DE GRUYTER
OPEN

JACSM 2017, Vol. 9, No. 1, pp. 23 - 47
10.1515/jacsm-2017-0002

FRIEDMAN AND WILCOXON EVALUATIONS COMPARING SVM, BAGGING, BOOSTING, K-NN AND DECISION TREE CLASSIFIERS

Vinai George Biju[1], Prashanth CM[2]

[1] Department of Computer Science and Engineering
Christ University Faculty of Engineering, India
vinai.george@christuniversity.in

[2] Department of Computer Science and Engineering
Sapthgiri College of Engineering, India

Abstract

This paper describes a number of experiments to compare and validate the performance of machine learning classifiers. Creating machine learning models for data with wide varieties has huge applications in predictive modelling across multiple domain of science. This work reviews state of the art techniques in machine learning classifiers methods with several extent of magnitude in statistics and key findings that will be helpful in establishing best methodological practices for class predictions. Comprehensive comparative review analysis with statistical validations for various machine learning algorithm for SVM, Bagging, Boosting, Decision Trees and Nearest Neighborhood algorithm on multiple data sets is carried out. Focus on the statistical analysis of the results using Friedman-Test and Wilcoxon Test as well as other interpretative metrics like classification rate, ROC, F-measure are evaluated to benchmark results.

Key words: bagging, boosting, SVM, KNN, decision tree

1 Introduction

Given the different types of input instances with output labels, predicting the output using machine learning tasks has been challenging for quite some time. The newly developed machine learning methods follows a rigorous criterion of analysis against previous approaches to verify its correctness of predictions. The results rely on choosing possibilities between output cases and empirical comparisons measuring the performance derived from the configuration parameters of the experiments. In order to set up on a firm conclusion on a radical learning technique, the statistical validation of

produced results is a requisite in current times. Many approaches have been proposed in present-years contributing towards optimized and transformed features and there by using well known machine learning techniques with out assuming independence or relationships among attributes making interpretable, dense and accurate learning models. Classification is mostly beneficial when the examples collected in a database can be used as the foundation for making future decisions; e.g., for judging risks for credit, analysing scientific data and for diagnosis of diseases taking biological data. Scientists have established extensive variety of classification algorithms namely decision tree, nearest neighbor, support vector machines, boosting, and bagging.

The comparative study should perhaps be done with utmost significance using a statistically adequate background. Pattern recognition with enhanced feature selection assigning groups or classes to data instances could be executed for either models that are based on supervised classification or models that extract relationships between objects and its properties namely clustering or unsupervised classification.

Even though plenty of work can be found in literature that describes more appropriate classifiers for particular tasks, only limited studies reflect a more systematic statistical analysis with regards to their performance. The typical initial outcome of this work is to find the performance of various machine learning classifiers under various parameter settings taking detailed input values from multiple data sets.

Evaluating classifiers giving priority to maximum accuracy alone under different classifier parameters for specific tuned data and values is usually not the best approach, because for a different dataset the result would be different for most of the cases. Since the key study in this work is evaluation of practical results comparing classifiers, the outcome of classifiers with generative models are compared to the discriminative models. Specifically the effect of varied data sets on average classifier classification results performed with wide-ranging experiments are explored. The behavior of feature combination and class labels can be briefly explained using the framework with some of the machine learning techniques like kNN, SVM, Boosted and Bagged Trees. Data scientists typically investigate with different classifiers taking varied features and data sets to compare with specialized guidelines. It should be dealt with caution that the detailed experiments carried out, not applying specific statistical tests could lead to invalid inferences. The degree to which the contending classifiers, disagree or agree on output class values deliver evidence about reliability of classification output over perceived input data sets. The fraction of class instances that are positive and correctly predicted is indicated by classifier sensitivity and likewise specificity is the fraction of negative class instances that are correctly predicted [1].

Performances are evaluated for CHAID, neural network and logistic regression for imbalanced data set executed in an actual marketing application of a bank in [2]. The classifier performance for k-NN, Naive Bayes, SVM, LDA and Decision Tree are evaluated using characteristics including specificity, sensitivity, classification accuracy, computational time and kappa in [3]. Analysis of ROC towards results in machine learning, describing various challenges and providing concise substitute methods to ROC analysis like Lift chart, Calibration chart, Detection error trade-off curve had been discussed in [4]. Sentiment analysis and opinion mining for business analytics and market research scrutinizing word-of-mouth data for movie reviews are explored using support vector machines, neural network and bayesian decision tree in [5]. In [6] the influence of lexically normalized, naive, and semantic features on the performance of classifier for various diseases have been assessed using support vector machines. Statistical tests for evaluations of machine learning algorithms on several data sets using Wilcoxon signed ranks test and Friedman test is detailed in [7] The raisins superiority for agriculture is graded by means of machine learning techniques after selecting the best features using feature selection based on correlation in [8]. Data from wireless kinematic sensors for the job of physical movement recognition is taken for comparing the performance of AdaBoostM1 as the classifier of meta level with base level classifier C4.5 Graft in [9]. Investigating classifier performance with optimization to categorize non-randomized readings and classification of biomedical quotations for text selection using organized reviews are studied in [10]. Classifiers namely Support vector machines, Conditional Random fields and Latent Dynamic conditional random fields are compared for user intention understanding in analysing web search engines was shown in [11]. Soil profiles were analysed, sampled, selected and predicted for taxonomic soil class after investigating the classification power of data mining classifiers in [12]. Chi-Square Methods and R- Square techniques were used for high dimensional curve fitting using machine learning in [36]. A review was carried out for forecasting the share trading from the stock market database using state of the art machine learning in [35].

2 Support vector machines with kernel evaluation

To classify instances of two classes using SVM, the input data x is mapped to higher dimensional space geometry $S = \phi(x)$ and then devising an optimal hyperplane denoted by $w \cdot S - b = 0$ separating the two classes [13]. The function is expressed as:

$$f(x) = \langle w, \Phi(x) \rangle + b \tag{1}$$

which acts as decision boundary and is evaluated thereby using the function Φ that maps x to S space which is in higher dimension [14]. The distance is maximized for the set of data points $\Phi(x_k)$ that are consistent on the training set with hyperplane characterized by (w,b). The vector w is represented by: $w=\sum_{k=1}^{m}\alpha_k^* y_k \Phi(x_k)$ and the quadratic optimization problem:

$$\max_{\alpha} W(\alpha)=\sum_{k=1}^{m}\alpha_k-\frac{1}{2}\sum_{k,l}^{m}\alpha_k\alpha_l y_k y_l\left(K(x_k,x_l)+\frac{1}{C}\delta_{k,l}\right) \quad (2)$$

is solved through α_k^* [15].

