BOOTSTRAPPING SUPERVISED CLASSIFIER PARADIGM

Oloyede I.

Department of Statistics, University of Ilorin

Corresponding Author: Oloyede I., oloyede.i@unilorin.edu.ng

ABSTRACT: The study investigates the classification of learning algorithms in a bootstrap paradigm, the study examined features classification with binary class attributes in a bootstrap paradigm. Support Vector Machine, k-Nearest Neighbour, Random Forest, rpart, Artificial Neural Network and Naïve Bayes learning algorithms were compared. Accuracy, Prediction error, Sensitivity and Specificity were used as assessment criteria of the classifier after tuning to have minimum cost. The study therefore sample the training set and classifying each of the training set, the summary of the prediction error was obtained based on the testing dataset, the study showed that artificial neural network outperformed other learning algorithms with respect to accuracy criterion whereas the celebrated support vector machine performed poorly amongst the learning algorithms considered**,** the study depicted that artificial neural network outperformed other learning algorithms with the least misclassification error. The study depicted that K nearest neighbour outperformed other learning algorithms with highest sensitivity while ANN outperformed other learning algorithms with highest specificity. This study affirmed that there would be need to use more than a learning algorithm when there are irrelevant features in the data sets.

KEYWORDS: SVM, Naïve Bayes, Bootstrap, Classification and learning algorithms.

1. INTRODUCTION

In an attempt to elicit information from the data, estimation, detection and classification approaches are the tools commonly used by the research. Classification is one of the technique used to elicit information from the data in the way in which objects of features are separated into classes particularly when the dependent variable is categorical either binary or multiclass, Ahmad et. al ([AIM13]). The classification algorithms occur in the phases of learning algorithm which examine model for the class attribute as function of the features (independent variable) of the datasets, which adopted training dataset whereas the second aspect is to apply the designed model in the first part to the new dataset (testing) so as to determine the related class of each of the features Tan et. al ([TSK06]).

Classification had been applied to many fields such as sciences, education, engineering as a way of pattern recognition and visualization of datasets based on the class attribute, Li and Jain ([LJ98]) pointed out that naïve Bayes and subspace method outperformed nearest neighbour, decision tree in their document classification experiment, naïve Bayes outperformed others learning algorithm in testing dataset1 whereas subspace method outperformed others algorithm in the testing dataset2.

Xhemali et. al ([XHS09]) compared Naïve Bayes and decision tree in a web page training set and found out that Naïve Bayes accuracy exceed decision tree. Ahmad et. al ([AIM13]) opined that Naïve Bayes is simple but always outperformed other classification methods, Naïve Bayes had been proved to be fast, consistent and accurate in the classification procedures. Ahmad et. al ([AIM13]) showed that decision tree proved to be the fastest algorithm while k-nearest neighbour was the slowest algorithm for the three classifiers compared in their study. This was due to absence of calculation in the tree. They added that Naïve Bayes outperformed other leaning algorithms. Rich and Alexandru ([RA06]) claimed that learning algorithm should be evaluated in a broader performance metrics since different learning algorithms are built to optimize different criteria.

Amancio et.al ([A+14]) claimed in their study that k-nearest neighbour usually outperformed other algorithms with the default parameter of weka software used in their study when they used artificial dataset. They added that multilayer perception outperformed Bayesians network, c4.5, svm and cart in most datasets considered in their study. They further observed that support vector machine depicted low overall performance compared to other learning algorithms. Rich and Alexandru ([RA06]) argued that calibrated boosted decision trees outperformed other learning algorithms considered in their study followed by calibrated random forest, uncalibrated neural networks, calibrated svm and bagged trees. They concluded that Naïve Bayes, logistic regression, decision trees and boosted stumps performed poorly. Aik and David ([AD03]) compared different learning algorithms to classifying biological datasets. They concluded that none of the learning algorithms consistently performed well over other in all the training datasets, though when they were integrated they overperformed one another using accuracy, specificity, sensitivity and positive predicted value criteria. Abhisek ([Abh17]) compared different learning algorithms which were used to classify the heart disease correctly with different performance metrics and concluded that Artificial Neural Network and Support Vector Machine were the best learning algorithms for the heart disease dataset. Having examined series of literature, this study therefore seeks to investigate the comparison of learning algorithms in a bootstrap paradigm where moderate Multicollinearity exists amongst some irrelevant features.

2. MATERIAL AND METHODS

In an alternative to cross validation which seems to be cumbersome and complicated bootstrap proved to be best technique for the learning algorithms. This study therefore adopted bootstrap technique to validate the learning algorithms compared. In an attempt to have unique result we set the seed to 1223, thus 1000 iterations were adopted in the bootstrap paradigm. We reported the average of the accuracy, misspecification error of prediction, sensitivity and specificity. The whole iteration is depicted in the graphs displayed for each learning algorithms. The model was examined on the bootstrapped training datasets while the prediction was examined on the bootstrapped testing datasets so as to validate the learning algorithms and make reasonable comparisons.

