neural networks have become a method after the development of new algorithms, such as the multilayer perceptron (MLP), the radial basis function network, SOM, and BP. The MLP architecture consists of three layers of neurons, namely the input layer, the hidden layer, and the output layer, all connected by feedback weights. After receiving the input pattern, the NN passes the signal through the network to predict the output of the output layer. The NN then compares the predicted target value with the actual target and estimates the error to modify the weight. Minimize the scalar error function of the weights by repeating the learning process until the network produces the correct response to each input [46 - 47]. WEKA uses BP algorithm to train the model. BP uses gradient descent to minimize the error function. The main disadvantage of the BP algorithm is that it is slower than some other popular machine learning techniques, and it is easy to fall into the local minimum of the error function [48].

3. MATERIALS AND METHODS

3.1 Data and Experiment

Table 1: The details of datasets from UCI machine learning repository

S/N	Dataset	No. of Instance	No. of Attribute	No. of Class	Type of Attribute
1.	Page Blocks	5,473	11	5	Integer, Real
2.	Dry_bean	13,611	17	7	Integer, Real
3.	Letter Recognition	20,000	17	26	Integer
4.	Connect-4	67,557	43	3	Categorical

In this paper we have used four multi-classification problem dataset. All the dataset in Table 1 has been taken from UCI machine learning repository [49], we can find the detail description of attributes and instance from the respective site.

3.2 Evaluation of Classifiers Methodology

3.2.1 Weighted Performance Measure

There are various measures for evaluating the classifier performance of classification problem. No single measure can give us the whole story about the classifier performance. We used the most common two measures, accuracy and computational time on training dataset and test dataset for classifier performance evaluation. First, we use weighted TPR, weighted TNR, weighted precision, weighted recall, weighted ROC curve area and weighted F-measure, as suggested by [48] is help to minimize the impact of the imbalanced dataset. Secondly, we measure performance through a 10-fold cross-validation, since it is powerful to measure due to different reasons [48]. Finally, we try to minimize the impact of the imbalance between the distribution of the minority and the majority class by using the weighted F- measurement method, F-measure considers the weighted distribution of the data set as of [50] and then computing the weighted average area under the receiver operating characteristic (AUC-ROC). Each of the methods was trained (estimated) and tested using the 10-fold cross validation procedure and holdout methods as mentioned above, so that the results could be compared by using the same subsets of data for training and testing.

The total classification rate (i.e. the proportion of correctly classified cases in the test set) is used to measure the performance of all models on each of the test samples, and a 10-fold cross-validation procedure was used to test the models' generalization ability. In this paper, the cross-validation procedure (or leave k cases out, where k =1/10 of the total sample) is used, because it produces no statistical bias of the result because each tested sample is not a member of the training set [48]. Extensive tests on numerous datasets, according to [51], have shown that 10 is a sufficient value for n in the n -fold cross validation. Following the 10-fold cross-validation procedure, the average of the total classification rate is computed and used to estimate a model's generalization error. Also, the classification rates of each class were observed in order to compute the sensitivity and specificity of the models. The sensitivity and specificity ratios were computed according to Simon and Boring [52].

$$sensitiv\ ity = \frac{p_1}{p_1+d_0}, \qquad specificity = \frac{p_0}{p_0+d_1}$$

where p_0 is the number of instance correctly predicted to have output class-1, p_1 is the number of instances accurately predicted to have output other than class-1, d_0 is the number of false negatives (the number of instances falsely predicted to have output of class-1), and d_1 is the number of false positives (the number of instances falsely predicted to have output other than class-1). The type I error (α *=1-specificity)* and type II error (β *=1-sensitivity)* were calculated in order to compare the cost of misclassification produced by each of the models, whereas the likelihood for positives and likelihood for negatives in classification is computed according to:

$$L_1 = \frac{Sensitivity}{\alpha}, \qquad L_0 = \frac{Specificity}{\beta}$$

where L_1 is likelihood for being class-1,while L_0 is the likelihood for not being class-1, this is directly working for Adult and Mushroom dataset whereas for Connect-4 dataset L_1 is being class-1 and L_0 is not being class-1 or in other word the likelihood being class-2 or class-3, and so for other datasets.

Some additional performance evaluations of classifiers considered in this paper are:

i) Accuracy

The accuracy of an algorithm is the measure of how correctly the algorithm classifies the unseen instances. It can be computed by the following formula:

$$\frac{Numbers\ of\ correctly\ classified\ instances}{Total\ number\ of\ instances} * 100$$

ii) Confusion matrix

Accuracy is not the only way of evaluating the performance. Occasionally, we may need a more detailed picture of the performance of the classifier. One approach is a table called Confusion Matrix as shown in the following table [50].

Table 2: Confusion matrix for two classes

Predicted Class			
Actual Class	Positive Rate	Negative Rate	Total
Positive	True Positive Rate (TPR)	False Negative Rate (FNR)	p
Negative	False Positive Rate (FPR)	True Negative Rate (TNR)	n
Total	p’	n’	N

The following rules can be extracted from the table above:

$$\text{Sensitivity = recall= TPR} = \frac{TP}{p} = \frac{TP}{TP+FN} = 1 - \text{ FNR}$$

$$\text{Specificity = TNR} = \frac{TN}{N}\ \frac{TN}{FP+TN} = 1 - \text{ FPR}$$

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

$$\text{Error rate = 1- Accuracy} = \frac{FN+FP}{TN+FN+FP+TP}$$

According to Visa, Sofia, et al, Sensitivity is the ratio of the positive examples that are correctly classified, whereas Specificity is the ratio of the negative examples that are incorrectly classified. Higher is better for accuracy, specificity and sensitivity. But, for the error rate, lower is better. A good classifier should be sensitive and specific with a higher degree.

Table 3: Confusion matrix for multi classes (say n - class)

		Actual number			
		Class 1	Class 1	...	Class n
Predicted Number	Class 1	x_{11}	x_{12}	...	x_{1n}
	Class 2	x_{21}	x_{22}	...	x_{2n}
	⋮	⋮	⋮	⋮	⋮
	Class n	x_{n1}	x_{n2}	...	x_{nn}

In this paper we used the following procedures to compute the weighted average precision (*P*), recall (*R*), and specificity (S_p) and Sensitivity (S_s) for each class i, since all of our dataset is multiclass type of data.

$$P_i = \frac{TTP_{all}}{TTP_{all} + TFP_i}$$

$$S_{si} = R_i = \frac{TTP_{all}}{TTP_{all} + TFN_i}$$

$$S_{pi} = \frac{TTN_{all}}{TTN_{all} + TFP_i}$$

$$\text{Overall accuracy} = \frac{TTP_{all}}{Total\ number\ of\ testing\ intances}$$

Where, the total numbers of false positive (*TFP*), false negative (*TFN*), and true negative (*TTN*) and true positive (TTP) for each class ***i*** will be calculated as follow:

$$TFP_i = \sum_{j=1}^{n} x_{ij}\ , where\ j \neq i$$

$$TFN_i = \sum_{j=1}^{n} x_{ji}\ , where\ j \neq i$$

$$TTN_i = \sum_{j=1}^{n}\sum_{k=1}^{n} x_{jk}\ , where\ j \neq i\ and\ k \neq i$$

$$TTP_{all} = \sum_{j=1}^{n} x_{jj}$$

Finally, we can formulate the hold-out and cross-validation estimation by following [47]. Suppose the unlabelled sample space is X, with corresponding labels Y. The labeled sample space $\chi = X \times Y$ and $S = \{x_1, x_2, . . ., x_n\}$ is a dataset, which consists of n labeled samples, where $x_i = \{u_i \in X, y_i \in Y\}$. The inducer (S, u) will denote the label assigned to an unlabelled sample u by the classifier built by the inducer on dataset S, i.e., $(S, u) = (S)(u)$. The cross-validation and hold-out estimation consider the dataset is independent and identically distributed and equal misclassification costs using a 0/1 loss function.