Table 1. Classification Output Parametrics for Support Vector Machine.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	1.69	0.36	0.6	0.41	0.64	0.5	0.54	63.71
BreastCancer	0.11	0.3	0.55	0.67	0.7	0.68	0.63	69.58
ContactLenses	0.1	0.31	0.41	0.69	0.71	0.7	0.65	70.83
GermanCredit	1.54	0.25	0.5	0.74	0.75	0.74	0.68	75.1
PimaDiabetes	0.03	0.23	0.48	0.77	0.77	0.76	0.7	77.34
Glass	0.81	0.21	0.32	0.52	0.56	0.52	0.48	56.07
Hypothyroid	7.73	0.26	0.32	0.89	0.94	0.91	0.88	93.61
Ionosphere	0.36	0.11	0.34	0.89	0.89	0.88	0.83	88.6
Iris	0.13	0.23	0.29	0.96	0.96	0.96	0.94	96
Labor	0.21	0.11	0.32	0.89	0.9	0.89	0.85	89.47
Soybean	1.81	0.09	0.21	0.94	0.94	0.94	0.91	93.85
Vote	0.36	0.04	0.2	0.96	0.96	0.96	0.94	96.09
Weather	0.05	0.43	0.65	0.53	0.57	0.54	0.54	57.14
Segment	0.59	0.21	0.3	0.92	0.92	0.92	0.88	91.93
Weather	0.05	0.43	0.65	0.53	0.57	0.54	0.54	57.14
Segment	0.59	0.21	0.3	0.92	0.92	0.92	0.88	91.93

The solution to the above problem is established using the Lagrangian formulation and it is shown that $\sum_{k=1}^{m} y_k\alpha_k=0$ and $\forall_k, \alpha_k \geq 0$, where $\delta_{k,l}$ denotes Kronecker symbol with $K(x_k,x_l)=\langle\Phi(x_k),\Phi(x_l)\rangle$ representing the Gram matrix data set used for training. The predicted class label for each x can be computed after examining the sign of $f(x)$.

The mapped data $\Phi(x_k)$ could be contained in the smallest sphere of radius R. The radius margin bound $E \le 4R^2 \|w\|^2$ is evaluated to determine E i.e leave one out error bounds for SVMs. The distance S_p between a support vector $\Phi(x_p)$ that is mapped and the span of all other support vectors $E \le \sum_p \alpha_p^* S_p^2$ is used to devise methodically a tighter bound called span estimate. SVM with the variable S_p^2 and the quadratic slack variables ξ is dependent on $K_{sv} = \begin{pmatrix} K & 1 \\ 1^T & 0 \end{pmatrix}$ which is the the dot product between support vectors extended matrix by the equation $S_p^2 = 1/(K_{SV}^{-1})_{PP}$. We have made use of linear Kernel represented by $k(x^i, x^j) = \langle x^i \cdot x^j \rangle$ and quadratic kernel denoted $k(x^i, x^j) = (\langle x^i \cdot x^j \rangle + 1)^2$ for classifying instances using SVM.

If the number of instances are fewer than no of features representing the dimension space, it would result in an under par performance. It would definitely be an undetermined problem to find a hyperplane that fits the data in such cases. Then maximizing the margin with optimal parameters in SVM to find a solution will not be sufficient enough. Retaining only the features that are relevant, the dimensionality of the input space could be reduced [16].

The L1 soft-margin expression which is the fundamental problem for SVMs is solved by

$$\min \frac{1}{2} \| w \|^2 + C \sum_i \xi_k \text{ where } y_k (w \cdot z_k - b) \ge 1 - \xi_k, \xi_k \ge 0 \; \forall_k \tag{3}$$

This computational problem explained by its dual form through the kernel function implementing the non linear transformation.

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_k \sum_j \alpha_k \alpha_j y_k y_j k(x_k, x_j) \text{ where } 0 \le \alpha_k \le C \; \forall_k \sum_k y_k \alpha_k = 0 \text{ where } k(x_k, x_j) = \phi(x_k) \cdot \phi(x_j) \tag{4}$$

Gaussian kernel represented by $k(x_k, x_j) = \exp\left(-\frac{\|x_k - x_j\|^2}{2\sigma^2} \right)$ and Polynomial kernel denoted by $k(x_k, x_j) = (1 + x_k \cdot x_j)^d$ are other popular kernel functions used in this paper.

Table 2. Classification Output Parametrics for Decision Stump.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	0.13	0.42	0.46	0.68	0.64	0.65	0.64	64.4
BreastCancer	0.05	0.38	0.44	0.68	0.69	0.68	0.63	68.53
ContactLenses	0.01	0.23	0.36	0.71	0.71	0.68	0.71	70.83
GermanCredit	0.08	0.38	0.43	0.49	0.7	0.58	0.68	70
PimaDiabetes	0.05	0.38	0.44	0.72	0.72	0.72	0.68	71.88
Glass	0.01	0.18	0.3	0.21	0.45	0.28	0.34	44.86
Hypothyroid	0.21	0.03	0.12	0.95	0.95	0.95	0.95	95.39
Ionosphere	0.15	0.27	0.37	0.86	0.83	0.81	0.75	82.62
Iris	0.01	0.22	0.33	0.5	0.67	0.56	0.67	66.67
Labor	0.01	0.21	0.34	0.81	0.81	0.8	0.84	80.7
Soybean	0.01	0.08	0.2	0.13	0.28	0.16	0.21	27.96
Vote	0.01	0.08	0.2	0.96	0.96	0.96	0.93	95.63
Weather	0.02	0.49	0.59	0.29	0.29	0.29	0.58	28.57
Segment	0.11	0.21	0.32	0.11	0.3	0.16	0.28	30.4

3 DECISION TREES FOR INDEPENDENT OBSERVATIONS

A decision tree is a directed acyclic graph form of tree classifier. There is no incoming edges for the root of the tree and every internal node have outgoing edges with an incoming edge [17]. We apply binary decision trees in this study so that every node has outgoing edges either with number zero or two. The *leafnode* does not have any outgoing edges and is labeled with a class label. The splitting attribute X_n or predictor attribute is associated with each internal node. If X_n denotes a numerical attribute, then q_n which is the splitting predicate holds the form $X_n \le x_n$ and $x_n \in dom(X_n)$ where x_n is called the split point of node n. If X_n denotes a categorical attribute, then q_n holds in the form $X_n \in J_n$ where $J_n \subset dom(X_n)$ and J_n represents the splitting subset at the node n [18]. A classification tree is typically built using training data in two phases namely growing phase and pruning phase. The split selection techniques producing binary splits at each node is usually established on impurity-based method [19]. The problem of decision tree induction formally giving background terminology is indicated as follows: Let random variables be represented as $X_1,...,X_m$, C. The domain of X_i is

denoted as $dom(X_i)$ and $dom(C) = \{1, 2, ..., k\}$. The decision Tree classifier is represented as a function $d: dom(X_1) \times ... \times dom(X_m)$ a $dom(C)$. Let the probability distribution be represented as $P(X', C')$ and a random record $t = \langle t.X_1, ...t.X_m, t.C \rangle$ be drawn from P where $\langle t.X_1, ...t.X_m \rangle \in X'$ and $t.C \in C'$ [20]