2.1. Support Vector Classification

Let $\{(x_1,y_1),(x_2,y_2),(x_3,y_3),\dots(x_m,y_m)\}$ where with $x_i \varepsilon\ H$ being a feature variables and $y \sim (\pm 1)$ be training set supposedly to be classified. Intuitively, the task is to obtain a linear decision boundary parameterised by weight vector w and constant b : $w'x_i + b \geq 0$ whenever $y_i = +1$ and $(w, x_i) + b < 0$ whenever $y_i = -1$.

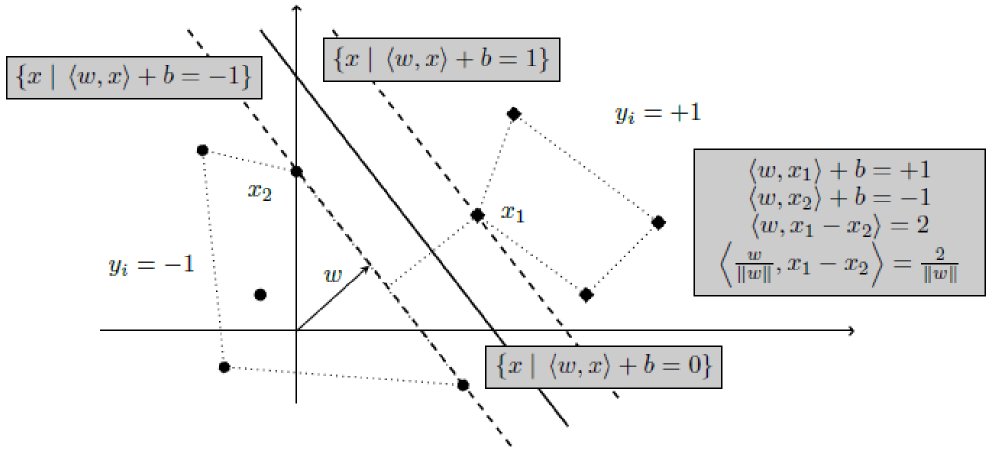
$\{x \mid \langle w,x\rangle + b = -1\}$
$\{x \mid \langle w,x\rangle + b = 1\}$
$y_i = +1$
$\langle w,x_1\rangle + b = +1$
$\langle w,x_2\rangle + b = -1$
$\langle w,x_1 - x_2\rangle = 2$
$\left\langle \frac{w}{\|w\|}, x_1 - x_2\right\rangle = \frac{2}{\|w\|}$
x_2
x_1
$y_i = -1$
w
$\{x \mid \langle w,x\rangle + b = 0\}$

Figure 1.

in an attempt to maximize the margin between the two class via hyperplane, we have:

$$\underset{w,b}{max}\ \frac{1}{\|w\|} \text{ or } \underset{w,b}{mix}\ \frac{1}{2}\| w \|^2 \qquad (2.1)$$

$$s.t\ y_i(w,x_i) + b \geq 1 \quad \forall\ i \qquad (2.2)$$

2.1.1 Hard margin classifier

If the training data $\{(x_1,y_1),(x_2,y_2),(x_3,y_3),\dots(x_m,y_m)\}$ is linearly separable thus the hyperplane correctly classifies the training data and then refer to as hard margin classifier.

2.1.2 Soft margin classifier

The constrained convex optimisation problem with a quadratic objective function where implicitly the data is not linearly separable, then non-negative slack variable $\mathcal{E}_i$ is introduced to relax the constraints.

$$y_i\left((w,x_i) + b\right) \geq 1 - \mathcal{E}_i \qquad (2.3)$$

Given any w and b the constrained can now be satisfied by making $\mathcal{E}_i$ large enough. This renders the whole optimization problem useless. Therefore, one has to penalise large $\mathcal{E}_i$. This is done via the following modified optimization problem:

$$\underset{w,b,\mathcal{E}}{min}\ \frac{1}{2}\| w \|^2 + \frac{C}{m}\sum_{i=1}^{m} \mathcal{E}_i \qquad (2.4)$$

$$s.t. y_i\big((w,x_i)+b\big) \geq 1-\mathcal{E}_i \qquad (2.5)$$

$$\mathcal{E}_i \geq 0 \qquad (2.6)$$

where $C\ > 0$ is a penalty parameter. The resultant classifier is said to be a soft margin classifier. By introducing non-negative Lagrange multiplier α_i and β_i one can write the Lagrangian.

$$L(w,b,\varepsilon,\alpha,\beta)=\frac{1}{2}\parallel w \parallel^2+\frac{C}{m}\sum_{i=1}^{m}\mathcal{E}_i+\sum_{i=1}^{m}\alpha_i\Big(1-\mathcal{E}_i-y_i\big((w,x_i)+b\big)\Big)-\sum_{i=1}^{m}\beta_i\,\mathcal{E}_i \qquad (2.7)$$