The cross-validation method [48] is estimates the correct classification of average percentage for all folds. The 10-fold cross-validation method is splits a dataset by 10-fold. Each fold contains 90% of the samples to construct a model and the remaining is used to evaluate the model performance. Lastly, we estimate the accuracy is the overall number of correct classification averaged across all 10-fold.

Suppose, S_i is the test set that contains sample $x_i = (u_i, y_i)$ and the cross-validation accuracy estimation is defined as:

$$A_{CVE} = \frac{1}{n}\sum_{(u_i,y_i)} \gamma(\varphi(S \backslash S_{(i)}, u_i), y_i)\ , where\ n\ is\ the\ number\ of\ folds.$$

The hold-out method [50] is called the test sample estimation method. The very common procedure is 70% samples for the training set and the remaining for test set. Let us consider a hold-out set S_h be a subset of S of size h, and let S_t be $S\backslash S_h$. Now the hold-out estimation (HOE) is defined as [50]:

$$A_{HOE} = \frac{1}{h} \sum_{(u_i y_i) \in S_h} \gamma(\varphi(S_t, u_i), y_i), where\ \gamma(i,j) = 1, if\ i = j\ and\ 0\ otherewise.$$

We follow [50] to find the combined performance of the cross-validation and hold-out accuracy estimation for all given problems of a given classifier ‘percentage of correct classification’ in our weighted performance measure methodology.

iii) Receiver operating characteristics (ROC) curve and AUC

In machine learning, performance measurement is an essential task. The area under a receiver operating characteristic (ROC) curve, abbreviated as AUC, is a single scalar value that measures the overall performance of a binary classifier [54]. We can use weighted AUC to check or visualize the performance of a multi-class classification problem. It is one of the most important evaluation metrics for checking any classification model’s performance. It is also written as Area under the Receiver Operating Characteristics (AUROC)**.** The AUC value is in the range [0.5–1.0], where the minimum value represents the performance of a random classifier and the maximum value corresponds to a perfect classifier (i.e., with a classification error rate equivalent to zero, or AUC = 1). AUC is a robust overall measure to evaluate the performance of score classifiers because its calculation relies on the complete ROC curve and thus involves all possible classification thresholds. ROC is a probability curve and AUC represents the degree or measure of separability. It indicates how much the model is capable of distinguishing between classes [55].

iv) Kappa Statistic

Cohen's kappa [56] was introduced as a consistency metric, which avoided the problem by adjusting the observed proportional consistency to, taking into account the amount of consistency expected by chance. For the last three decades, Kappa statistics have been mainly used in social sciences, biology and medical sciences [57]. However, in the context of expert systems, machine learning, and data mining communities, Cohen's Kappa has not received much attention as an accuracy measure. There are some research have been done in machine learning [58-60], in which the Cohens Kappa statistics are calculated as one of the measures of accuracy. Although it has some criticisms of the Kappa metric, it is statistically robust [61]. The Kappa Statistic measures the performance of a classifier compared to the classifier that makes predictions based only on random guessing. According to Viera and Garrett, the better the classifier, the closer the kappa statistic value is to one [62].

4. EXPERIMENTAL RESULTS OF REAL DATASET

4.1 Performance Analysis

Based on the analysis and observations from the Table 4, 10 fold cross validation is relatively better than percentage split. The average correct classification performance of Random Forest in both 10 fold cross validation and percentage split is 92.20% and 91.76% respectively; and followed by IBK.

Table 4: Percentages of correctly classified instance for 6 dataset on 10 fold cross validation and percentage split using nine classifiers

Dataset	J48		Random Forest		Multilayer Perceptron		SMO		IBK		Naïve Bayes		PART		Decision Table		Logistic	
	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split	10 Fold CV	% Split
Page block	96.88	97.44	97.50	97.62	96.13	96.41	92.93	92.08	96.02	95.68	90.85	82.16	97.00	97.02	95.63	95.43	96.46	96.95
Dry Bean	91.32	91.38	92.55	91.92	92.49	91.80	92.20	91.92	90.30	90.06	89.71	89.52	91.31	90.77	88.02	87.36	92.60	92.11
Letter recognition	87.92	86.4	96.37	95.67	82.21	82.15	82.44	81.73	95.96	95.20	64.01	64.22	89.02	87.40	64.90	63.32	77.43	76.98
Connect-4	80.90	80.58	82.36	81.83	81.88	81.77	75.86	75.76	80.90	80.25	72.14	71.77	79.27	78.26	75.27	75.02	75.75	75.72
Ave.	89.26	88.95	**92.20**	**91.76**	88.18	88.03	85.86	85.37	90.80	90.30	79.18	76.92	89.15	88.36	80.95	80.28	85.56	85.44

Whereas Naïve Bayes for both 10 fold cross validation and percentage split is the worst, 79.18% and 76.92% respectively. Moreover, we can explain each classifier performance with the respective dataset accordingly, we can deduced for Page Block dataset, the performance of percentage split is relatively better except SMO, IBK, Naïve Bayes and Decision Table. But in the case of Dry_Bean, Letter recognition and Connect-4 dataset 10 fold cross validation is relatively perform better than percentage split.

Both Table 5 and Fig.2 shows that Random Forest is relatively outperform for all dataset except Dry_Bean dataset. But Naïve Bayes has least performance. Logistic is the better for

Dry_Bean dataset whereas Decision Table is the worst classifier.

As we seen from Fig.1, the average value of 10 fold cross validation is relatively higher than the percentage split. Moreover that, Random forest is outperform over the other in both 10 fold cross validation and percentage split, whereas, Naïve Bayes has worst classification performance as compare as the rest eight classification algorithms.

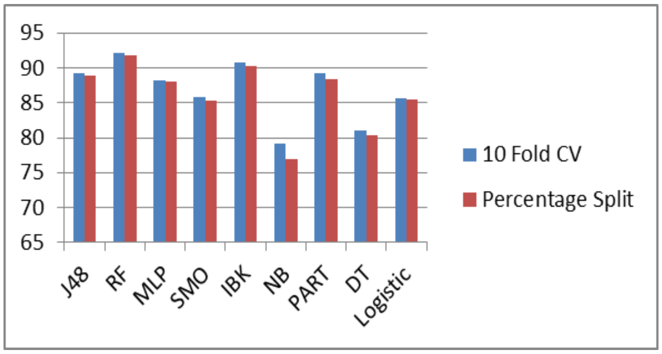


Figure 1: 10 Fold cross validation versus percentage split of average correct classification performance of instances for four dataset.

