ORIGINAL PAPER

ENSEMBLE LEARNING ALGORITHMS

SELIN CEREN TURAN[1], MEHMET ALI CENGIZ[1]

Manuscript received: 16.02.2022; Accepted paper: 06.05.2022;
Published online: 30.06.2022.

Abstract. *Artificial intelligence is a method that is increasingly becoming widespread in all areas of life and enables machines to imitate human behavior. Machine learning is a subset of artificial intelligence techniques that use statistical methods to enable machines to evolve with experience. As a result of the advancement of technology and developments in the world of science, the interest and need for machine learning is increasing day by day. Human beings use machine learning techniques in their daily life without realizing it. In this study, ensemble learning algorithms, one of the machine learning techniques, are mentioned. The methods used in this study are Bagging and Adaboost algorithms which are from Ensemble Learning Algorithms. The main purpose of this study is to find the best performing classifier with the Classification and Regression Trees (CART) basic classifier on three different data sets taken from the UCI machine learning database and then to obtain the ensemble learning algorithms that can make this performance better and more determined using two different ensemble learning algorithms. For this purpose, the performance measures of the single basic classifier and the ensemble learning algorithms were compared*

Keywords: *Adaboost; Bagging; classification; ensemble learning algorithms.*

1. INTRODUCTION

Ensemble learning algorithms are a classification method. Classification is to group units according to their affinity or similarity in terms of characteristics. In classification practices, it is tried to predict which of the different classes or groups the existing units will belong to base on a few input variables.

Thanks to machine learning methods, models are established by classifying and grouping data or by creating relationships, correlations, and statistical results between data. The created model allows a prediction about the new data in case new data comes in that is not in the data set it was created. The accuracy level of the predictions reveals the performance of the created model on the data. Therefore, which algorithm produces better results in machine learning applications is important for the performance of the application.

Classification algorithms can be used in operations such as classical classifications, numerical estimates, credit approval, medical diagnoses, fraud detection, web page categorization, etc. In classification, firstly, a model is created by using training data. At this stage, the data of the input variables and the class values representing the target variable are given as input to the classification algorithm. The classification algorithm creates a model using known input and output values. In the next step, the model developed using training data is tested on a new data set. However, at this stage, class values of the target variable are not given as input to the model. Instead, the model is expected to accurately predict which classes the observations in the test data will belong to. An interpretation can be made on the

[1] Ondokuz Mayıs University, Faculty of Arts and Sciences, Department of Statistics, 55139 Samsun, Turkey.
E-mail: scturan1@gmail.com, macengiz@omu.edu.tr.

https://doi.org/10.46939/J.Sci.Arts-22.2-a18

performance of the model by looking at the results produced by the model tested on the test data [1]. As can be seen from these explanations, the classification consists of two stages: 1) Training phase: the system builds and learns a model for itself (the system learns the problem / learning the path to the solution). 2) Test phase: using the model obtained in the training phase for prediction. At this stage, how successful the prediction is important (using the learned path in new problems).

Classification algorithms are part of the learning process. That is, the world to be classified is learned first. Later, when unknown data comes into the system, a prediction can be made about the new incoming data. So, what has been learned needs to be tested. The mentioned learning process is also divided into two as supervised learning and unsupervised learning. Supervised learning consists of training data with inputs and outputs. That is, it is the case that an expert is constantly providing information to the system. Inputs and outputs are taught to the system such as what classes are available in the system, how many classes there are, etc. Supervised learning is a machine learning technique that produces functions based on these training data. With this function, it is tried to predict new data coming to the system. In other words, a function that matches between inputs and desired outputs is generated. The function can be determined by classification or regression algorithms. There is no training data in unsupervised learning. This method is a machine learning technique that uses a function to predict an unknown structure on unmarked data. It is essential for the system to learn everything on its own. Here, it is unclear to which class the input data belongs or not. That is, the system itself is asked to determine which class people belong to or how many there are. Then, when a data of an unknown class is called to the system, it assigns it to one of the appropriate classes it has determined and waits for a result.