Then differentiate $L(w,b,\varepsilon,\alpha,\beta)$ with respect to w,b, ϵ and set them to zero

$$\frac{dL}{dw}=\frac{1}{2}\frac{d}{dw}\parallel w \parallel^2+\frac{d}{dw}\frac{C}{m}\sum_{i=1}^{m}\mathcal{E}_i+\frac{d}{dw}\sum_{i=1}^{m}\alpha_i\Big(1-\mathcal{E}i-yiw,xi+b-ddwi=1m\beta i\mathcal{E}i \qquad (2.8)$$

$$\frac{dL}{dw}=\frac{1}{2}\frac{d}{dw}\parallel w \parallel^2+\frac{d}{dw}\sum_{i=1}^{m}(\alpha_i-\alpha_i\mathcal{E}_i-\alpha iyiw,xi-\alpha iyib \qquad (2.9)$$

$$\frac{dL}{dw}=w-\sum_{i}^{m}\alpha_i\,y_ix_i=0 \qquad (2.10)$$

$$w=\sum_{i}^{m}\alpha_i\,y_ix_i \qquad (2.11)$$

$$\frac{dL}{db}=\frac{1}{2}\frac{d}{db}\parallel w \parallel^2+\frac{d}{db}\frac{C}{m}\sum_{i=1}^{m}\mathcal{E}_i+\frac{d}{db}\sum_{i=1}^{m}\alpha_i-\alpha i\mathcal{E}i-\alpha iyiw,xi-\alpha iyib-ddbi=1m\beta i\mathcal{E}i \qquad (2.12)$$

$$\frac{dL}{db}=\frac{d}{db}\sum_{i=1}^{m}\alpha_i-\alpha_i\mathcal{E}_i-\alpha_iy_iw,x_i-\alpha_iy_ib \qquad (2.13)$$

$$\frac{dL}{db}=\frac{d}{db}\sum_{i=1}^{m}\alpha_i-\alpha_i\mathcal{E}_i-\alpha_iy_iw,x_i-\alpha_iy_ib \qquad (2.14)$$

$$\frac{dL}{db}=-\sum_{i}^{m}\alpha_i\,y_i=0 \qquad (2.15)$$

$$\frac{dL}{d\mathcal{E}_i}=\frac{1}{2}\frac{d}{d\mathcal{E}_i}\parallel w \parallel^2+\frac{d}{d\mathcal{E}_i}\frac{C}{m}\sum_{i=1}^{m}\mathcal{E}_i+\frac{d}{d\mathcal{E}_i}\sum_{i=1}^{m}\alpha_i-\alpha i\mathcal{E}i-\alpha iyiw,xi-\alpha iyib-dd\mathcal{E}ii=1m\beta i\mathcal{E}i \qquad (2.16)$$

$$\frac{dL}{d\mathcal{E}_i}=\frac{d}{d\mathcal{E}_i}\frac{C}{m}\sum_{i=1}^{m}\mathcal{E}_i+\frac{d}{d\mathcal{E}_i}\sum_{i=1}^{m}\alpha_i-\alpha_i\mathcal{E}_i-\alpha_iy_iw,x_i-\alpha_iy_ib-\frac{d}{d\mathcal{E}_i}\sum_{i=1}^{m}\beta_i\,\mathcal{E}_i \qquad (2.17)$$

$$\frac{dL}{d\mathcal{E}_i}=\frac{d}{d\mathcal{E}_i}\frac{C}{m}\sum_{i=1}^{m}\mathcal{E}_i+\frac{d}{d\mathcal{E}_i}\sum_{i=1}^{m}\alpha_i\mathcal{E}_i-\frac{d}{d\mathcal{E}_i}\sum_{i=1}^{m}\beta_i\,\mathcal{E}_i \qquad (2.18)$$

$$\frac{dL}{d\mathcal{E}_i}=\frac{C}{m}-\sum_{i=1}^{m}\alpha_i-\sum_{i=1}^{m}\beta_i=0 \qquad (2.19)$$

$$\frac{C}{m}=\sum_{i=1}^{m}\alpha_i+\sum_{i=1}^{m}\beta_i \qquad (2.20)$$

Substituting into the Lagrangian and simplifying yields the dual objective function.