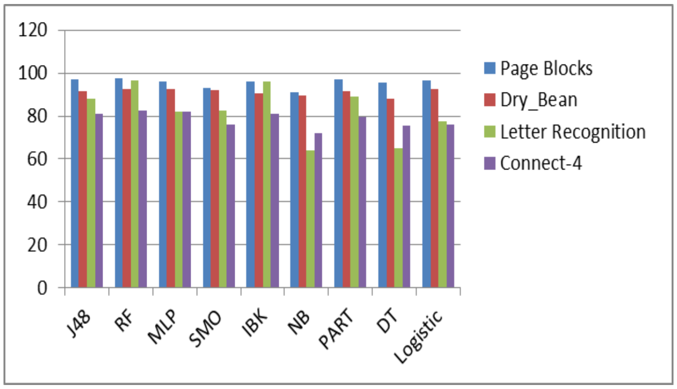


Figure 2: Correct classification for10 fold cross validation using four dataset

Table 5: Percentages of correctly classified instance of 10 fold cross validation using six dataset

Dataset	Types of Classifiers								
	J48	Random Forest	MLP	SMO	IBK	Naïve Bayes	PART	Decision Table	Logistic
Page Blocks	96.88	**97.50**	96.13	92.93	96.02	90.85	97.00	95.63	96.46
Dry_Bean	91.32	92.55	92.49	92.20	90.30	89.71	91.31	88.02	**92.60**
Letter Recognition	87.92	**96.37**	82.21	82.44	95.96	64.01	89.02	64.90	77.43
Connect-4	80.90	**82.36**	81.88	75.86	80.90	72.14	79.27	75.27	75.75

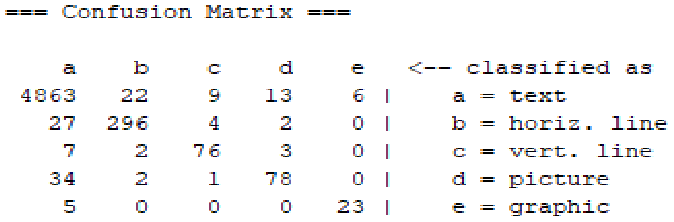
As we seen from Table 6, F-Measure (0.975), Roc area (0.991), and Kappa Statistic (0.866) of Random Forest is better than the others classifiers. The more the Kappa statistic value close to 1 is the better the classifier. The time taken to build a model of classifier multilayer perceptron (MLP) is (4.57 seconds) higher than the rest classifiers, so MLP need more time and space to build a model.

Table 6: Weighted measure for each metrics and time taken to build model of 10 fold cross validation using 9 classifiers for page Block dataset

Accuracy Metrics	J48	Random Forest	MLP	SMO	IBK	Naïve Bayes	PART	Decision Table	Multinomial Logistic
TP Rate	0.969	0.975	0.961	0.929	0.960	0.908	0.970	0.956	0.965
FP Rate	0.148	0.117	0.220	0.600	0.199	0.234	0.140	0.260	0.198
Precision	0.967	0.974	0.959	0.932	0.959	0.938	0.969	0.953	0.963
Recall	0.969	0.975	0.961	0.929	0.960	0.908	0.970	0.956	0.965
F-Measure	0.968	**0.975**	0.959	0.909	0.959	0.919	0.969	0.954	0.963
ROC Area	0.939	**0.991**	0.968	0.737	0.880	0.940	0.949	0.970	**0.987**
Kappa Statistic	0.832	**0.866**	0.781	0.467	0.782	0.577	0.840	0.749	0.803
Time in second	0.18	1.21	**4.57**	0.23	0	0.02	0.09	0.33	0.66

From Table 7 we can deduced that 4863of 4913 instances (or 98.98%) of the actual were correctly classified as class of *text*, 22of 4913 were wrongly classified as class of *horiz. line*, 9 of 4913 were also incorrectly classified as class of *vert. line*, 13 of 4913 are also wrongly classified as class of ***picture*** and 6 were incorrectly classified as ***graphic*** class. Similarly 296 of 4913 instance (89.97%) were correctly classified as *horiz.line*, whereas 27 out of 329 instances were wrongly classified as *text* class. And 4 of 329 and 2 of 329 are wrongly classified as *vert.line* as *picture* classes respectively.

Table 7: 10 fold cross validation confusion matrix of Random Forest for Page Block dataset



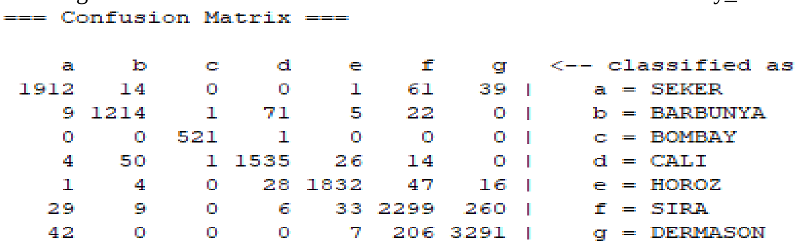
```
=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 4863   22    9   13    6 |    a = text
   27  296    4    2    0 |    b = horiz. line
    7    2   76    3    0 |    c = vert. line
   34    2    1   78    0 |    d = picture
    5    0    0    0   23 |    e = graphic
```

As we have seen from Table 8, logistic (multinomial logistic regression with ridge estimator) is performing better than the other classifiers relatively in all metrics for Bry_Bean dataset. And from confusion matrix of Table 9, we can see that 1912 of 2027 instances which is about 94.33% were correctly classified as SEKER class and the rest were wrongly classified to different classes, 1214 of 1322 instances, about 91.83% were correctly predicted as BARBUNYA class and the rest are wrongly classified to different classes. 521 of 522 which are about 99.81% were correctly classified as BOMBAY class, whereas 1 out of 522 was incorrectly predicted to the class of CALL. And the rest can be interpreted in similar way.