Decision tree learning induction using complete observations is as follows: For each data point and its neighbors $x_i, i = 1, ..., k$, along with a ranking associated as $\sigma_i \in \Omega$, the probability distribution is denoted as $P(.|x)$ on Ω which is locally constant. Since the observations are assumed to be independent and $\sigma = \{\sigma_1, ...\sigma_k\}$ with the parameters (θ, π), the probability is observed as

$$P(\sigma | \theta, \pi) = \prod_{i=1}^{k} \frac{\exp - \theta D(\sigma_i, \pi)}{\phi(\theta)} \tag{5}$$

Table 3. Classification Output Parametrics for J48.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	0.28	0.46	0.48	0.41	0.64	0.5	0.54	63.71
BreastCancer	0.11	0.37	0.43	0.75	0.76	0.71	0.65	75.52
ContactLenses	0.08	0.15	0.32	0.85	0.83	0.84	0.81	83.33
GermanCredit	0.39	0.35	0.48	0.69	0.71	0.69	0.66	70.5
PimaDiabetes	0.28	0.32	0.45	0.74	0.74	0.74	0.73	73.83
Glass	0.03	0.1	0.29	0.67	0.67	0.67	0.61	66.82
Hypothyroid	0.58	0	0.04	1	1	1	1	99.58
Ionosphere	0.37	0.09	0.29	0.92	0.92	0.91	0.88	91.45
Iris	0.06	0.04	0.16	0.96	0.96	0.96	0.92	96
Labor	0.05	0.32	0.47	0.75	0.74	0.74	0.68	73.68
Soybean	0.31	0.01	0.08	0.92	0.92	0.91	0.92	91.51
Vote	0.17	0.06	0.17	0.96	0.96	0.96	0.96	96.32
Weather	0.01	0.29	0.48	0.63	0.64	0.63	0.81	64.29
Segment	0.4	0.01	0.11	0.96	0.96	0.96	0.95	95.73

The parameter (θ, π) has the maximum likelihood estimation for π given as $\hat{\pi} = \arg\min_{\pi} \sum_{i=1}^{k} D(\sigma_i, \pi)$ [21].

4 Aggregated bagging for bootstrap samples

Classifier optimization worked over estimation of error rate and model selection while learning from sample data sets can conclude in bias and over fitting [22]. This could result in an unstable classification model being generated and could be improved by the aggregation of classifiers. Bagged classification trees could solve to reduce misclassification error substantially in most of the applications and bench mark problems [23]

Table 4. Classification Output Parametrics for Bagging.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	2.33	0.46	0.48	0.41	0.64	0.5	0.54	63.71
BreastCancer	0.09	0.38	0.45	0.64	0.69	0.64	0.69	69.23
ContactLenses	0.03	0.31	0.4	0.53	0.58	0.55	0.77	58.33
GermanCredit	0.28	0.33	0.42	0.73	0.75	0.73	0.77	74.7
PimaDiabetes	0.51	0.32	0.41	0.75	0.76	0.75	0.81	75.78
Glass	0.03	0.12	0.24	0.71	0.72	0.71	0.76	72.43
Hypothyroid	1.4	0	0.05	1	1	1	1	99.52
Ionosphere	0.47	0.14	0.26	0.91	0.91	0.91	0.95	91.17
Iris	0.09	0.05	0.17	0.94	0.94	0.94	0.98	94
Labor	0.29	0.3	0.38	0.84	0.84	0.84	0.86	84.21
Soybean	0.39	0.03	0.11	0.84	0.86	0.84	0.92	85.65
Vote	0.3	0.07	0.17	0.96	0.96	0.96	0.98	95.63
Weather	0.02	0.53	0.56	0.38	0.5	0.43	0.44	50
Segment	0.7	0.02	0.1	0.96	0.96	0.96	0.99	95.87

Let $L = \{(x_k, y_k), k = 1,..., N\}$ denotes learning from N observations of independent sample that comprise of predictors which are $q-$ dimensional vectors denoted by $x_k = (x_{k1},..., x_{kp}) \in R^P$. The learning sample have observations assumed to be identical distributed and random variables that are independent with a distinct distribution function F_L where $(x_1, y_1),..., (x_N,..., y_N) \sim F_L$. The class denoted y-values for the subsequent data is predicted by the classifier $C(\%\wp L)$ from a set of vector of parameters $\%\wp$ established through the learning sample L [24].

Let the distribution be denoted by $F_{\%\%}$ for future observations represented by $(\%,\%)$. To maintain stability for the classifiers C over averaged multiple learning samples, classifier C_A is aggregated for the observation $\%$ and is illustrated as $C_A(\%) = E_{FL}C(\%,L)$. The learning samples L and its expectation is distributed accordingly as F_L.

The aggregated rule $C_A(\%)$ applying bootstrap as shown by $\hat{C}_A(\%) = E_{FL}C(\%,L^*)$, is measured by bagging where L^* denotes a random sample from the formulated distribution evaluated from samples and is denoted by the function $F_L.(x_1^*, y_1^*),...,(x_N^*, y_N^*) \sim \hat{F}_L^{\%}$.

Based on B the bootstrap samples, the bagged classifier $\hat{C}_A^B$ is computed as follows. Initially B samples L of size N are drawn randomly $L^{*(1)},...,L^{*(B)}$ with replacement.

The iterative algorithm for Bagging is shown as follows:

1. The bootstrap sample $L^{*(b)}$ is used to create the classifier C.
2. Classifier model is constructed iteratively for all bootstrap samples $b = 1,...,B$.
3. A new instance $\%$ is classified as,

$$\hat{C}_A^B = \arg\max_{j\in\{1,2\}} \sum_{b=1}^{B} \chi_{\{j\}}(C(\%, L^{*(b)})) \tag{6}$$

we apply majority voting where χ is the indicator function

$$\chi_Z(x) = \begin{cases} 1 & x \in Z \\ 0 & else \end{cases} \tag{7}$$

To summarize performance of the bagged trees with smaller number of splits with smaller node size is found to be better in some data distributions than maximal unpruned trees and that the application requires careful tuning of the relevant classifier parameters while applying bagging [25].

5 Combining hypothesis with boosting

In boosting the weak rules or hypotheses which are moderately accurate are combined to design a classification rule that are highly accurate [26]. A single rule combined hypothesis is then linearly combined from these weak hypotheses. The predictive model function denoted by $f : \chi \to R$ is designed so that for example x and $f(x)$, the sign illustrated as (-1 or +1) indicates the predicted class and the magnitude $|f(x)|$ is evaluated as the confidence measure while creating a predictive model for learning [27].