$$L(w,b,\varepsilon,\alpha,\beta)=\frac{1}{2}\Big(\sum_{i}^{m}\alpha_i\,y_ix_i\Big)^2+\Big(\sum_{i=1}^{m}\alpha_i+\sum_{i=1}^{m}\beta_i\Big)\sum_{i=1}^{m}\mathcal{E}_i+\sum_{i=1}^{m}(\alpha_i-\alpha_i\mathcal{E}_i-\alpha_iy_iw,x_i-\alpha_iy_ib)\ -\sum_{i=1}^{m}\beta_i\,\mathcal{E}_i \qquad (2.21)$$

$$L(w,b,\varepsilon,\alpha,\beta)=\frac{1}{2}\Big(\sum_{i}^{m}\alpha_i\,y_ix_i\Big)^2+\sum_{i=1}^{m}\alpha_i\mathcal{E}_i+\sum_{i=1}^{m}\beta_i\mathcal{E}_i+\sum_{i=1}^{m}\alpha_i-\sum_{i=1}^{m}\alpha_i\mathcal{E}_i-\sum_{i=1}^{m}\alpha_iy_iw,x_i-\sum_{i=1}^{m}\alpha_iy_ib\ -\sum_{i=1}^{m}\beta_i\,\mathcal{E}_i \qquad (2.22)$$

$$L(w,b,\varepsilon,\alpha,\beta)=\frac{1}{2}\Big(\sum_{i}^{m}\alpha_i\,y_ix_i\Big)^2+\sum_{i=1}^{m}\alpha_i-\sum_{i=1}^{m}\alpha_iy_iw,x_i-\sum_{i=1}^{m}\alpha_iy_ib \qquad (2.23)$$

$$L(w,b,\varepsilon,\alpha,\beta)=\frac{1}{2}\Big(\sum_{i}^{m}\alpha_i\,y_ix_i\Big)^2+\sum_{i=1}^{m}\alpha_i-\sum_{i=1}^{m}\alpha_iy_i\alpha_iy_ix_ix_i-\sum_{i=1}^{m}\alpha_iy_ib \qquad (2.24)$$

$$L(w,b,\varepsilon,\alpha,\beta)=\frac{1}{2}\sum_{ij}y_i\,y_j\alpha_i\alpha_j\big(x_ix_j\big)+\sum_{i=1}^{m}\alpha_i-\sum_{ij}y_i\,y_j\alpha_i\alpha_j\big(x_ix_j\big)-\sum_{i=1}^{m}\alpha_iy_ib \qquad (2.25)$$

$$L(w,b,\varepsilon,\alpha,\beta)=-\frac{1}{2}\sum_{ij}y_i\,y_j\alpha_i\alpha_j\big(x_ix_j\big)+\sum_{i=1}^{m}\alpha_i\ \ -\sum_{i=1}^{m}\alpha_iy_ib \qquad (2.26)$$

Recall that $-\sum_{i}^{m}\alpha_i\,y_i=0$ therefore we have

$$L(w,b,\varepsilon,\alpha,\beta)=-\frac{1}{2}\sum_{ij}y_i\,y_j\alpha_i\alpha_j\big(x_ix_j\big)+\sum_{i=1}^{m}\alpha_i \qquad (2.27)$$

There is need maximized the objective function with respect to α which is equivalent to minimization as expressed below. Recall that $\alpha_i \geq 0$ and $\beta_i \geq 0$ and $0 \leq \sigma_i \leq \frac{c}{m}$.

$$\frac{min}{\alpha}\frac{1}{2}\sum_{i,j}y_i\,y_j\alpha_i\alpha_j\big(x_i,x_j\big)-\sum_{i=1}^{m}\alpha_i \qquad (2.28)$$

$$s.t\ \sum_{i=1}^{m}\alpha_iy_i=0 \qquad (2.29)$$

$$0\leq\alpha_i\leq\frac{C}{m} \qquad (2.30)$$

By the KKT conditions (Section 3.3) we have

$$\alpha_i(1-\varepsilon_i-y_i\big((w,x_i)\big)+b))=0\ and\ \beta_i\varepsilon_i=0 \qquad (2.31)$$

The study considered two cases for $y_i((w,x_i)+b$ and the implications of the KKT condition

2.2. Neural Networks

A neural network is a two stage regression or classification model, typically represented by a network diagram. For K-class classification, there are *K* units at the top, with the *k*th unit modelling the probability of class *k*. There are *K* target measurements Y*k*, k=1…, each being coded as a 0-1 variable for the *k*th class

Derived features Z_m are created from linear combinations of the inputs, and then the target Ym is modelled as a function of linear combinations of the inputs, and then the target Y_k is modelled as a function of linear combination of the Z_m

$$Z_m=\sigma(\sigma_{0m}+\alpha_m^TX),m=1\ldots,M, \qquad (2.32)$$

$$T_k=\beta_{0k}+\beta_k^TZ,k=1,\ldots K, \qquad (2.33)$$

$$f_k(X)=g_k(T),k=1,\ldots,K, \qquad (2.34)$$

Where
$Z = (Z_i, Z_2, \dots Z_m), and\ T = (T_i, T_2, ..T_k)$ (2.35)