Table 8: Weighted measure for each metrics and time taken to build model of 10 fold cross validation using 9 classifiers for Dry_ Bean dataset

Accuracy Metrics	J48	Random Forest	MLP	SMO	IBK	Naïve Bayes	PART	Decision Table	Multinomial Logistic
TP Rate	0.913	0.926	0.925	0.922	0.903	0.897	0.913	0.880	**0.926**
FP Rate	0.020	0.018	0.018	0.019	0.023	0.022	0.021	0.027	**0.018**
Precision	0.913	0.926	0.925	0.923	0.903	0.898	0.913	0.880	**0.926**
Recall	0.913	0.926	0.925	0.922	0.903	0.897	0.913	0.880	**0.926**
F-Measure	0.913	0.925	0.925	0.922	0.903	0.897	0.913	0.880	**0.926**
ROC Area	0.967	0.992	0.991	0.976	0.941	0.990	0.977	0.984	**0.994**
Kappa Statistic	0.895	0.910	0.910	0.906	0.883	0.876	0.895	0.855	**0.911**
Time in second	0.65	5.7	**23.65**	0.49	0.01	0.08	1.67	1.23	137.13

Table 9: Logistic classifier confusion matrix of 10 fold cross validation for Dry_Bean dataset



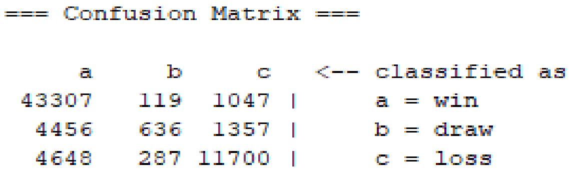
```
=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
 1912   14    0    0    1   61   39 |    a = SEKER
    9 1214    1   71    5   22    0 |    b = BARBUNYA
    0    0  521    1    0    0    0 |    c = BOMBAY
    4   50    1 1535   26   14    0 |    d = CALI
    1    4    0   28 1832   47   16 |    e = HOROZ
   29    9    0    6   33 2299  260 |    f = SIRA
   42    0    0    0    7  206 3291 |    g = DERMASON
```

As we seen from Table 5 and 10 Random Forest is relatively better classifier followed by multilayer perceptron by comparing different metrics that has been used in this paper, but SMO is costs much time (21386.83 seconds) to build the model followed by Multilayer perceptron (3532.99 seconds). Moreover, the confusion matrix Table 11 shows that 43,307 of 44,473 of the instances (97.38%) were correctly assigned as an actual class of ***win***, whereas 119 of 44,473 and 1,047 of 44,473 were wrongly assigned as the class of ***draw*** and ***loss*** respectively. And only 636 of 6,449 were (9.86%) correctly predicted as ***draw*** class but 4,456 of 6,449 which is about 69.10% were wrongly assigned as ***win*** class and 1,357 of 6,449 instances (21.04%) were incorrectly classified as ***loss*** class. Similarly, 11,700 of 1,6635 instances, about 70.33% were correctly classified as ***loss*** class and 4,648 of 16,635 instances (27.94%) were wrongly assigned as ***win*** class and the rest 287 of 16,635 were also ***draw*** class.

Table 10: Weighted measure for each metrics and time taken to build model of 10 fold cross validation using 9 classifiers for Connect-4 dataset

Accuracy Metrics	J48	Random Forest	MLP	SMO	IBK	Naïve Bayes	PART	Decision Table	Multinomial Logistic
TP Rate	0.809	**0.824**	0.819	0.759	0.809	0.721	0.793	0.753	0.758
FP Rate	0.213	0.272	0.202	0.340	0.326	0.426	0.206	0.343	0.338
Precision	0.791	**0.806**	0.793	0.720	0.800	0.681	0.781	0.719	0.729
Recall	0.809	**0.824**	**0.819**	0.759	0.809	0.721	0.793	0.753	0.758
F-Measure	0.797	0.792	**0.798**	0.717	0.771	0.681	0.786	0.724	0.717
ROC Area	0.868	**0.937**	0.899	0.722	0.935	0.807	0.861	0.823	0.856
Kappa Statistic	0.595	0.596	**0.613**	0.448	0.545	0.333	0.571	0.436	0.448
Time in second	1.98	26.39	3532.99	**21386.83**	0.02	1.18	165.5	40.77	28.95

Table 11: The Random Forest classifier confusion matrix of 10 fold cross validation for Connect-4 dataset



```
=== Confusion Matrix ===

     a     b     c   <-- classified as
 43307   119  1047 |     a = win
  4456   636  1357 |     b = draw
  4648   287 11700 |     c = loss
```

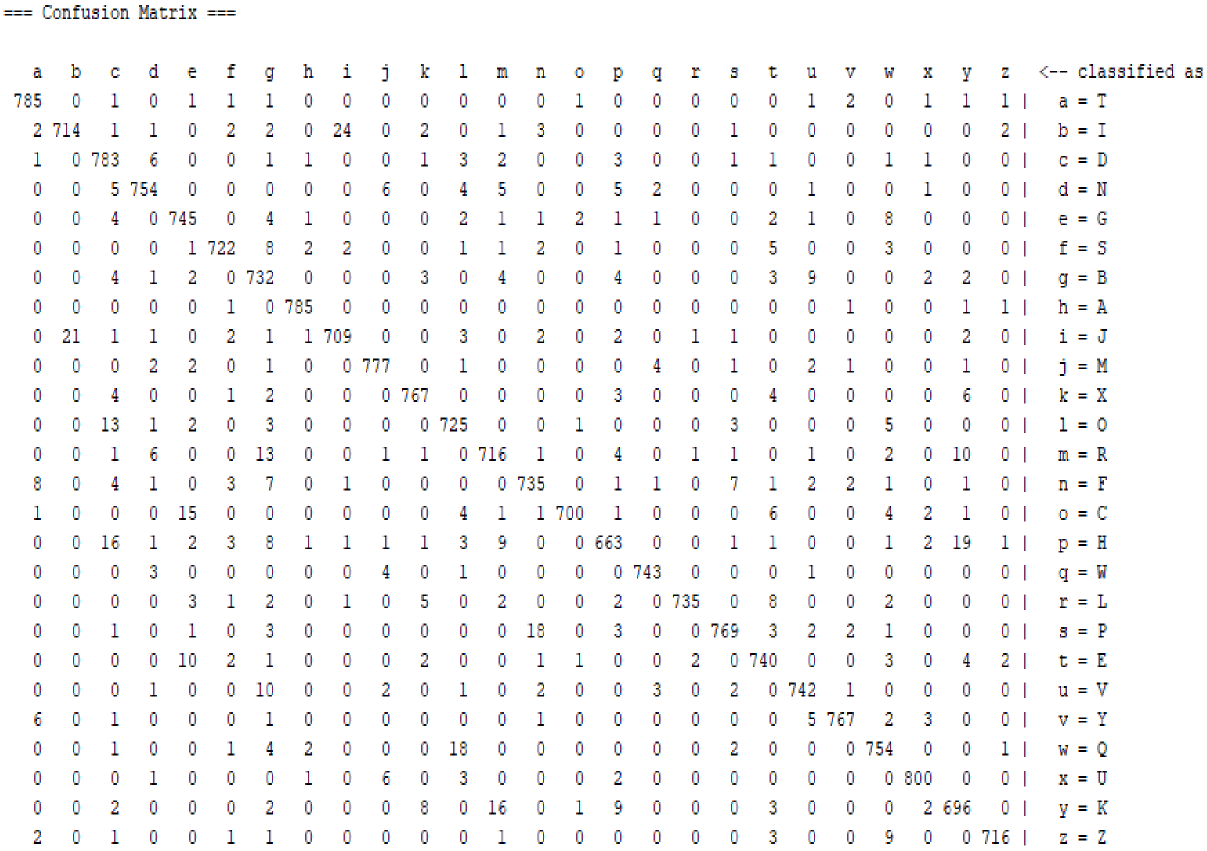
For Letter Recognition dataset of all metrics on Table 12 confirm that Random Forest classifier was outperform except the cost of time to build a model, whereas multilayer perceptron were costs much time than the other.