Classification methods are used in many areas such as machine learning, shape recognition, and data mining. There are many basic classification methods put forward in the literature. Artificial neural networks, naive bayes, decision trees, logistic regression, k-nearest neighbor the commonly used methods. These methods are called base classifiers in ensemble learning algorithms and are used in the classification stage by training with a specific data set. Ensemble learning algorithms allow a combination of commonly used powerful classifiers. Running more than one classification algorithm at the same time provides better model performance. In recent years, it can be seen that ensemble learning algorithms are used effectively and successfully in various applications [2-8]. This method has been successfully applied to solve a wide variety of machine learning problems, including feature selection, confidence estimation, various classifications, and prediction problems.

In this study, it is aimed to analyze the problems in classification and prediction processes and an application on ensemble learning algorithms, which are very useful in preventing performance losses. For this purpose, the classification success of three different data sets taken from the UCI machine learning database with ensemble learning algorithms was examined

2. MATERIALS AND METHODS

2.1. SUPPORT VECTOR MACHINES

The support vector machine (SVM) is a method that performs classification by finding the most appropriate linear splitter hyperplane located farthest from any input point in a multidimensional space [9]. SVM developed by Cortes and Vapnik (1995) [10] performs the learning process independently of the distribution without the need for any joint distribution function information regarding the data [11]. It can be used for both linear and nonlinear problems. Kernel functions are used to convert the feature space to another feature space in nonlinear problems. Then, the hyperplane separating the classes from each other is found in the new feature space obtained by transformation. SVMs have low generalization error and computational cost and are easy to interpret. Besides, they need proper parameter and kernel selection. In addition, they only perform binary classification specifically [12]. The method tries to find a plane separating the points belonging to two classes.

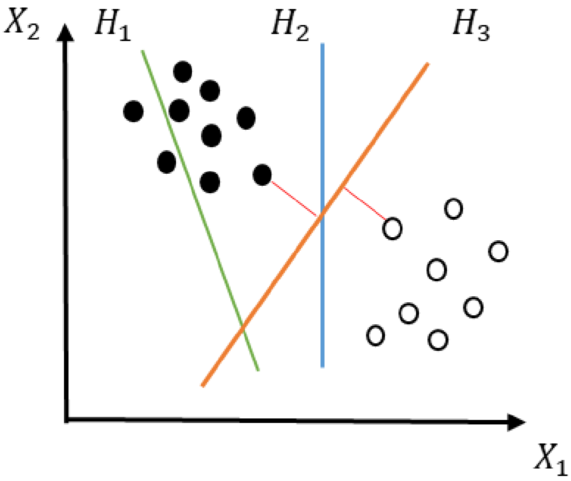

Figure 1. Support Vector Machines.

2.2. *K- NEAREST NEIGHBORHOOD*

K-nearest neighbor method (KNN) is one of the supervised learning algorithms used in classification problems and calculates the proximity of the test sample to the samples in the training set according to a predetermined distance criterion. After this process, the closest k samples are determined, and the test samples belong to the most class of these samples are included in that class [13]. The closest class member is determined using the Euclidean method to define the data that is unknown to which class it belongs. In order to realize this, all data are placed in an n-dimensional space and the data, which is not known to which class it belongs, is determined as the member of the class closest to it [14].

2.3. DECISIONS TREES

Decision trees (DT) are a basic method for classifying data. It has been used in many different data sets because it is easy to create and interpret the results. It consists of four basic steps. Firstly, with the creation of the training data set, the set of rules is created. Root node, inner node and leaves are determined with the selected features. Information gain is calculated by using the probability of occurrence of unexpected situations and uncertainties for each selected feature. The highest rate of knowledge gain is determined as root [15]. The most important problem in decision trees is the difficulty in determining the root, node, inner node and leaves of the decision tree to be formed.

2.4. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) is a successful and powerful classification method for learning nonlinear relationships between data when data have a complex structure and contain noise [16]. The basis of ANN is to imitate the superior characteristics of the human brain and to improve the performance of the software recognition-learning process.

2.5. NAIVE BAYES

Naive Bayes (NB) is a classification method based on probability and Bayesian decision theory. This method can be summarized as selecting the decision with the highest probability. The method tries to find the class to which the relevant input belongs, using probability and conditional probability. Therefore, it makes use of the following Bayesian rule [12],

$$p(c_i|x,y) = \frac{p(x,y|c_i)\, p(c_i)}{p(x,y)} \tag{1}$$

where i is the number of classes. Considering the example with two classes, the following rule is used to decide on the class [17]:

$$C(x) = \begin{cases} c_1, & p(c_1|x,y) \geq p(c_2|x,y) \\ c_2, & p(c_1|x,y) < p(c_2|x,y) \end{cases} \tag{2}$$

This rule can be easily extended for problems involving more than two classes. NB works even if a small amount of data is available and can easily solve problems with too many classes.