2.3. K-Nearest Neighbor Classifier

KNN is a learning algorithm that makes predictions on the model using the testing dataset by making new instance, this is achieved by searching through the whole training dataset for the K most similar instances which is regarded as the neighbors and summarizing the output variables for those set of k instances. Euclidean distance is the tool used to determine the K instances in the training dataset that is most similar to a new input. The Euclidean distance is obtained as $(x, x_i) = \sqrt{\sum(x_i - y_i)^2}$. for the uniformity of the values of the distances, the features in the training dataset is standardized as

$$X_i = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (2.36)$$

2.4. Naïve – Bayes Classifier

Naïve Bayes classifier or idiot Bayes in a learning algorithms which uses the Bayes theorem to classify the dataset into model attributes and it assumed that the probability of certain feature X_i is totally independent of another feature X_j Ethan (2010). Bayes theorem can be expressed as $P(A/B) = \frac{P(B/A)P(A)}{P(B)}$ and it provides the means through which the probability of the hypothesis can be obtained given the prior knowledge about the problem.

$$Posterior\ = \frac{Likelihood\ *\ Prior}{Evidence}$$

2.5. Random Forest

This is a classifier that consists of a collection of decision trees, where each tree is constructed by applying an algorithm A on the training dataset S and an additional random vector θ where θ is sampled i.i.d from some distribution. The prediction of the random forest is obtained by a majority vote over the predictions of the individual trees. Random K data points are selected from the training dataset, then a decision tree is built for the selected K data points. Thereafter, the Ntree subset are selected from the trees, finally, result on the basis of the majority of votes is decided Shai and Shai (2014) and Abhisek (2017).

3. DATA GENERATION PROCESS

Artificial dataset of low dimensional was generated, the main focus is to assess various learning algorithms in a way to classify training datasets in to binary class attribute subject to the redundancy in the dataset generated. The dataset was partitioned into training dataset which was used to fit the model for all the learning algorithms and the testing dataset which was used to predict the model for all the learning algorithms at the ratio of 80:20 (training:testing) respectively. 10 dimensions were redundant with moderately collinear, the probability of class attribute of y=1 or -1 was set to be equal. The first three features' were drawn from uniform distribution with minimum(4.3;2;1) and maximum (7.9;4.4;6.9) respectively whereas the remaining fifteen features were generated with normal distribution with mean 0 and standard deviation of 20.

3.1. Performance Metrics

Performance metrics imply measurement or performance criteria on how well a learning algorithm perform in features classification, regression and selection. This adduces to various criteria such as accuracy, precision, misspecification error rate, sensitivity and specificity.

3.2. Confusion matrix

This depicts how the learning algorithm is confused after classification when model is predicted with testing datasets.

Table 1.

Correct Classification	Positive	Negative
Positive	TP (1,1)	FN (1,0)
Negative	FP (0,1)	TN (0,0)

TP is the true positive value that is truly classified, FP is the false positive value that falsely classified, TN is the true negative value that is truly classified while FN is the false negative value that falsely classified.

Accuracy is the measure of the proportion of proportion that is correctly classified, ([Max13]).

$acc = \frac{TP+TN}{TP+TN+FP+FN}$ THIS implies how close the predicted value close to the actual value, the misclassification error can be obtained a $1 - accuracy$.

4. RESULTS AND INTERPRETATION

Table 2. Results

Learning Algorithms	Accuracy	MissClassification Errors	Sensitivity	Specificity	PPV
SVM	.4818	.5183	.8267	.3339	0.3474
Rpart	.4931	.5070	.591	0.4511	0
Random forest	.4912	.5089	.601	.4441	.3169
Naïve Bayes	.5023	.4978	.6923	.4208	.3376
KNN	.5	.5	1	. 2857	.375
Ann	.5195	.4805	.6123	.4797	.3379

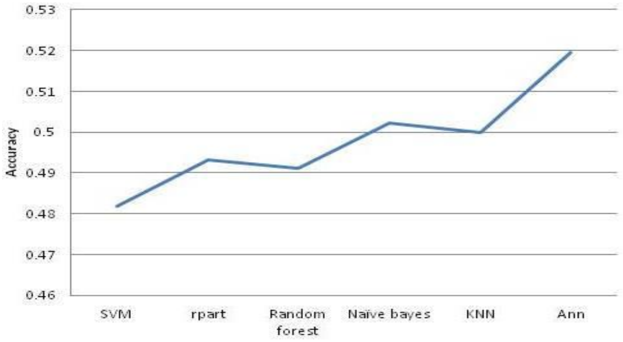

Figure 2. The accuracy of learning algorithms

Accuracy: the study showed that artificial neural network outperformed other learning algorithms in a bootstrap paradigm where the model was predicted based on the testing datasets. This was followed by naïve Bayes, K nearest neighbour, rpart, Random forest. it is support vector machine that performed poorly amongst the learning algorithms with 1000 bootstrap replications.