Table 12: Weighted measure for each metrics and time taken to build model of 10 fold cross validation using 9 classifiers for Letter Recognition dataset

Accuracy Metrics	J48	Random Forest	MLP	SMO	IBK	Naïve Bayes	PART	Decision Table	Multinomial Logistic
TP Rate	0.879	**0.964**	0.822	0.824	0.960	0.640	0.890	0.649	0.774
FP Rate	0.005	**0.001**	0.007	0.007	0.002	0.014	0.004	0.014	0.009
Precision	0.879	**0.964**	0.827	0.831	0.960	0.655	0.890	0.680	0.773
Recall	0.879	**0.964**	0.822	0.824	0.960	0.640	0.890	0.649	0.774
F-Measure	0.879	**0.964**	0.820	0.826	0.960	0.637	0.890	0.658	0.773
ROC Area	0.954	**0.999**	0.955	0.980	0.982	0.957	0.954	0.953	0.981
Kappa Statistic	0.874	**0.962**	0.815	0.817	0.958	0.626	0.886	0.635	0.765
Time in second	0.74	6.67	**130.46**	6.74	0.02	0.09	9.71	8.73	49.94

The confusion matrix in Table 13 is revealed that of Random Forest classifier classifies 785 of 796 instances (98.62%) were correctly classified as letter ***T*** class, the rests are wrongly assigned to different classes (refer the Table 12 below). 714 of 796 instances (89.97%) were correctly classified as letter ***I*** class and the rests are wrongly classified to different letter classes. Similarly, 783 of 796 instances (98.37%) were correctly assigned as letter ***D*** class, the rest about 1.63% were wrongly assigned to different classes. And 800 out of 813 instances, about 98.40% were correctly classified as an actual class of letter ***U***; whereas the rest 13 instances were wrongly assigned to 5 different classes (see Table 12). We can interpret the rest all letters classification in similar condition.

Table 13: The Random Forest classifier confusion matrix of 10 fold cross validation for Letter recognition dataset



```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   s   t   u   v   w   x   y   z   <-- classified as
 785   0   1   0   1   1   1   0   0   0   0   0   0   0   1   0   0   0   0   0   1   2   0   1   1   1 |   a = T
   2 714   1   1   0   2   2   0  24   0   2   0   1   3   0   0   0   0   1   0   0   0   0   0   0   2 |   b = I
   1   0 783   6   0   0   1   1   0   0   1   3   2   0   0   3   0   0   1   1   0   0   1   1   0   0 |   c = D
   0   0   5 754   0   0   0   0   0   6   0   4   5   0   0   5   2   0   0   0   1   0   0   1   0   0 |   d = N
   0   0   4   0 745   0   4   1   0   0   0   2   1   1   2   1   1   0   0   2   1   0   8   0   0   0 |   e = G
   0   0   0   0   1 722   8   2   2   0   0   1   1   2   0   1   0   0   0   5   0   0   3   0   0   0 |   f = S
   0   0   4   1   2   0 732   0   0   0   3   0   4   0   0   4   0   0   0   3   9   0   0   2   2   0 |   g = B
   0   0   0   0   0   1   0 785   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   1 |   h = A
   0  21   1   1   0   2   1   1 709   0   0   3   0   2   0   2   0   1   1   0   0   0   0   0   2   0 |   i = J
   0   0   0   2   2   0   1   0   0 777   0   1   0   0   0   0   4   0   1   0   2   1   0   0   1   0 |   j = M
   0   0   4   0   0   1   2   0   0   0 767   0   0   0   0   3   0   0   0   4   0   0   0   0   6   0 |   k = X
   0   0  13   1   2   0   3   0   0   0   0 725   0   0   1   0   0   0   3   0   0   0   5   0   0   0 |   l = O
   0   0   1   6   0   0  13   0   0   1   1   0 716   1   0   4   0   1   1   0   1   0   2   0  10   0 |   m = R
   8   0   4   1   0   3   7   0   1   0   0   0   0 735   0   1   1   0   7   1   2   2   1   0   1   0 |   n = F
   1   0   0   0  15   0   0   0   0   0   0   4   1   1 700   1   0   0   0   6   0   0   4   2   1   0 |   o = C
   0   0  16   1   2   3   8   1   1   1   1   3   9   0   0 663   0   0   1   1   0   0   1   2  19   1 |   p = H
   0   0   0   3   0   0   0   0   0   4   0   1   0   0   0   0 743   0   0   0   1   0   0   0   0   0 |   q = W
   0   0   0   0   3   1   2   0   1   0   5   0   2   0   0   2   0 735   0   8   0   0   2   0   0   0 |   r = L
   0   0   1   0   1   0   3   0   0   0   0   0   0  18   0   3   0   0 769   3   2   2   1   0   0   0 |   s = P
   0   0   0   0  10   2   1   0   0   0   2   0   0   1   1   0   0   2   0 740   0   0   3   0   4   2 |   t = E
   0   0   0   1   0   0  10   0   0   2   0   1   0   2   0   0   3   0   2   0 742   1   0   0   0   0 |   u = V
   6   0   1   0   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   5 767   2   3   0   0 |   v = Y
   0   0   1   0   0   1   4   2   0   0   0  18   0   0   0   0   0   0   2   0   0   0 754   0   0   1 |   w = Q
   0   0   0   1   0   0   0   1   0   6   0   3   0   0   0   2   0   0   0   0   0   0   0 800   0   0 |   x = U
   0   0   2   0   0   0   2   0   0   0   8   0  16   0   1   9   0   0   0   3   0   0   0   2 696   0 |   y = K
   2   0   1   0   0   1   1   0   0   0   0   0   1   0   0   0   0   0   0   3   0   0   9   0   0 716 |   z = Z
```

The statistical significance of the difference in accuracy of the tested classifier was tested by the mean difference of multi comparison. Table14 shows that the standard deviation (5.75) and standard error (2.35) of Random Forest were the least as compare as the other classifier, whereas, Decision Table classifier results with the highest standard deviation (13.63) and standard error (5.56). The results of the t-test are shown in Table 15, the p-value is only for the difference between Random Forest and Naïve Bayes model is (0.028) significant at the 5% level. But all the other pairs were not statistically significant. But the p-value of the mean difference between Random Forest and Decision Table is (0.064) statistically significant at 10% level of significance.

Also as we have seen from Fig. 3 mean plot, it visualize that the mean difference among Random Forest (RF) and Naïve Bayes (NB) was higher than the others. And also the mean difference between random forest (RF) and Decision Table (DT) is relatively higher next to RF Vs NB.