2.6. MODEL PERFORMANCE EVALUATION METHODS

Appropriate models can be produced from large data sets using machine learning methods. It is necessary to evaluate the success of the model to find out which model is better and to understand how well the learning model will work in the future. When only training data is used, an overfitting problem may be encountered. There are two commonly used methods to get rid of this situation. These are hold-out and cross validation methods. To avoid the problem of overfitting and to measure model performance, both approaches use test data from data that has not been used in the training data.

If there is a large data set, three different data sets as training, verification and testing can be obtained with the hold-out method. Training set includes input properties and output / outputs. Validation dataset is a sub-dataset used to evaluate the performance of the model obtained during the training phase. In addition, this dataset provides a test platform to determine which model is better and to determine the most suitable parameters for the models. Not all models need a validation data set. Testing dataset is used to evaluate the future performance of the model. If the results in the test data are worse than the training stage, an overfitting problem occurs.

In the cross-validation method, training data is divided into subsets. A single subset is used for training, while the remaining sets are used for validation. This process is repeated for all subsets in a cross-over. This process is done a predetermined number of k times. The data is divided into k pieces of equal size and evaluated k times.

2.7. MODEL PERFORMANCE EVALUATION MEASURES

Various methods have been developed to determine the best among classifiers. The most used of these methods is the contingency table. The contingency table shows the status of actual output values and values estimated by the model. This table is given in Table 1.

Table 1. Contingency table.

		True Condition		
		Positive	**Negative**	**Total**
Predicted	**Positive**	t_p	f_p	$f_p + t_p$
	Negative	f_n	t_n	$f_p + t_n$
	Total	$t_n + f_n$	$t_n + f_p$	

According to Table 1, some performance evaluation criteria calculated to evaluate the binary classification results, and the calculation formulas for these criteria are listed below. Considering the Table 1, the classification accuracy, error rate, sensitivity and specificity are calculated as follows, respectively [18].

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{3}$$

$$E\text{rror rate} = 1 - \text{Accuracy} \tag{4}$$

$$\text{Sensitivity} = \frac{t_p}{t_n + f_p} \tag{5}$$

$$\text{Specificity} = \frac{t_n}{t_n + f_p} \tag{6}$$

2.8. ENSEMBLE LEARNING ALGORITHMS

Ensemble learning algorithms is a machine learning technique and one of the most powerful modeling methods. It allows the combination of powerful classifiers that are widely used in different applications. The method provides better model performance by running multiple classification algorithms simultaneously. In addition, it produces more accurate solutions than the model obtained if a single classifier is used.

The basic principle behind the ensemble model is to come to a group of weak learners together to form a strong learner, thereby increasing the model accuracy. The main reason for errors encountered in learning arises from noise, bias, and variance. Combining multiple classifiers helps reduce these factors and can produce a more reliable classification than a single classifier. Combining multiple classifiers helps reduce these factors and provides a more reliable classification than a single classifier. The functional logic of ensemble learning algorithms is like the daily life of human beings. As human beings, we frequently use this technique in our daily life without realizing it [19]. Examples are consulting more than one doctor, seeking advice from different experts before receiving a major medical treatment, interviewing a candidate before hiring a candidate, evaluating, and making decisions according to references before buying a book, considering user reviews before purchasing a product. Apart from these, ensemble learning algorithms are used to develop the confidence of a classification model by weighting various individual classifiers and combine them to come to a final decision.