The training sample S represented by: $S = \{(x_i, y_i)\}_{i=1}^{m}$ contains input features x and output label y.

Let D_1 be the distribution and is initialized for all $D_1(i) = 1/m$, $\forall i\, 1 \le i \le m$ .The weak hypothesis $h_t : \chi \to R$ is found and later $\alpha_t \in R$ is chosen to update the distribution $D_t, \forall i, 1 \le i \le m$ and $D_{t+1}(i) = \frac{D_1(i)\exp(-\alpha_t y_i h_t(x_i)}{Z_t}$. Here Z_t is selected with distribution D_{t+1}.

Finally the hypothesis is combined and returned as $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ [28].

Table 5. Classification Output Parametrics for LogitBoost.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	1.16	0.31	0.4	0.76	0.76	0.76	0.81	76.03
BreastCancer	0.09	0.36	0.44	0.7	0.72	0.71	0.72	72.38
ContactLenses	0.05	0.19	0.37	0.75	0.75	0.75	0.84	75
GermanCredit	0.22	0.36	0.43	0.68	0.71	0.68	0.75	70.8
PimaDiabetes	0.31	0.31	0.41	0.73	0.74	0.74	0.81	74.09
Glass	0.09	0.1	0.24	0.71	0.72	0.7	0.75	71.5
Hypothyroid	1.49	0.01	0.04	1	1	1	1	99.58
Ionosphere	0.23	0.14	0.28	0.91	0.91	0.91	0.95	91.17
Iris	0.2	0.05	0.18	0.94	0.94	0.94	0.94	94
Labor	0.06	0.15	0.31	0.89	0.9	0.89	0.91	89.47
Soybean	0.61	0.01	0.07	0.93	0.93	0.93	0.97	92.97
Vote	0.23	0.06	0.18	0.96	0.95	0.95	0.99	95.4
Weather	0.01	0.46	0.6	0.38	0.5	0.43	0.57	50
Segment	1.27	0.02	0.1	0.96	0.96	0.96	0.99	95.93

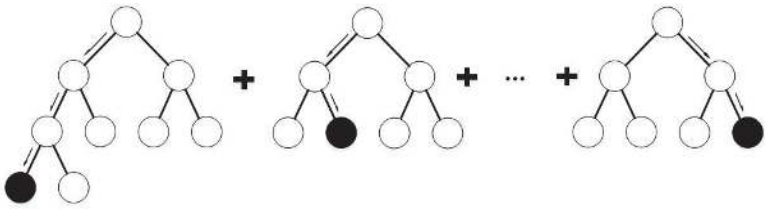

Figure 1. Boosted Tree Ensemble

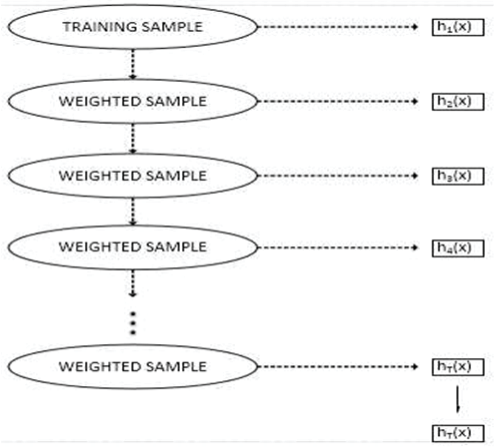

Figure 2. Weighted Boosting

Let the subset X_1 and X_0 be the examples for which p predicate holds true and does not hold true respectively. If π holds true, then $\left[\left[\pi\right]\right]$ be 1 corresponding to that predicate π and 0 otherwise.

The following values for W_b^j is evaluated when $b \in \{+1, -1\}$ and $j \in \{0,1\}$ for D_t, which represents the current distribution.

$$W_b^j = \sum_{i=1}^{m} D_t(i)\left[\left[x_i \in X_j \wedge y_i = b\right]\right] \tag{8}$$

Table 6. Classification Output Parametrics for AdaBoost.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	1.42	0.33	0.41	0.74	0.75	0.74	0.79	74.86
BreastCancer	0.42	0.35	0.43	0.69	0.7	0.7	0.73	70.28
ContactLenses	0.12	0.36	0.41	0.72	0.71	0.69	0.7	70.83
GermanCredit	0.23	0.36	0.43	0.66	0.7	0.67	0.74	69.5
PimaDiabetes	0.2	0.31	0.42	0.74	0.74	0.74	0.8	74.35
Glass	0.01	0.18	0.3	0.21	0.45	0.28	0.34	44.86
Hypothyroid	0.37	0.03	0.12	0.91	0.93	0.92	0.97	93.21
Ionosphere	0.22	0.16	0.27	0.92	0.91	0.91	0.94	90.88
Iris	0.12	0.07	0.17	0.95	0.95	0.95	0.94	95.33
Labor	0.28	0.15	0.34	0.88	0.88	0.88	0.87	87.72
Soybean	0.01	0.08	0.2	0.13	0.28	0.16	0.21	27.96
Vote	0.23	0.06	0.19	0.95	0.95	0.95	0.99	95.4
Weather	0.02	0.49	0.63	0.53	0.57	0.54	0.52	57.14
Segment	0.03	0.21	0.32	0.11	0.3	0.16	0.28	30.4

W_b^j represents the weight of class b for the examples used for training in partition X_j which follows the distribution D_t. Setting $\alpha_t = 1$ and choosing $c_j = \frac{1}{2}\ln\left(\frac{W_{+1}^j}{W_{-1}^j}\right)$, the value for Z_t is minimized for a certain predicate. This background indicates that $Z_t = 2\sum_{j\in\{0,1\}}\sqrt{W_{+1}^j\, W_{-1}^j}$ and shows that for boosting's generalization error derives an upper bound

$$\Pr[H(x) \neq y] \leq \Pr[H(x) \neq y] + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right) \tag{9}$$

The training examples are assumed to be generated over the probability distribution Pr[·] and the training sample generated over empirical probability distribution is denoted by Pr[·].

Though the bound for d which is the space of VC-dimension for all likely base classifiers convert to be very feeble as the rounds T increases, prediction using AdaBoost will rapidly overfit with number of rounds which is usually moderate. Overfitting normally does not happen on the training examples by the notion of margins in the case of boosting. The margin $(x,y) = \frac{y\sum_t \alpha_t h_t(x)}{\sum_t \alpha_t}$

for the example (x, y) is based on the votes $h_t(x)$ along with α_t denoting the weights for all hypotheses [29].