Table 14: Average descriptive Statistics

Types of Classifiers	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean	
				Lower Bound	Upper Bound
J48	90.28	7.49	3.06	82.41	98.14
RF	95.13	**5.75**	**2.35**	89.10	101.17
MLP	89.12	8.31	3.39	80.40	97.84
SMO	87.90	8.74	3.57	78.73	97.07
IBK	90.17	8.70	3.55	81.04	99.30
NB	82.71	12.23	4.99	69.87	95.54
PART	89.89	7.98	3.26	81.52	98.27
DT	84.70	**13.63**	**5.56**	70.39	99.00
ML	87.86	10.16	4.15	77.20	98.53
Total	88.64	9.39	1.28	86.10	91.20

Table 15: Statistical Comparison of the Average Classification Rates of nine Algorithms

Hypothesis	Mean Difference	P-value	Hypothesis	Mean Difference	P-value
H_0 : J48=RF	- 4.860	0.381	H_0 : IBK=NB	7.458	0.181
H_0 : J48=MLP	1.153	0.834	H_0 : IBK=PART	0.275	0.960
H_0 : J48=SMO	2.375	0.667	H_0 : IBK=DT	5.472	0.324
H_0 : J48=IBK	0.108	0.984	H_0 : IBK=Logistic	2.303	0.677
H_0 : J48=NB	7.567	0.175	H_0 : NB=J48	-7.567	0.175
H_0 : J48=PART	0.383	0.945	H_0 : NB=RF	**-12.423****	**0.028**
H_0 : J48=DT	5.580	0.315	H_0 : NB=MLP	-6.413	0.249
H_0 : J48=Logistic	2.412	0.662	H_0 : NB=SMO	-5.192	0.349
H_0 : RF=J48	4.857	0.381	H_0 : NB=IBK	-7.458	0.181
H_0 : RF=MLP	6.010	0.279	H_0 : NB=PART	-7.183	0.197
H_0 : RF=SMO	7.232	0.194	H_0 : NB=DT	-1.987	0.719
H_0 : RF=IBK	4.965	0.370	H_0 : NB=Logistic	-5.155	0.352
H_0 : RF=NB	**12.423****	**0.028**	H_0 : PART=J48	-0.383	0.945
H_0 : RF=PART	5.240	0.345	H_0 : PART=RF	-5.240	0.345
H_0 : RF=DT	**10.437***	**0.064**	H_0 : PART=MLP	0.770	0.889
H_0 : RF=Logistic	7.268	0.192	H_0 : PART=SMO	1.992	0.718
H_0 : MLP=J48	-1.153	0.834	H_0 : PART=IBK	-0.275	0.960
H_0 : MLP=RF	-6.010	0.279	H_0 : PART=NB	7.183	0.197
H_0 : MLP=SMO	1.222	0.825	H_0 : PART =DT	5.197	0.349
H_0 : MLP=IBK	-1.045	0.850	H_0 : PART=Logistic	2.028	0.713
H_0 : MLP=NB	6.413	0.249	H_0 : DT=J48	-5.580	0.315
H_0 : MLP=PART	-0.770	0.889	H_0 : DT=RF	**-10.437***	**0.064**
H_0 : MLP=DT	4.4267	0.424	H_0 : DT=MLP	-4.427	0.424
H_0 :MLP=Logistic	1.258	0.820	H_0 : DT=SMO	-3.205	0.562
H_0 : SMO=J48	-2.375	0.667	H_0 : DT=IBK	-5.472	0.324
H_0 : SMO=RF	-7.232	0.194	H_0 : DT=NB	1.987	0.719
H_0 : SMO=MLP	-1.222	0.825	H_0 : DT=PART	-5.197	0.349
H_0 : SMO=IBK	-2.267	0.681	H_0 : DT=Logistic	-3.168	0.566
H_0 : SMO=NB	5.192	0.349	H_0 : Logistic=J48	-2.412	0.662
H_0 : SMO=PART	-1.992	0.718	H_0 : Logistic=RF	-7.268	0.192
H_0 : SMO=DT	3.205	0.562	H_0 : Logistic=MLP	-1.258	0.820
H_0 : SMO=Logistic	0.037	0.995	H_0 : Logistic=SMO	-0.037	0.995
H_0 : IBK=J48	-0.108	0.984	H_0 : Logistic=IBK	-2.303	0.677
H_0 : IBK=RF	-4.965	0.370	H_0 : Logistic=NB	5.155	0.352
H_0 : IBK=MLP	1.045	0.850	H_0 : Logistic=PART	-2.028	0.713
H_0 : IBK=SMO	2.267	0.681	H_0 : Logistic=DT	3.168	0.566

**significant at 0.05 level and
*significant at 0.1 level
Source: Authors' work

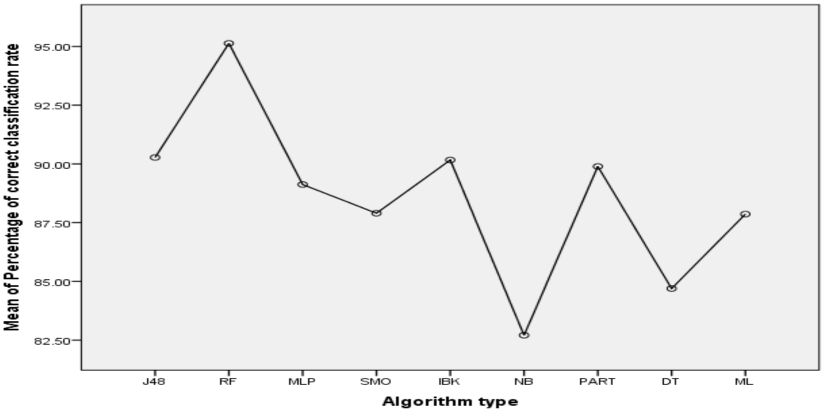


Fig. 3: The mean plot of correct classification in percentage

4.2 Discussion

This paper compares the classification performance of supervised machine learning algorithms in high dimensional problems. The execution of the algorithms was observed by the classification rates gotten in a 10-fold cross validation method and percentage split. The 10 fold cross validation is relatively outperform over percentage split. The algorithms used to compare were, J48 (tree based), Random Forest (tree based), Multilayer Perceptron (MLP) is a class of feed forward artificial neural network (ANN), IBK (k-nearest neighbour), sequential minimal optimization (SMO) works as of support vector machines, Naïve Bayes, PART, Decision Table and Logistic (it works by using Multinomial logistic regression model with ridged estimator) techniques were trained and tested. The results showed that the Random Forest classifier provides the most efficient model and outperforms other machine learning methods according to criteria of classification performance used in this paper: accuracy, precision, recall, F-Measure, ROC Area, kappa statistic and time to execution of the model. However, the accuracy of Random Forest is significantly higher at 5% level as compare with Naïve Bayes classifier, while the difference between the Random Forest and other tested methods is not found to be statistically significant. The reason for effectiveness of Random Forest model could be found in its ensemble nature usually trained with “bagging” method and the ability to minimize the error in the iterative procedure of optimizing its parameters such as learning rate. The "forest" it builds, is an ensemble of decision trees. The general idea of the bagging method is that a combination of learning models increases the accuracy. Most of the time, Random Forest is generating random subsets of the features and constructing smaller trees by using these subsets. Afterwards, it combines the sub trees. But this doesn’t works in every occasion and it also makes the computation slower, depending on how many trees the random forest builds. The main drawback of random forest is that a huge number of trees can make the algorithm too slow and ineffective for real-time predictions. In general, this algorithm is fast to train, but quite slow to create predictions once it trained. A more accurate prediction requires more trees, which results in a slower model. In most real-world situations, the Random Forest algorithm is fast enough but there can certainly be conditions where run-time performance is vital and other methods would be chosen. Both J48 and IBK were closely followed, but IBK has been taken more time to build the model as compare as the others. The J48, however, also learn fast and by providing a slightly lower classification average rate than Random Forest, are a very strong candidate for an efficient tool in these datasets after the Random Forest.