Each model is different from the other in the ensemble learning algorithms system. The technique, hypothesis, and population used for modeling differ from each other. The system of ensemble learning algorithms consists of three basic stages as diversity, training ensemble members and combing ensemble members. Diversity refers to the diversity in the results (or errors) of ensemble members, which is important in the ensemble learning algorithm system, and can be summarized as the creation of the training dataset for each classifier. Less variety leads to poor performance. Achieving diversity in ensemble learning algorithm systems can be achieved in several ways. For this purpose, it is the most common approach to create different subsets from a training data set. Using different data sampling and selection methods allows different algorithms to be formed. Diversity in ensemble learning algorithms is mostly achieved by classical random sampling and bootstrap method [19]. The bootstrap method involves a random sampling of data sets smaller than the data set. The selection of all samples in the data set is equally likely. It enables the model or algorithm to better understand the various biases, variances, and features that exist in the resampling. Taking a sample of the data allows the resampling to contain different features, and then it can be included as a whole. Each sample population is different. This property then affects the overall mean, standard deviation, and other descriptive metrics of a data set. In turn, it can develop more robust models. The method is also great for small-sized datasets that can tend to overfit. Algorithms using bootstrap enable testing of multiple models using multiple sample datasets, thus increasing robustness. The training ensemble members step is the core of any ensemble learning system. Various methods have been proposed and applied for training ensemble members. Combing ensemble members is the third step of the ensemble learning algorithms system, where individual classifiers are combined to obtain results that enable the system to increase its performance. Combing process is used immediately after training. Majority voting technique can be applied to improve prediction accuracy and increase performance [19].

Bagging, Boosting, Random Forest, and Random Subspaces are ensemble learning algorithms used as building blocks for building more powerful prediction models.

2.8.1. Bagging

Bagging (Bootstrap Aggregators) algorithm proposed by Breiman (1996) [20], also known as replacement learning, is a method that aims to retrain the basic learner by deriving new training sets from an existing training set. It is calculated by averaging the educated principal learner's decisions on a particular test sample for each data set. The purpose of the bagging method is to create differences by generating new data sets and thus increase the total classification success [21]. It is generated by random selection from training data containing K samples, replacing training data containing k samples (k<K). In this case, some training examples are not included in the new training set (approximately 33%), while some are included more than once. Each basic learner in the ensemble learning algorithms is trained with training sets containing different examples produced in this way and independent models are obtained. Then, according to the estimates obtained from the models, the results are combined with the majority vote [22].

With the bagging method, overfitting problems that may occur in trained data can be prevented. The process here is to help reduce the variance in models that may be very accurate. This is because the actual equation is too complex to account for every data point and outliers. Bagging eliminates this situation by creating its own variance among the data by sampling and modifying the data when testing multiple models. Thus, noise is reduced by using many samples consisting of data with various attributes (median, average). All models work at the same time and decide which hypothesis is the most correct by voting. Bagging algorithm is given in Table 2 [23].

Table 2. Bagging algorithm for classification

```
Input: Dataset $Z = \{z_1, z_2, \dots, z_N\}$ with $z_i = (x_i, y_i)$, where $x_i \in \mathfrak{X}$ and $y_i \in \{-1, +1\}$.
       $B$, number of bootstrap samples.
Output: $H: \mathfrak{X} \longrightarrow \{-1, +1\}$, the final classifier.

for $b$=1 to $B$ do
    Draw, with replacement, $N$ samples from $Z$, obtaining the $b$-th bootstrap sample $Z_b^*$.
    From each bootstrap sample $Z_b^*$, learn classifier $H_b$.
end for
Produce the final classifier by a majority vote of $H_1, \dots, H_B$, that is,
$H(x) = sign(\sum_{b=1}^{B} H_b(x))$.
```

2.8.2. Boosting

The Boosting algorithm is one of the widely used algorithms that has attracted the attention of many researchers working in the fields of data mining and machine learning. It was introduced by Schapire (1990) [24]to create a strong learner by bringing weak learners together and refers to a group of algorithms that use weighted averages. This algorithm combines the outputs without preference to any particular model, unlike Bagging algorithms in which each model works independently. Boosting is all about teamwork. Each working model determines which features of the next model to focus on. Observations are weighted to be repeated more frequently. The reason for this is to enable the algorithm to better learn from incorrectly predicted observations. Thus, new training data is created. All models run consecutively, and it is decided by weighted voting which hypothesis is the most correct. Boosting algorithm is given in Table 3 [23].