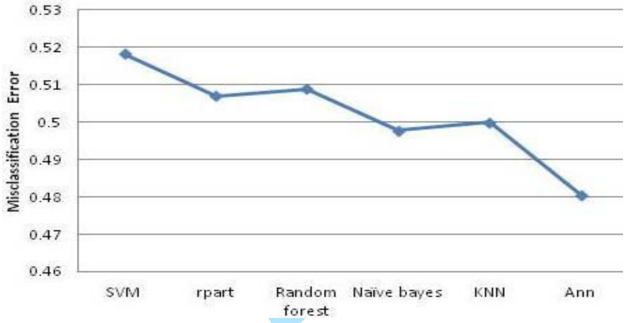

Figure 3. The misclassification errors of learning algorithms

Misclassification Error: the study depicted that artificial neural network outperformed other learning algorithms in a bootstrap paradigm with minimum misclassification error of 0.4805. This was followed by naïve Bayes, K nearest neighbour, rpart, Random forest. It was support vector machine that performed poorly amongst the learning algorithms with highest misclassification error of 0.5183 in a bootstrap replication.

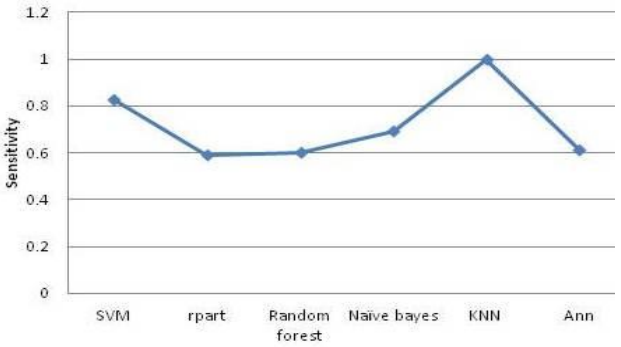

Figure 4. The sensitivity of learning algorithms

Sensitivity: the study depicted that K nearest neighbour outperformed other learning algorithms in a bootstrap paradigm with highest sensitivity. This was followed by support vector machine, naive Bay, Random forest, ANN and rpart.

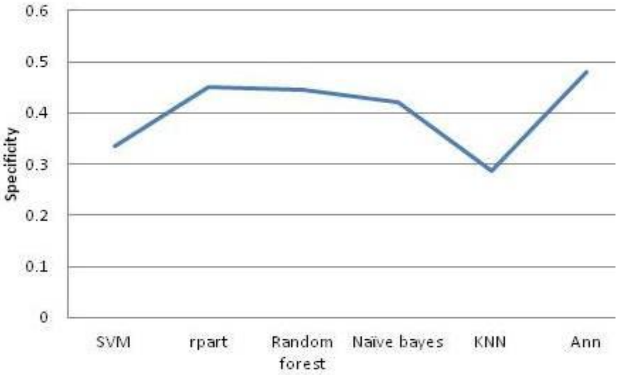

Figure 5. The specificity of learning algorithms

Specificity: the study depicted that ANN outperformed other learning algorithms in a bootstrap paradigm with highest specificity. This was followed by rpart, random forest, naïve Bayes, support vector machine while the least is KNN.

CONCLUSION

The study examined features classification with binary class attributes in a bootstrap paradigm. Support Vector Machine, k-Nearest Neighbour, Random Forest, rpart, Artificial Neural Network and

Naïve Bayes learning algorithms were compared, the study showed that artificial neural network outperformed other learning algorithms with respect to accuracy criterion whereas the celebrated support vector machine performed poorly amongst the learning algorithms considered**,** the study depicted that artificial neural network outperformed other learning algorithms with the least misclassification error. The study depicted that K nearest neighbour outperformed other learning algorithms with highest sensitivity while ANN outperformed other learning algorithms with highest specificity. This study affirmed that there would be need to use more than a learning algorithm when there are irrelevant features in the data sets. Our study is in line with what are found in the literature.

REFERENCES

[Abh17] **Abhisek A. -** *Comparative Study of Machine Learning Algorithms for Heart Disease Prediction*, unpublished thesis, Helsinki Metropolia University of Applied Sciences Bachelor of Engineering, Information Technology, Pp 1:52, 2017.

[AIM13] **Ahmad A., Iman P., Min T. -** *Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool*, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11, pp 33:39, 2013, www.ijacsa.thesai.org

[AD03] **Aik C.T., David G. -** *An empirical comparison of supervised machine learning techniques in bioinformatics*, appear in the proceedings of the first Asia Pacific Bioinformatics Conference, 2003.

[A+14] **Amancio D. R., Comin C. H., Casanova D., Travieso G., Bruno O. M. -** *A Systematic Comparison of Supervised Classifiers*. PLoS ONE 9(4): e94137, 2014. doi:10.1371/journal.pone.0094137

[Eth10] **Ethan A. -** *Introduction to Machine Learning,* 2nd ed. Cambridge Massachusetts, MIT Press, 2010.

[LJ98] **Li Y. H., Jain A. K. -** *Classification of Text Document*, The Computer Journal, Vol. 41(8), 1998.

[Max13] **Max B. -** *Principles of Data Mining*. 2nd ed. Springer, 2013.