Although the above results cannot directly compared to earlier research outcomes, because of the reality that different authors used distinct datasets and have been mostly evaluating some limited classifiers used on this research, sure similarities and variations may be identified. Our results were consistent with the findings of [67, 68], even though, it were comparing different supervised machine learning algorithms for disease prediction, Random Forest were perform better than the other in the respective problems. However, our findings differ from the results of [69] who found that SVM (SMO) had best performed over NB, KNN, QDC and even the combined classifiers used in this study. But in our result KNN (IBK) was relatively better than SVM (SMO). Our findings show that accuracy of J48 was not significantly different from the accuracy of KNN (IBK), but confirm that J48 method produces the model with less variability, next to Random Forest.

4.3 Conclusion

Classification accuracy in a high-dimensional problem is still research area. Specially, using many and combined algorithms is a best way to see the highly accurate classifier. The objective of this paper was to provide a wide research by comparing the accuracy of nine supervised machine learning methods in order to analyze their classification efficiency in recognizing different set of problems with a large number of input variables. Our results show that all nine tested methods: J48, Random Forest, Multilayer Perceptron, SMO, IBK, Naïve Bayes, PART, Decision Table and Logistic regression are generally able to learn fast and achieve high classification accuracy even with a high-dimensional problems. The Random Forest outperformed other methods in classification accuracy, although the difference was significant at 5% level of significance only between the Random Forest and the Naïve Bayes model; and significant at 10% level between Random Forest and Decision Table classifier. The obtained findings partially confirm, and somewhat differ from previous research findings. Our t-test of mean difference results shows that except Random Forest versus Naïve Bayes and Random Forest versus Decision Table, the other pair was not significantly differ in their performance at 5% and 10% level of significant respectively, in fact, when we see the average classification performance, one is relatively better than the other. Our findings were differing from previous research in showing that Random Forest outperforming over J48, SMO, IBK and Multilayer Perceptron.

ACKNOWLEDGEMENT

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61702251, Youth Academic Talent Support Program of Northwest University, Natural Science Foundation of Guangdong Province under Grant No. 2018A030310688, Young Innovative Talents Project in Ordinary University of Guangdong Province under Grant No. 601821K37052, and NSERC Canada.

Authors Contribution:

Workineh Menna Eligo: Conceptualization, Data organizing, Formal analysis, Investigation and Writing original draft.

Aklilu Elias Kurika: Involving in pre-processing of dataset to the software and interpreting the output.

Chengcai Leng: Over all supervision
Anup Basu: review & editing

Ethical Statement

This research is not needed any ethical issues, since we have used already published data from UCI machine repository.

Funding Statement

The following are the funding list for each author:

1. **Chengcai Leng:** National Natural Science Foundation of China under Grant No. 61702251, Youth Academic Talent Support Program of Northwest University, Natural Science Foundation of Guangdong Province under Grant No. 2018A030310688, Young Innovative Talents Project in Ordinary University of Guangdong Province under Grant No. 601821K37052

2. Anup Basu: NSERC Canada

Data Accessibility

The dataset that we were used are available in UCI machine Learning Repository.

Competing Interests

There is no competing interest!

REFERENCES

[1] AE. Mohamed, *Comparative study of four supervised machine learning techniques for classification*. Information Journal of applied science and technology, 7 (2017).

[2] J. Zhu and T. Hastie, *Classification of gene microarrays by penalized logistic regression*. Biostatistics, 5 (2004), pp.427-443. doi:10.1093/biostatistics/kxg046

[3] Y. Liang, C. Liu, XZ. Luan, KS. Leung, T.M. Chan, Z.B. Xu and H. Zhang, *Sparse logistic regression with a $L_{1/2}$ penalty for gene selection in cancer classification*. BMC bioinformatics, 14 (2013), pp.1-12. doi: 10.1186/1471-2105-14-198

[4] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov and T. Poggio, *Multiclass cancer diagnosis using tumor gene expression signatures*. Proceedings of the National Academy of Sciences, 98 (2001), pp.15149-15154. doi: 10.1073pnas.211566398.

[5] B. Van Calster, K. Van Hoorde, Y. Vergouwe, S. Bobdiwala, G. Condous, E. Kirk, T. Bourne and E.W. Steyerberg, *Validation and updating of risk models based on multinomial logistic regression*. Diagnostic and prognostic research, 1 (2017), pp.1-14. doi: 10.1186/s41512-016-0002-x

[6] G. Gujari, Basic econometrics (2004), pp. 650-737.

[7] A. C. Tan and D. Gilbert, *An empirical comparison of supervised machine learning techniques in bioinformatics*, (2003).

[8] H. Park, *An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain*. J Korean Acad Nurs, 43 (2013), pp.154-164. doi.org/10.4040/jkan.2013.43.2.154.

[9] R. Caruana and A. Niculescu-Mizil, *An empirical comparison of supervised learning algorithms*. In Proceedings of the 23rd international conference on Machine learning (2006), pp. 161-168.

[10] R. King, C. Feng and A.Shutherland, *Statlog: comparison of classification algorithms on large real-world problems*. Applied Artificial Intelligence 9 (1995).

[11] S. Uddin, A. Khan, M.E. Hossain and M. A. Moni, *Comparing different supervised machine learning algorithms for disease prediction*. BMC medical informatics and decision making, 19 (2019), pp.1-16.

[12] M. Zekić-Sušac, S. Pfeifer, and N. Šarlija, *A comparison of machine learning methods in a high-dimensional classification problem*. Business Systems Research Journal, *5* (2014), pp. 82-96.

[13] J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for generalized linear models via coordinate descent*. J. Stat Softw, 33 (2010), pp: 1–22.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media (2009), pp: 656-661

[15] T. Obuchi, and Y. Kabashima, *Accelerating cross-validation in multinomial logistic regression with l_1-regularization*. The Journal of Machine Learning Research, 19 (2018), pp.2030-2059. http://jmlr.org/papers/v19/17-684.html.

[16] B.E. Boser, I.M. Guyon and V.N. Vapnik, *A training algorithm for optimal margin classifiers*. In Proceedings of the fifth annual workshop on Computational learning theory (1992), (pp. 144-152).

[17] V. Jakkula, *Tutorial on support vector machine (svm)*. School of EECS, Washington State University, 37(2006).

[18] F. Lauer, and Y. Guermeur,. MSVMpack: *a multi-class support vector machine package*. The Journal of Machine Learning Research, 12(2011), pp.2293-2296.

[19] T. Hastie, R. Tibshirani, and J.Friedman, *The Elements of Statistical Learning.* Springer-Verlag (2009), ISBN 978-0-387-84857-0

[20] M. Mohri, A. Rostamizadeh, A.Talwalkar, *Foundations of Machine Learning*. USA, Massachusetts: MIT Press, (2012), ISBN 9780262018258.