Table 3. Boosting algorithm for classification

```
Input: Dataset $Z = \{z_1, z_2, \dots, z_N\}$ with $z_i = (x_i, y_i)$, where $x_i \in \mathfrak{X}$ and $y_i \in \{-1, +1\}$.
Output: A classifier $H: \mathfrak{X} \longrightarrow \{-1, +1\}$.

Randomly select, without replacement, $L_1 < N$ samples from $Z$ to obtain $Z_1^*$.
Run the weak learner on $Z_1^*$, yielding classifier $H_1$.
Select $L_2 < N$ samples from $Z$, with half of the samples misclassified by $H_1$, to obtain $Z_2^*$.
Run the weak learner on $Z_2^*$, yielding classifier $H_2$.
Select all samples from $Z$ on which $H_1$and $H_2$disagree, producing $Z_3^*$.
Run the weak learner on $Z_3^*$, yielding classifier $H_3$.
Produce the final classifier as a majority vote: $H(x) = sign(\sum_{b=1}^{3} H_b(x))$
```

Boosting algorithm is a general approach that can be applied to many statistical learning methods for regression or classification such as Bagging. The main idea of many Boosting methods is to train estimators sequentially. The commonly used Boosting algorithm type is AdaBoost (Adaptive Boosting). It was first proposed by Freund and Schapire (1995) [25]. Compared to other Boosting algorithms, it is a method with high prediction speed, uses less memory and is easy to apply. Like bagging, the AdaBoost algorithm generates several classifiers and votes them. Beyond that, the two algorithms are drastically different. While AdaBoost algorithm generates classifiers sequentially, Bagging produces them in parallel. As the Boosting algorithm runs each model, it focuses on which data samples are the most successful and which are not. That is, to obtain an Adaboost classifier, the classifier is first trained on training data and predicted. Then, the "Relative Weight" of the misclassified training data is increased. The second classifier is trained with these increased weights and re-estimated. The weights are updated again and the system continues in this way.

In this method, while selecting a random sample for a classifier, priority is given to samples that previously made mistakes by the same classifier. In each of bagging iteration, the probability of selecting all samples to the training set is the same. However, the selection probabilities of the samples are updated in each cycle of Boosting. This ensures that the system focuses on the wrong decisions rather than the right decisions. The overall classification success is provided by updating the selection probabilities. AdaBoost algorithm is given in Table 4 [23].

Table 4. Adaboost algorithm for classification

```
Input: Dataset $Z = \{z_1, z_2, \dots, z_N\}$ with $z_i = (x_i, y_i)$, where $x_i \in \mathfrak{X}$ and $y_i \in \{-1, +1\}$.
       $M$, the maximum number of classifiers.
Output: A classifier $H: \mathfrak{X} \longrightarrow \{-1, +1\}$.

Initialize the weights $w_i^{(1)} = 1/N$, $i \epsilon \{1, \dots, N\}$, and set $m = 1$.
while $m \leq M$ do
    Run weak learner on $Z$, using weights $w_i^{(m)}$, yielding classifier $H_m: \mathfrak{X} \longrightarrow \{-1, +1\}$.
    Compute $err_m = \sum_{i=1}^{N} w_i^{(m)} \, h(-\, y_i H_m(x_i))$, the weighted error of $H_m$.
    Compute $\alpha_m = \frac{1}{2} \log \left(\frac{1 - err_m}{err_m}\right)$.
    $\{/*$ Weight of the weak learner$/*\}$
    For each sample $i = 1, \dots, N$, update the weight $v_i^{(m)} = w_i^{(m)} \exp\big(-\alpha_m y_i H_m(x_i)\big)$.
    Renormalize the weights: compute $S_m = \sum_{j=1}^{N} v_j$ and, for $i = 1, \dots, N$, $w_i^{(m+1)} = v_i^{(m)} / \, S_m$.
    Increment the iteration counter: $m \leftarrow m + 1$
end while
Final classifier: $H(x) = sign\big(\sum_{j=1}^{M} \alpha_j H_j(x)\big)$.
```

A high-performance classifier is obtained by combining the strongest weak classifiers in the Adaboost as in other ensemble learning algorithms.

3. RESULTS AND DISCUSSION

3.1. RESULTS

The aim of the study is to apply ensemble learning algorithms on different data sets taken from the UCI machine learning database, and to emphasize the importance of the method. For this purpose, three different data sets from the UCI machine learning database were used. The names of these data sets are Breast Cancer Wisconsin, Ionosphere and Pima Indians Diabetes. Information about the data sets used in the study is given in Table 5.