The power of settlement for the base classifiers is indicated by the magnitude of the margin and the correct prediction combining votes is indicated by the sign it produces. The number of boosting rounds is independent on the bound and that the generalization error θ is maximum for the case as shown as:

$$\Pr[margin(x, y) \le \theta] + \tilde{O}\left(\sqrt{\frac{Td}{m\theta^2}}\right) \tag{10}$$

Table 7. Classification Output Parametrics for K Nearest Neighbor.

DataSet	Time	MAE	RMS	Prc	Rec	Fm	PRC	Class%
Supermarket	0.02	0.62	0.78	0.69	0.37	0.21	0.53	37.13
BreastCancer	0.02	0.33	0.51	0.7	0.72	0.7	0.69	72.38
ContactLenses	0.01	0.23	0.32	0.8	0.79	0.8	0.89	79.17
GermanCredit	0.02	0.28	0.53	0.72	0.72	0.72	0.67	72
PimaDiabetes	0.02	0.3	0.55	0.7	0.7	0.7	0.64	70.18
Glass	0.01	0.09	0.29	0.71	0.71	0.7	0.6	70.56
Hypothyroid	0.03	0.04	0.21	0.91	0.92	0.91	0.9	91.52
Ionosphere	0.01	0.14	0.37	0.87	0.86	0.86	0.81	86.32
Iris	0.01	0.04	0.17	0.95	0.95	0.95	0.93	95.33
Labor	0.01	0.19	0.41	0.83	0.83	0.83	0.79	82.46
Soybean	0.01	0.01	0.09	0.92	0.91	0.91	0.91	91.22
Vote	0.01	0.07	0.24	0.93	0.92	0.93	0.96	92.41
Weather	0.01	0.25	0.43	0.8	0.79	0.79	0.78	78.57
Segment	0.01	0.01	0.1	0.96	0.96	0.96	0.93	96.2

6 K-Nearest Neighbour with cost-distance metrics

The k-Nearest-Neighbours (kNN) is a effective but simple non-parametric classification technique. For classification of a data record t, a neighbourhood is formulated from its nearest k neighbours and retrieved using distance measure metric[30]. Considering weight-based distance, the class label for t is decided usually among the data records with majority voting in the neighbourhood of t [31].

Applying kNN requires choosing a suitable value for k, and the feat of classification is greatly dependent on the value of k. Since the kNN method is influenced by k and out of several ways of selecting the k value, a modest way is to execute the algorithm for several epochs with diverse k values and the one which supports the finest performance is chosen. In direction to kNN being not to be too much dependent on the selection of k, it is pre-eminent to observe sets of multiple nearest neighbours than rather just few k-nearest neighbour sets [32].

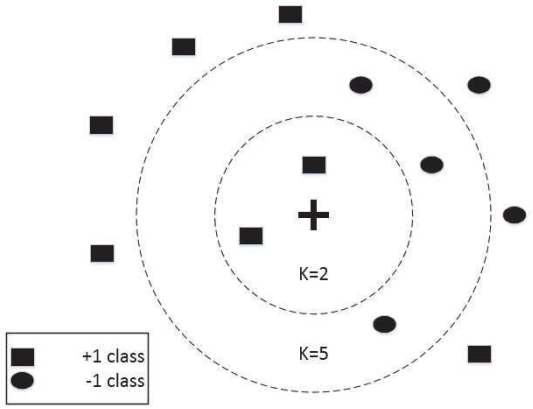

Figure 3. K-Nearest Neighbour

The extreme cost of kNN for classifying novel instances is mainly due to the reason that almost all computation happens during classification time rather than the training examples when first come across. To relieve the problem of heavy cost incurred storing the entire training set when it is very large, recent studies have attempted to eliminate the redundancy of the training set applied to k-Nearest-Neighbours classifier. kNN preserves the entire training data for classification and is a learning method which is case-based. Let $< Sim(d_i), Cls(d_i), Rp(d_i), Num(d_i) >$ represents the lower bound similarity of d_i to data values enclosed by N_i the class label of d_i, an illustration of d_i to itself and the data tuples enclosed by N_i respectively for the the model created M. The $Sim(d_i)$ value with minimum value is chosen, viz. representative with maximum density if equivalent maximal number of neighbours exist for more than one neighbour.

The classification algorithm is illustrated as follows:

1. For classification of a novel data tuple d_t, the similarity to every representation point in the model M is evaluated.

2. If only one representation point is covered for $d_t, < Sim(d_i), Cls(d_i), Rp(d_i), Num(d_i) >, d_t$ is classified as the grouping of d_j as followed by the Euclidean distance of d_j to d_t has a value less than Sim(d_j)
3. d_t is classified to be the category of the grouping with highest Num(d_j) if d_t is covered by minimum two symbolic diverse category, following the neighbourhood spanning the highest number of data tuples among the dataset used for training .
4. d_t is classified to be the category of grouping in which the boundary is nearest to d_t if there is no grouping in the model M that covers d_t.

The Euclidean distance of d_t to d_i subtracted with Sim(d_i) indicates the Euclidean distance of d_t to d_i represented with the closest boundary [33].

Let $\{(\vec{x}_i, y_i)\}_{i=1}^n$ denote data used for training with *n* examples labeled using inputs $\vec{x}_i \in R^d$ and class labels usually discrete y_i. The binary input matrix is used $y_{ij} \in \{0,1\}$ to show that the labels y_i and y_j match or otherwise. The goal learns a linear transformation which could be used to find squared distances: $D(\vec{x}_i, \vec{x}_j) = \| L(\vec{x}_i, \vec{x}_j) \|^2$. The cost function parameterized and concluding the distance metrics has significant terms penalizing heavy distances between each its target neighbors and input, while the other term penalizes minor distances between every inputs that does not form similar label and each input [34].

Precisely, the cost function is computed as:

$$y_t = \arg\min_{yt} \sum_j \eta_{tj} \| L(\overline{x_i}, \overline{x_j}) \|^2 + c \sum_{j,i=t \vee l=t} \eta_{tj}(1-y_{il})[1 + \| L(\overline{x_i}, \overline{x_j}) \|^2 - \| L(\overline{x_i}, \overline{x_l}) \|^2] \quad (11)$$

7 Statistical significance using Friedman test and Wilcoxon test

Let *S* and *L* be the number of +ve class and -ve class in the data set, respectively; let S_+ denote the number of +ve classes that are correctly classified by a system, and S_- the number of +ve classes misclassified as -ve class. In the same way, let L_+ and L_- be the number of -ve classes classified by a system as +ve class and -ve class, respectively. These four values form a contingency table which summarizes the behavior of a system. The widely-used measures precision (*p)*, recall (*r*) and F_β are defined as follows:

$$p = \frac{S_+}{S_+ + L_+} \quad r = \frac{S_+}{S_+ + S_-} \quad F_\beta = \frac{(\beta^2 + 1)pr}{\beta^2 p + r} \tag{12}$$

We apply Friedman Test when we cannot assume that the data from each of groups are normally distributed populations. Blocks of data are assumed to be independent and the underlying variable in the data are mostly numeric in nature. When compared to F test, the Friedman rank test makes less stringent assumptions.The Friedman rank test concludes that the populations differs atleast from one of the other populations in variation, central tendency and shape. Friedman rank test also concludes if the input data groups have been generated from the whole original data set with the medians.