[RA06] **Rich C., Alexandru N. -** *An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics*, Appearing in Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, pp 1-12, 2006.

[SS14] **Shai S. S., Shai B. D. -** *Understanding Machine Learning*. New York, Cambridge University Press, 2014.

[TSK06] **Tan P.-N., Steinbach M., Kumar V. -** *Introduction to Data Mining*. Addison Wesley Publishing, 2006.

[XHS09] **Xhemali D., Hinde C. J., Stone R. G. -** *Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages*, International Journal of Computer Science Issue, Vol. 4(1), 2009.

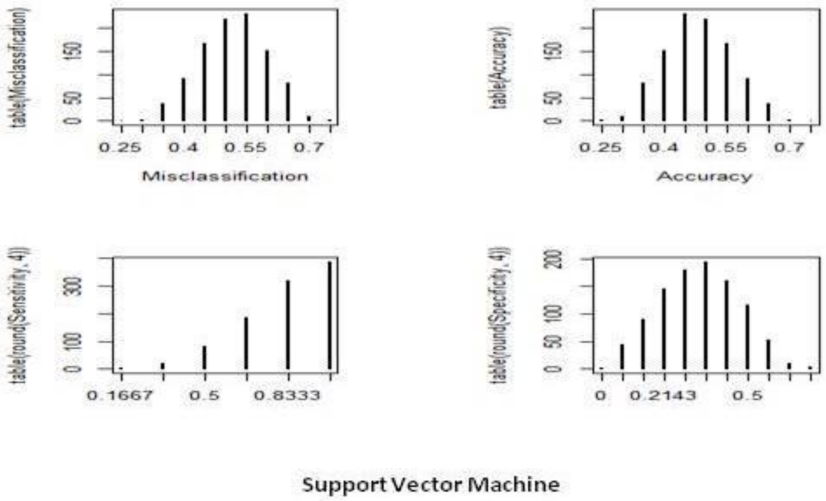

Figure 6. The bootstrap outcome of the misclassification error rate of the SVM classification algorithm

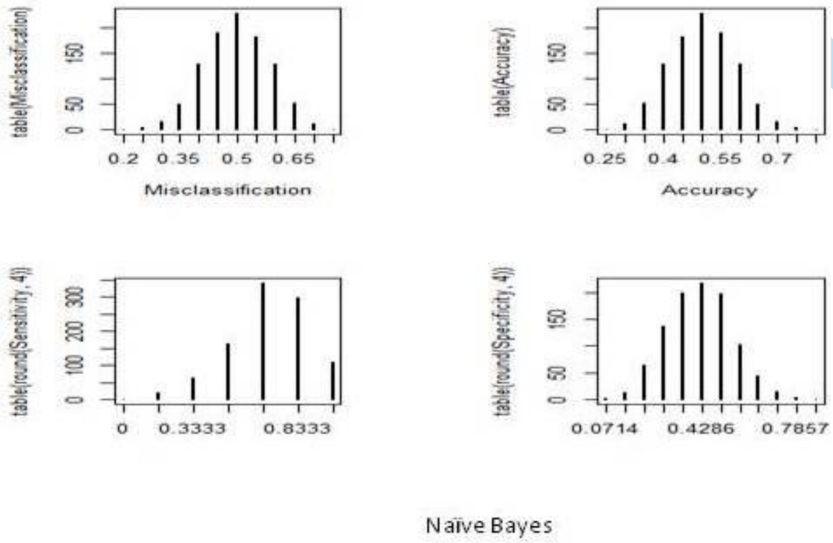

Figure 7. The bootstrap outcome of the misclassification error rate of the Naïve Bayes classification algorithm

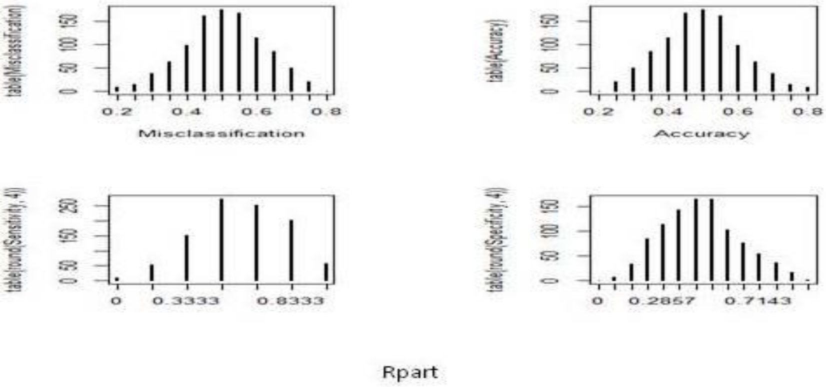

Figure 8. The bootstrap outcome of the misclassification error rate of the Rpart classification algorithm