Table 5. Information about the data sets

Name of Data set	Variables	Explanation
Breast Cancer Wisconsin	X_1	Clump Thickness
	X_2	Uniformity of Cell Size
	X_3	Uniformity of Cell Shape
	X_4	Marginal Adhesion
	X_5	Single Epithelial Cell Size
	X_6	Bare Nuclei
	X_7	Bland Chromatin
	X_8	Normal Nucleoli
	X_9	Mitoses
	Class	Benign / Malignant
Ionosphere	X_1	V1
	X_2	V2
	.	.
	.	.
	.	.
	X_{34}	V34
	Class	Good / Bad
Pima Indians Diabetes	X_1	Pregnancies
	X_2	Glucose
	X_3	Blood Pressure
	X_4	Skin Thickness
	X_5	Insulin
	X_6	BMI
	X_7	Diabetes Pedigree Function
	X_8	Age
	Class	Diabetes / Not diabetes

The data set named Breast Cancer Wisconsin includes 9 quantitative independent variables and 1 qualitative dependent variable with 2 categories (444 benign, 239 malignant). It includes variables named clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses as independent variable. The qualitative variable shows whether the patients have breast cancer (Number of observations: 683). The data set named Ionosphere includes 34 quantitative independent variables and 1 qualitative dependent variable with 2 categories (126 good, 225 bad). It contains variables V1, V2, …, V34 as independent variables, and all variables are continuous. The qualitative variable shows whether the ionosphere structure is good or not (Number of observations: 351). The data set named Pima Indian Diabetes includes 8 quantitative independent variables and 1 qualitative dependent variable with 2 categories (500 patients, 268 not patients). It includes the variables of pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age as independent variables. The qualitative variable shows whether they have diabetes or not (Number of observations: 768).

CART was used as the basic classifier in the application phase of the study. The reason for this is that it is a method widely used in the literature. Also, Bagging and Boosting methods from ensemble learning algorithms have been applied. The Adaboost algorithm was used in the application of the Boosting ensemble learning algorithms. K-fold cross validation method was used to evaluate the performance of basic classifiers and ensemble learning algorithms in data sets. In addition, while applying bagging and Adaboost algorithms, separate analyzes were performed using 15, 50, 100 base classifiers. The accuracy metric was used to evaluate the classification performance of basic classifiers and ensemble learning algorithms. RStudio program was used for all analyzes in the study. The accuracy values obtained after after the classification methods applied for each data set used in the study are given in Table 6.

Table 6. Comparison of classification methods

Data set name	Classifiers	Number of classifiers	Accuracy
Breast Cancer Wisconsin	CART		0.652
	CART – Bagging	i=15	0.945
		i=50	0.980
		i=100	0.962
	CART – Adaboost	i=15	0.955
		i=50	0.979
		i=100	0.956
Ionosphere	CART		0.642
	CART – Bagging	i=15	0.919
		i=50	0.887
		i=100	0.892
	CART – Adaboost	i=15	0.864
		i=50	0.875
		i=100	0.914
Pima Indians Diabetes	CART		0.654
	CART – Bagging	i=15	0.783
		i=50	0.803
		i=100	0.736
	CART – Adaboost	i=15	0.783
		i=50	0.805
		i=100	0.794

As seen in Table 6, the changes in the performance of the models obtained by using CART alone on each data set and using Bagging and Adaboost algorithms were compared. When the classification performance of the CART base classifier applied for each data set is examined, it is seen that the accuracy values of this basic classifier are 0.652 for the Breast Cancer Wisconsin data set, 0.642 for the Ionosphere data set and 0.654 for the Pima Indians Diabetes data set. It is seen that the accuracy scores obtained after Bagging and Adabost methods for Breast Cancer Wisconsin data were significantly increased. Similarly, an effective increase in accuracy scores was observed after Bagging and Adabost methods applied on Ionosphere and Pima Indians Diabetes data sets. Therefore, if the ensemble learning algorithms are included in the system, we can say that the performance of the models increases, and better classifiers are obtained with more accurate results. According to these results, it can be said that ensemble learning methods show higher model performance despite using a single classifier.

4. CONCLUSIONS

The aim of the study is to find the best performing classifier with the CART basic classifier on various data sets, and after this process, to determine the ensemble algorithm that can make this performance better and more stable by using two different ensemble learning algorithms. For this purpose, three different data sets taken from the UCI machine learning database were used in the study. The Breast Cancer Wisconsin data set, the first of these data sets, consists of 9 quantitative independent and one 2-category qualitative dependent variable. The dependent variable, which is the class variable, consists of two categories (444 benign and 239 malignant) and indicates whether the patients have breast cancer. It consists of 683 observations in total. The results of the analysis are summarized in Table 6, and it has been revealed that Bagging and Adaboost algorithms have shown higher performance in detecting breast cancer disease. Similarly, it has been observed that the use of ensemble learning algorithms in the Ionosphere and Pima Indian Diabetes datasets increases model performance and becomes stable.