[21] A.J. Smola, and B. Schölkopf, *A tutorial on support vector regression*. Statistics and computing, 14 (2004), 199-222.

[22] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. A Wiley-Inter-science Publication. *John Wiley & Sons, Inc*., 2001.

[23] V. N. Vapnik, *An overview of statistical learning theory*. IEEE transactions on neural networks, 10 (1999), pp.988-999.

[24] F. L. Chung, W. Shitong, D. Zhaohong, and H. Dewen, *Fuzzy kernel hyperball perceptron*. Applied Soft Computing, 5 (2004), 67-74.

[25] I. H. Whitten, and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kauffmann (2000).

[26] G. H. John, and P. Langley, *Estimating continuous distributions in Bayesian classifiers,* (2013) arXiv preprint arXiv: 1302.4964.

[27] H. Lodhi, J. Shawe-Taylor, N. Christianini, C. Watkins, *Text classification using string kernels*, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems, 13(2001), MIT Press.

[28] C. W. Hsu, and C. J. Lin, *A comparison of methods for multiclass support vector machines*. IEEE transactions on Neural Networks, 13 (2002), pp. 415-425.

[29] M. Awad and R. Khanna. Support vector machines for classification. In *Efficient learning machines,* (2015). (pp. 39-66). Apress, Berkeley, CA.

[30] T. Hastie, and R. Tibshirani, *Classification by pairwise coupling*. Annals of statistics, 26 (1998), pp.451-471, (doi: 10.1214/aos/1028144844).

[31] B. Scholkopf and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, (2018). (doi: https://doi.org/10.7551 /mitpress/4175.001.0001).

[32] J. Weston, and C. Watkins, *Multi-class support vector machines. Royal Holloway, University of London*. Technical Report, CSD-TR-98-04. (1998)

[33] Y. Lee, Y. Lin and G. Wahba, *Multi-category support vector machines: Theory and application to the classification of microarray data and satellite radiance data*. Journal of the American Statistical Association, 99 (2004), pp.67-81. (doi: 10.1198/016214504000000098)

[34] A.M. El-Habil, *An application on multinomial logistic regression model*. Pakistan journal of statistics and operation research, (2012) pp.271-291. doi:10.18187/pjsor.v8i2.234.

[35] Lee, K., Ahn, H., Moon, H., Kodell, R. L., & Chen, J. J. Multinomial logistic regression ensembles. *Journal of Biopharmaceutical Statistics*, *23*(3) (2013). (pp. 681-694). doi: 10.1080/10543406.2012.756500.

[36] S. Le Cessie, and J. C. Van Houwelingen, *Ridge estimators in logistic regression*. Journal of the Royal Statistical Society*:* Series C (Applied Statistics), 41(1992), pp. 191-201.

[37] G. Kaur and A. Chhabra, *Improved J48 classification algorithm for the prediction of diabetes*. International journal of computer applications, 98 (2014).

[38] R. C. Holte, *Very simple classification rules perform well on most commonly used datasets*. Machine learning, 11(1993), pp.63-90.

[39] E. Frank and I. H. Witten, *Generating accurate rule sets without global optimization*, (1998).

[40] R. P. Duin, *A note on comparing classifiers*. Pattern Recognition Letters, 17(1996), pp. 529-536.

[41] L. Breiman, *Random forests*. Machine learning, 45(2001), pp.5-32. https:// doi.org/ 10. 1023/A: 1010933404324.

[42] Y. Amit, and D. Geman, *Shape quantization and recognition with randomized trees*. Neural computation, 9 (1997), pp.1545-1588. https://doi.org/10.1162/neco.1997.9.7.1545.

[43] D.R Cutler, T.C. Edwards Jr, K.H. Beard, A. Cutler, K.T. Hess, J. Gibson, and J.J. Lawler, *Random forests for classification in ecology*. Ecology, 88(2007), pp.2783-2792. https://doi.org/10.1890/07-0539.1

[44] B. Ghimire, J. Rogan and J. Miller, *Contextual land-cover classification: incorporating spatial dependence in land-cover classification models using random forests and the Getis statistic*. Remote Sensing Letters, 1(2010), pp.45-54. doi: https://doi.org/10.1080/01431160903252327

[45] M. Seera, and C.P. Lim, *A hybrid intelligent system for medical data classification*. Expert Systems with Applications, 41(2014), pp.2239-2249. doi:10.1016/j.eswa.2013.09.022.

[46] J.I. Titapiccolo, M. Ferrario, S. Cerutti, C. Barbieri, F. Mari, E. Gatti, and M.G., Signorini,. *Artificial intelligence models to stratify cardiovascular risk in incident hemodialysis patients*. Expert Systems with Applications, 40 (2013), pp.4679-4686. doi: https://doi.org/10.1016/j.eswa.2013.02.005.

[47] J. R. Quinlan, *C4.5: Programs for machine learning*. Morgan Kaufman, San Mateo, CA, (1993).

[48] S. Ali and K.A. Smith, *On learning algorithm selection for classification*. Applied Soft Computing, 6 (2006), pp.119-138.

[49] UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets.php

[50] R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*. In Ijcai, 14 (1995), pp. 1137-1145.

[51] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementations*. Acm Sigmod Record, 31 (2002), pp. 76-77.

[52] D. Simon, and J.R. Boring, *Sensitivity, Specificity, and Predictive Value, in Walker, H.K., Hall, W.D., Hurst, J.W. (Eds.), Clinical Methods: The History, Physical and Laboratory Examinations, Butterworths, Boston*, (1990), pp. 49-54.

[53] S. Visa, B. Ramsay, A. Ralescu, and E. VanDerKnaap, *Confusion Matrix-Based Feature selection. Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference (2011)*, pp.120-127. Retrieved from https://openworks.wooster.edu/facpub/88.

[54] J. A. Hanley, and B. J. McNeil, *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. Radiology, 143 (1982), pp. 29-36.

[55] S. Narkhede, *Understanding AUC-ROC curve*. Towards Data Science, 26 (2018), pp.220-227.

[56] J. Cohen, *A coefficient of agreement for nominal scales*. Educational and psychological measurement, 20 (1960), pp.37-46.

[57] A. Ben-David, *Comparison of classification accuracy using Cohen's Weighted Kappa*. Expert Systems with Applications, 34 (2008), pp.825-832.

[58] I.H. Witten and E. Frank, *Data mining*. Academic Press, 2005

[59] M. L. McHugh, *Interrater reliability: the kappa statistic*. Biochemia medica, 22 (2012), pp. 276-282.

[60] S. Sun, *Meta-analysis of Cohen's kappa*. Health Services and Outcomes Research Methodology, 11 (2011), pp.145-163.

[61] S. M. Vieira, U. Kaymak, and J. M. Sousa, *Cohen's kappa coefficient as a performance measure for feature selection*. In International Conference on Fuzzy Systems IEEE, (2010), pp. 1-8.

[62] A.J. Viera, and J.M. Garrett, *Understanding inters observer agreement: the kappa statistic*. Fammed, 37 (2005), pp.360-363.