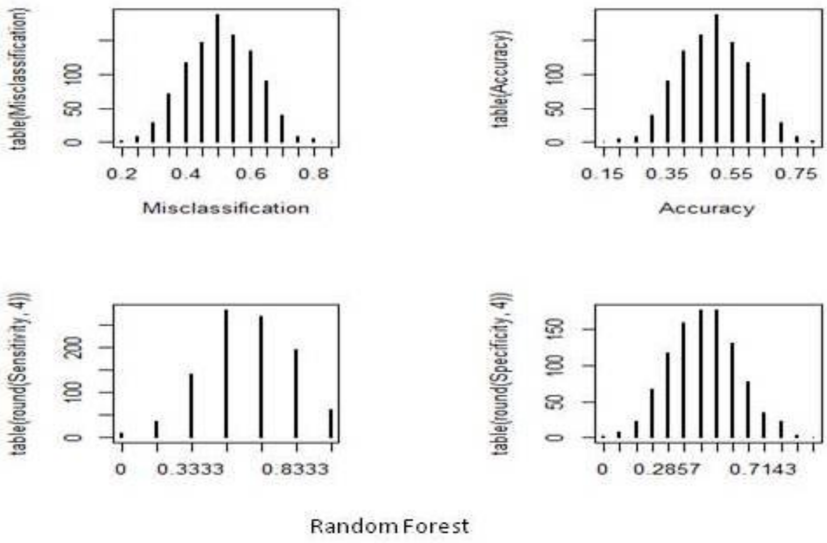

Figure 9. The bootstrap outcome of the misclassification error rate of the Random Forest classification algorithm

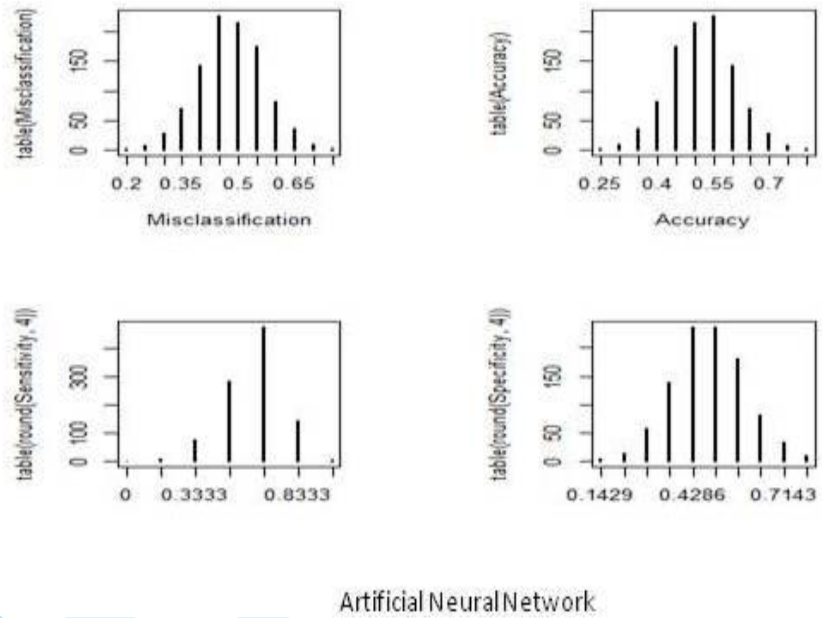

Figure 10. The bootstrap outcome of the misclassification error rate of the Artificial Neural Network classification algorithm

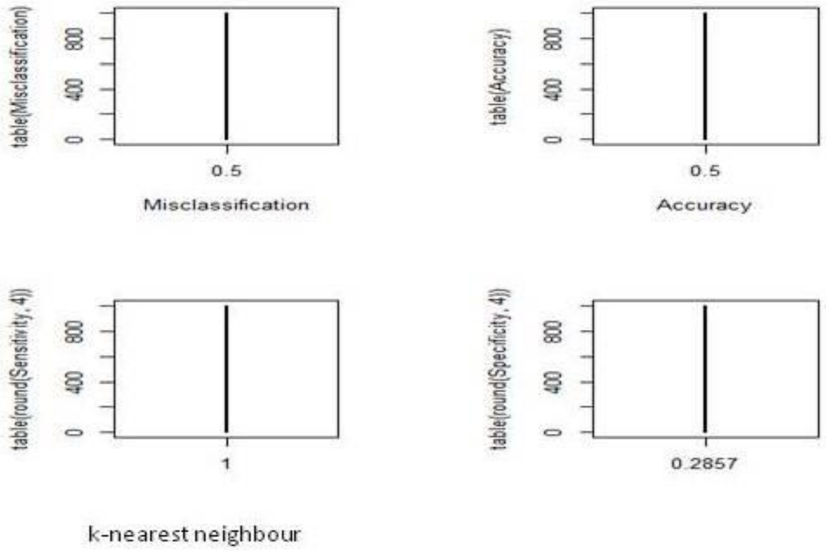

Figure 11. The bootstrap outcome of the misclassification error rate of the k-nearest classification algorithm