In summary, it is observed that using Bagging and Adaboost ensemble learning algorithm in three different data sets increases the CART basic classifier performance evaluation results. It is thought that the results of this study will shed light on other studies on the subject.

REFERENCES

[1] Seyrek, I.H., Ata, H.A., *Journal of BRSA Banking and Financial Markets*, **4**(2), 67, 2010.

[2] Sarmadi, H., Entezami, A., Saeedi Razavi, B., Yuen, K.V., *Structural Control and Health Monitoring*, **28**(2), e2663, 2021.

[3] Pinto, T., Praça, I., Vale, Z., Silva, J., *Neurocomputing*, **423**, 747, 2021.

[4] Guo, C., Liu, M., Lu, M., *Applied Soft Computing*, **103**, 107166, 2021.

[5] Sun, S., Jin, F., Li, H., Li, Y., *Applied Mathematical Modelling*, **97**, 182, 2021.

[6] Tama, B.A., Lim, S., *Computer Science Review*, **39**, 100357, 2021.

[7] Ghiasi, M.M., Zendehboudi, S., *Computers in Biology and Medicine*, **128**, 104089, 2021.

[8] Ullah, I., Liu, K., Yamamoto, T., Zahid, M., Jamal, A., *International Journal of Green Energy*, **18**(9), 896, 2021.

[9] Haykin, S., *Neural networks and learning machines*, 3rd Ed., Pearson Education, 2009.

[10] Cortes, C., Vapnik, V., *Machine Learning*, **20**(3), 1, 1995.

[11] Ayhan, S., Erdoğmuş, Ş., *Eskişehir Osmangazi Üniversitesi İktisati ve İdari Bilimler Fakültesi Dergisi*, **9**(1), 175, 2014.

[12] Harrington, P. *Machine learning in action*, Manning Publications, New York, 2012.

[13] Eroğlu, K., Palabaş, T., *Proceedings of Electrical, Electronics and Biomedical Engineering (ELECO)*, 512-516, 2016.

[14] Çalışkan, S.K., Soğukpınar, İ., K-means ve K en yakın komşu yöntemleri ile ağlarda nüfuz tespiti. *Ağ ve Bilgi Güvenliği Ulusal Sempozyumu*,16-18 Mayıs, Bildiri Özetleri Kitabı, 120-124, Girne, 2008.

[15] Cha, S.H., Tappert, C.C., *Journal of Pattern Recognition Research*, **4**(1), 1, 2009.

[16] Cortés, E., Martinez, M.G., Rubio, N.G., *International Advances in Economic Research*, **13**(3), 301, 2007.

[17] Kalaycı, T.E., *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, **24**(5), 870, 2018.

[18] Kartal, E., Programı, E., Balaban, M.E., *Sınıflandırmaya dayalı makine öğrenmesi teknikleri ve kardiyolojik risk değerlendirmesine ilişkin bir uygulama*. Doktora Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Enformatik Anabilim Dalı, 19-20, İstanbul, 2015.

[19] Raza, K., Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule. In *U-Healthcare Monitoring Systems*, pp. 179-196, 2019.

[20] Breiman, L., *Machine Learning*, **24**(2), 123, 1996.

[21] Kılınç, D., Borandağ, E., Yücalar, F., Özçift, A., Bozyiğit, F., Yazılım hata kestiriminde kolektif sınıflandırma modellerinin etkisi. *IX Ulusal Yazılım Mühendisliği Sempozyumu*, 9-11 Eylül, Bildiri Özetleri Kitabı, 113-121, Yaşar Üniversitesi, Bornova, 2015.

[22] Bulut, F., *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, **12**(3), 459, 2016.

[23] Ferreira, A.J., Figueiredo, M.A., Boosting Algorithms: A Review of Methods, Theory, and Applications. In Zhang, C., Ma, Y. (Eds), *Ensemble Machine Learning - Methods and Applications,* pp. 35-86, 2012.

[24] Schapire, R.E., *Machine Learning*, **5**(2), 197, 1990.

[25] Freund, Y., Schapire, R.E., *Journal of Computer and System Sciences*, **55**(1), 119, 1997.