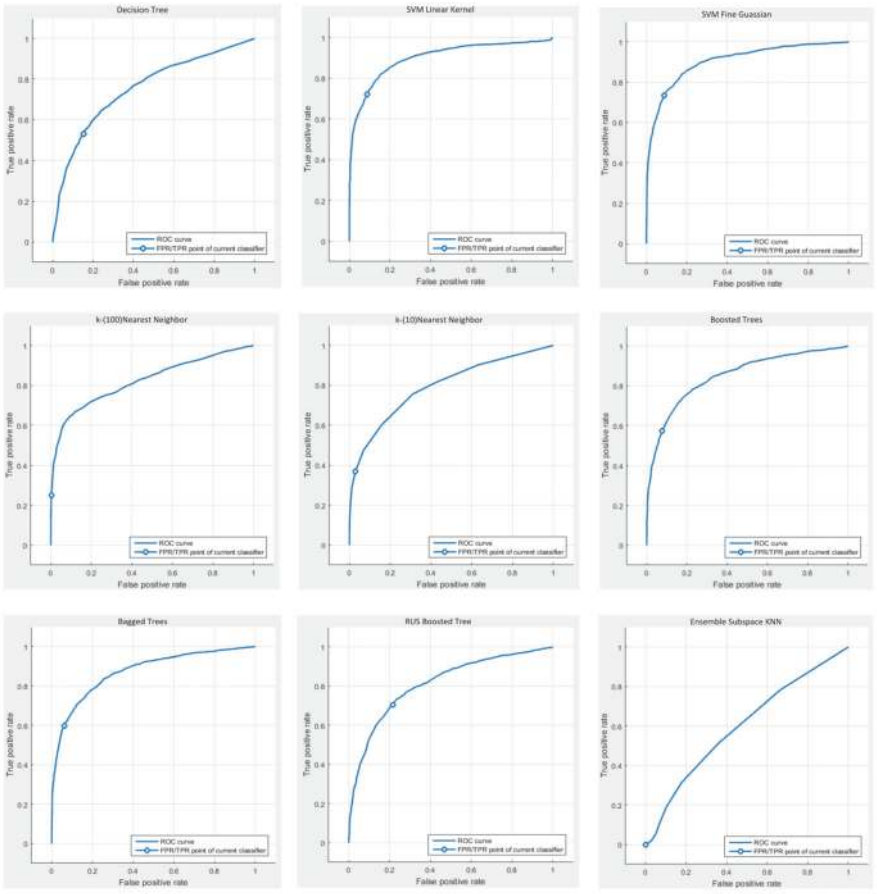

Figure 4. ROC evaluation for Decision Tree, SVM, KNN, Boosted and Bagged Tree Variants for Sparse Super Market Data.

Table 8. Friedman Test on Classifier Results.

N=14	Mean	StdD	Min	Max	FrdMR
SVM	79.95	14.63	56.07	96.09	4.68
KNN	79.67	15.45	37.13	96.2	3.75
DecisionStump	64.18	22.93	27.96	95.63	2.21
J48	81.59	13.2	63.71	99.58	5
Bagging	79.3	15.43	50	99.52	4.07
Adaboost	70.2	22.79	27.96	95.4	3.29
Logitboost	82.02	14.11	50	99.58	5
Bagging	79.3	15.43	50	99.52	4.07
Adaboost	70.2	22.79	27.96	95.4	3.29
Logitboost	82.02	14.11	50	99.58	5

Figure 5. Confusion Matrix evaluation for Decision Tree, SVM, KNN, Boosted and Bagged Tree Variants for Sparse Super Market Data.

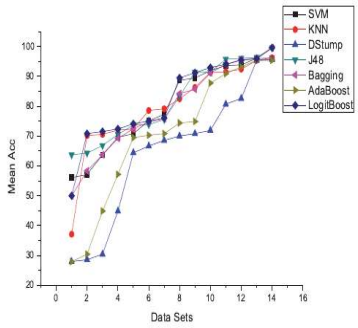

Figure 6. Classifier Mean Accuracy

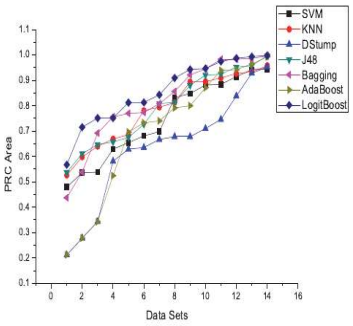

Figure 7. Classifier PRC Area

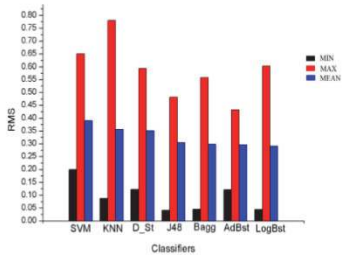

Figure 8. Classifier RMS

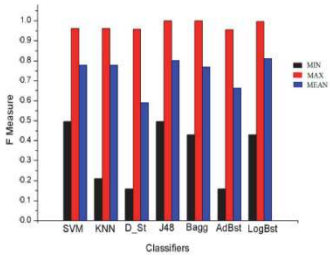

Figure 9. Classifier F-Measure

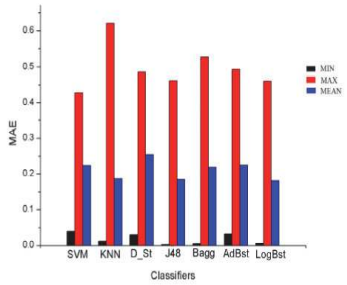

Figure 10. Classifier MAE

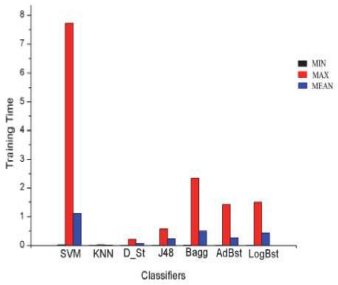

Figure 11. Classifier Training Time

The Classifier results are analysed using Friedman test under the following assumptions: One Data set is evaluated on three or more different classifiers. Training/Test set is generated as random sample from the population. Class/outcome variable is measured at the continuous or ordinal level. Samples are not necessarily normally distributed. The Wilcoxon signed rank test evaluates samples having size n greater than 10 observations and is evalu-

ated in pair of samples. Since W statistics is a non-parametric test, the multivariate normality is not essential to be assumed for the data. The Wilcoxon Signed Rank procedure evaluates under the illusion that the sample holds a frequency distribution that is symmetric and is from random population. The assumption which is symmetric never promises normality, as it is observed to have approximately the equal number of data points below and above the median.

The Wilcoxon technique evaluates a statistic for testing that is matched to an expected value. It is evaluated by summation of differences which is ranked along the deviation of every variable from a median. The Wilcoxon sign test compares the two dependent observations and quantifies the number of positive and negative differences.

Table 9. Wilcoxon Signed Ranks Test for Classifier Comparison.

Comparison	Ranking	Instances	MeanRank	SumOfRanks
J48-DecisionStump	Negative Ranks	2	5	10
	Positive Ranks	12	7.92	95
	Ties	0		
Bagging-DecisionStump	Negative Ranks	2	4.5	9
	Positive Ranks	11	7.45	82
	Ties	1		
Adaboost-DecisionStump	Negative Ranks	3	2.33	7
	Positive Ranks	7	6.86	48
	Ties	4		
Logitboost-DecisionStump	Negative Ranks	1	1	1
	Positive Ranks	13	8	104
	Ties	0		
Bagging-J48	Negative Ranks	8	7.25	58
	Positive Ranks	5	6.6	33
	Ties	1		
Logitboost-J48	Negative Ranks	6	7.5	45
	Positive Ranks	7	6.57	46
	Ties	1		
Adaboost-Bagging	Negative Ranks	8	7.75	62
	Positive Ranks	6	7.17	43
	Ties	0		

Logitboost-Bagging	Negative Ranks	4	4.75	19
	Positive Ranks	7	6.71	47
	Ties	3		
Logitboost-Adaboost	Negative Ranks	3	5.33	16
	Positive Ranks	10	7.5	75
	Ties	1		
KNN-SVM	Negative Ranks	9	6.22	56
	Positive Ranks	5	9.8	49
	Ties	0		
DecisionStump-SVM	Negative Ranks	11	7.73	85
	Positive Ranks	2	3	6
	Ties	0		
J48-SVM	Negative Ranks	4	6	24
	Positive Ranks	8	6.75	54
	Ties	2		
Bagging-SVM	Negative Ranks	9	6.22	56
	Positive Ranks	4	8.75	35
	Ties	1		
Adaboost-SVM	Negative Ranks	9	6.56	59
	Positive Ranks	3	6.33	19
	Ties	2		
Logitboost-SVM	Negative Ranks	6	5.33	32
	Positive Ranks	7	8.43	59
	Ties	1		
DecisionStump-KNN	Negative Ranks	10	8.3	83
	Positive Ranks	4	5.5	22
	Ties	0		
J48-KNN	Negative Ranks	5	7.6	38
	Positive Ranks	9	7.44	67
	Ties	0		
Bagging-KNN	Negative Ranks	6	7.33	44
	Positive Ranks	8	7.63	61
	Ties	0		

Adaboost-KNN	Negative Ranks	7	8.14	57
	Positive Ranks	6	5.67	34
	Ties	1		
Logitboost-KNN	Negative Ranks	5	6.6	33
	Positive Ranks	9	8	72
	Ties	1		

Table 10. Z Score and Significance on Wilcoxon Test

	KNN-SVM	DecStmp-SVM	J48-SVM	Bagg-SVM
Z	-0.22	-2.76	-1.177	-0.734
Asy.Sig	0.826	0.006	0.239	0.463
	Adbst-SVM	Logbst-SVM	DecStmp-KNN	J48-KNN
Z	-1.569	-0.943	-1.915	-0.91
Asy.Sig	0.117	0.345	0.056	0.363
	Bagg-KNN	Adbst-KNN	Logbst-KNN	J48-DecStmp
Z	-0.534	-0.804	-1.224	-2.668
Asy.Sig	0.594	0.422	0.221	0.008
	Bagg-DecStmp	Adbst-DecStmp	Logbst-DecStmp	Bagg-J48
Z	-2.551	-2.09	-3.233	-0.874
Asy.Sig	0.011	0.037	0.001	0.382
	Adbst-J48	Logbst-J48	Logbst-Bagg	Logbst-Adbst
Z	-1.977	-0.035	-1.245	-2.062
Asy.Sig	0.048	0.972	0.213	0.039

The significance is tested using the standard normal distributed z-value as shown in table 9 and table 10. The null hypothesis states that the median difference between pairs of classifier accuracy is zero. The null hypothesis is rejected when the significant value is less than 0.05 indicating one of the classifier outperforms the other. Here from table 11, Asy.Sig value of 0.826 indicates to accept the null hypothesis for KNN and SVM and Asy.Sig value of 0.006 shows that SVM and Decision Stump has statistically significant differences comparing the mean accuracy.

Conclusion

This work reviewed to assess various classification based machine learning techniques and investigated statistical evaluation measures to compare the results. Techniques for comparison and verification of classification results are Support Vector Machines, K-Nearest Neighbor, Decision Stump, J48, Bagging, Logitboost and Adaboost. MAE, RMS, Precision, Recall, F-Measure, PRC-Area and Accuracy was considered for comparison of classifiers. Comparison of classifiers was executed using Weka 3 open source machine learning software and MATLAB 2016. Friedman and Wilcoxon test was executed using IBM SPSS and Data sets were taken from UCI machine learning repository. The statistical techniques used for validation of results are Friedman Test and Wilcoxon Signed Rank Test in non-parametric setting. Classification performance using SVM under Linear Kernel and Fine guassian framework was found much better than other classifiers for Sparse Supermarket Data. When the classifiers were compared with multiple data sets like Iris, Labor, Vote, German Credit, Breast Cancer, Glass and many others, Friedman Mean Rank was found high for J48 and LogitBoost. Pair wise comparison with statistical significance was evaluated using Wilcoxon Signed Ranks Test.

References

1. Labatut, Vincent, and Hocine Cherifi "Accuracy measures for the comparison of classifiers". *arXiv preprint arXiv*, 1207.3790, 2012.
2. Duman, Ekrem, Yeliz Ekinci, and Aydin Tanriverdi "Comparing alternative classifiers for database marketing: The case of imbalanced datasets". *Expert Systems with Applications,* 39.1, pp.48-53, 2012.
3. Aydemir, Onder, and Temel Kayikcioglu "Comparing common machine learning classifiers in low-dimensional feature vectors for brain computer interface applications". *International Journal of Innovative Computing, Information and Control* 9.3, pp.1145-1157, 2013.
4. Majnik, Matjaz, and Zoran Bosnic "ROC analysis of classifiers in machine learning: A survey". *Intelligent data analysis*, 17.3, pp.531-558, 2013.
5. Kim, Yoosin, Do Young Kwon, and Seung Ryul Jeong "Comparing machine learning classifiers for movie WOM opinion mining". *KSII Transactions on Internet and Information Systems* 9.8, pp.3178-3190, 2015.
6. Kotfila, Christopher, and Ozlem Uzuner "A systematic comparison of feature space effects on disease classifier performance for phenotype identification of five diseases". *Journal of biomedical informatics* 58, S92-S102, 2015.
7. Demsar, Janez "Statistical comparisons of classifiers over multiple data sets". *Journal of Machine learning research* 7.Jan, pp.1-30, 2006.

8. Mollazade, Kaveh, Mahmoud Omid, and Arman Arefi "Comparing data mining classifiers for grading raisins based on visual features". *Computers and electronics in agriculture* 84, pp.124-131, 2012.
9. Dalton, Anthony, and Gearoid OLaighin "Comparing supervised learning techniques on the task of physical activity recognition". *IEEE journal of biomedical and health informatics* 17.1, pp.46-52, 2013.
10. Bekhuis, Tanja, and Dina Demner-Fushman "Screening nonrandomized studies for medical systematic reviews: a comparative study of classifiers. Artificial intelligence in medicine 55.3, pp.197-207, 2012.
11. Deufemia, Vincenzo, et al. "Comparing classifiers for web user intent understanding". *Empowering Organizations. Springer International Publishing*, pp.147-159, 2016.
12. Taghizadeh-Mehrjardi, R., et al. "Comparing data mining classifiers to predict spatial distribution of USDA-family soil groups in Baneh region", Iran. *Geoderma* 253: 67-77, 2015.
13. Orru, Graziella, et al "Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review". *Neuroscience and Biobehavioral Reviews* 36.4, pp.1140-1152, .2012.
14. Qi, Zhiquan, Yingjie Tian, and Yong Shi "Robust twin support vector machine for pattern classification". *Pattern Recognition* 46.1, pp.305-316, 2013.
15. Geng, Yishuang, et al. "Enlighten wearable physiological monitoring systems: On-body rf characteristics based human motion classification using a support vector machine". *IEEE transactions on mobile computing* 15.3, pp.656-671, 2016.
16. Tehrany, Mahyat Shafapour, et al. "Flood susceptibility assessment using GIS-based support vector machine model with different kernel types". *Catena* 125, pp.91-101, 2015.
17. Azar, Ahmad Taher, and Shereen M. El-Metwally. "Decision tree classifiers for automated medical diagnosis". *Neural Computing and Applications* 23.7-8, pp.2387-2403, 2013.
18. Lajnef, Tarek, et al. "Learning machines and sleeping brains: automatic sleep stage classification using decision-tree multi-class support vector machines". *Journal of neuroscience methods* 250, pp.94-105, 2015.
19. Wang, Ran, et al. "Segment based decision tree induction with continuous valued attributes". *IEEE transactions on cybernetics* 45.7, pp.1262-1275, 2015.
20. Oliver, Jonathan J., and David J. "Hand On pruning and averaging decision trees. Machine Learning": *Proceedings of the Twelfth International Conference, Morgan Kaufmann*, pp.430-437, 1995.
21. Parvin, Hamid, Miresmaeil MirnabiBaboli, and Hamid Alinejad-Rokny. "Proposing a classifier ensemble framework based on classifier selection and decision tree". *Engineering Applications of Artificial Intelligence* 37, pp.34-42, 2015.
22. Simidjievski, Nikola, Ljupco Todorovski, and Saso Dzeroski "Predicting long-term population dynamics with bagging and boosting of process-based models". *Expert Systems with Applications* 42.22, pp.8484-8496, 2015.

23. Wang, Guan-Wei, Chun-Xia Zhang, and Gao Guo "Investigating the Effect of Randomly Selected Feature Subsets on Bagging and Boosting". *Communications in Statistics-Simulation and Computation* 44.3, pp.636-646, 2015.
24. Abdollahi-Arpanahi, R., et al. "Assessment of bagging GBLUP for whole-genome prediction of broiler chicken traits." *Journal of Animal Breeding and Genetics* 132.3, pp.218-228, 2015.
25. Hegde, Chiranth, Scott Wallace, and Ken Gray "Using Trees, Bagging, and Random Forests to Predict Rate of Penetration During Drilling". *SPE Middle East Intelligent Oil and Gas Conference and Exhibition. Society of Petroleum Engineers*, doi:10.2118/176792-MS, 2015.
26. Korytkowski, Marcin, Leszek Rutkowski, and Rafal Scherer "Fast image classification by boosting fuzzy classifiers". *Information Sciences* 327, pp.175—182, 2016.
27. Appel, Ron, Thomas J. Fuchs, Piotr Dollar, and Pietro Perona "Quickly Boosting Decision Trees-Pruning Underachieving Features Early". *In ICML* (3), pp.594-602, 2013.
28. Kim, Tae-Kyun, and Roberto Cipolla "Multiple classifier boosting and tree-structured classifiers". *Machine Learning for Computer Vision. Springer Berlin Heidelberg*, pp.163-196, 2013.
29. Ye, Jerry, Jyh-Herng Chow, Jiang Chen, and Zhaohui Zheng "Stochastic gradient boosted distributed decision trees". *In Proceedings of the 18th ACM conference on Information and knowledge management, ACM*, pp. 2061-2064, 2009.
30. Nowak, Bartosz A., et al. "Multi-class nearest neighbour classifier for incomplete data handling". *International Conference on Artificial Intelligence and Soft Computing. Springer International Publishing*, 2015.
31. Osth, John, William AV Clark, and Bo Malmberg "Measuring the Scale of Segregation Using k-Nearest Neighbor Aggregates". *Geographical Analysis* 47.1, pp.34-49, 2015.
32. Chavez, Edgar, et al. "Near neighbor searching with K nearest references". *Information Systems* 51, pp.43-61, 2015.
33. Bhulai, Sandjai "Nearest neighbour algorithms for forecasting call arrivals in call centers". *Intelligent Decision Technologies. Springer International Publishing*, pp. 77-87, 2015.
34. Blaszczyn'ski, Jerzy, and Jerzy Stefanowski "Neighbourhood sampling in bagging for imbalanced data". *Neurocomputing*, 150, pp.529-542, 2015.
35. Kamley S, Jaloree S, Thakur RS. "Performance Forecasting of Share Market using Machine Learning Techniques: A Review". *International Journal of Electrical and Computer Engineering*. 6(6):3196, 2016.
36. Vidyullatha P, Rao DR. Machine Learning Techniques on Multidimensional Curve Fitting Data Based on R-Square and Chi-Square Methods. *International Journal of Electrical and Computer Engineering*. 1;6(3):974, 2016.