Cumhuriyet Science Journal

e-ISSN: 2587-246X
ISSN: 2587-2680

Cumhuriyet Sci. J., 41(1) (2020) 93-105
http://dx.doi.org/10.17776/csj.544639

Classification of the placement success in the undergraduate placement examination according to decision trees with bagging and boosting methods

Tuğba TUĞ KAROĞLU [1,*] ***Hayrettin OKUT*** [2]

[1] *İpekyolu İMKB Science High School, The Ministry of Education, Van / TURKEY*

[2] *School of Medicine, Kansas University, Kansas / USA*

Abstract

The purpose of this study is to classify the data set which is created by taking students who placed to universities from 81 provinces, in accordance with Undergraduate Placement Examination between the years 2010-2013 in Turkey, with Bagging and Boosting methods which are Ensemble algorithms. The data set which is used in the study was taken from the archives of Turk-Stat. (Turkish Statistical Institute) and OSYM (Assessment, Selection and Placement Center) and MATLAB statistical software program was used. In order to evaluate Bagging and Boosting classification performances better, the success rates of the students were grouped into two groups. According to this, the provinces that were above the average were coded as 1, and the provinces below the average were coded as 0 and dependent variables were created. The Bagging and Boosting ensemble algorithms were run accordingly. In order to evaluate the prediction abilities of the Bagging and Boosting algorithms, the data set was divided into training and testing. For this purpose, while the data between 2010-2012 yearrs were used as training data, the data of the year 2013 were used as testing data. Accuracy, precision, recall and f-measure were used to demonstrate the performance of the methods in the study. As a result, the performance in consequence of "Bagging" and "Boosting" methods were compared. According to this; it was determined that in all performance measure marginally "Boosting" method produced better results than the "Bagging" method.

Article info

History:
Received:26.03.2019
Accepted:21.01.2020
Keywords:
AdaBoost, Bagging, Boosting, Ensemble Model.

1. Introduction

Data mining is a process used to make valid predictions by utilizing links and relationships within the data. The aim here is to reveal decision-making modelsto predict future behaviors based on past studies [1]. Data mining; includes a combination of techniques in different disciplines such as database technology, statistics, machine learning, pattern recognition, artificial neural networks, visualization of data and spatial data analysis [2].One of the techniques of data mining, Machine learning method consists of supervised and unsupervised learning methods.Supervised learning is mostly used in methods such as classification and regression. One of the main techniques used in classification and regression models is Decision Trees. Decision trees consist of a two-step process, learning and classification. A portion of the data is used during the learning phase. It's called training data. Training data generally consists of a large portion of the total data. The rest of the data is called test data. The main purpose of the classification of the ensemble is to produce a result by bringing together the values previously obtained by different classifiers. While this process is being performed, it is tried to make calculation by giving certain weight points to other classifiers. The main problem here is to combine different classification algorithms and decide which ratios will be used. The biggest advantage is that it can achieve better values because it uses the data of other methods together [3]. Ensemble classifiers have a significant usage area in recent years.In particular, features such as minimizing errors in the individual

*Corresponding author. *Email address:tugkaroglu@hotmail.com*

classifier structures and providing a faster classification algorithm have moved ensemble classifiers to this position [4], [5].

It is the Bagging method that was first implemented among the Ensemble models. Bagging, an abbreviation of Bootstrap and Aggregation, combines the classifiers, which have been re-sampled from the original data sets and trained by Bootstrap by different training datasets, brings together the most recent results and by using the optimization process, the most appropriate ensemble model is obtained which is acquired by simplifying the Bagging algorithm.With the optimization process it is emphasized how to select the most appropriate (optimum) classifiers according to the accuracy and diversity of the basic classifiers [6]. The Bagging algorithm, which is one of the most widely, used ensemble methods, creates ensemble classifiers by Bootstrap samples and improves classification by using different Bootstrap instances as a learning set.Recent studies have shown that the Bagging method reduces the effect of outlier values within the learning set [7]. Boosting is a method which obtains the best usage by improving the accuracy of a classifier.This classification method is used as a subprogram in order to construct the correct classifier in the training set.Boosting method applies the repetitive classification system in the training data,but in each step, learning attention concentrates on the different samples of this set by using adaptive weights. When the progress is completed, the obtained single classifiers are combined with the final, which is the highest accuracy classifier in the training set. As the various authors show both theoretically and empirically, the final classifier thus achieves the highest degree of accuracy generally in the test set [8], [9].Bagging and Boosting are two commonly used ensemble methods for classification.Their common goal is to improve the accuracy of a classifier that combines single classifiers that are slightly better than the random prediction. Among the family of Boosting algorithms, the AdaBoost algorithm is the best known, although it is used only for dividing into two [10]. Bagging and Boosting methods, which are the heuristic approaches used to develop classification models, produce various ensemble classifiers by influencing the train data given to the basic learning algorithms. These are very successful in improving some algorithms in artificial and real world data sets. The Bagging and Boosting methods have online versions that require only one pass through training data.Boosting brings together the communities of powerless classifiers to form a single powerful classifier. Successful models in the Boosting method give an extra weight to old estimates. In the Bagging method, however, each model is independently configured using the Bootstrap instance of the data set. Finally, the general estimate is made by the majority of votes [11].

The data set which is used in this study was taken from the archives of Turk-Stat. (Turkish Statistical Institute) and OSYM (Assessment, Selection and Placement Center). The variable values forming the data set were defined as “number of students”, “schooling ratio”, “number of illiterate”, “number of schools”, ”number of teachers”, “unemployment rate”, “employment rate”, “number of university graduates”, “number of students entering YGS (the transition to higher train examination)”, “number of students who received 180 points and more and placed”, “the number of students who received 180 points and more”,“ the ratio of students received 180 points and above ” from 81 provinces between the years 2010-2013.

In the evaluation, the researches were made according to the principles in this manual by examining the OSYM exam guide. According to thismentioned manual; candidates, who fail to score 140 points or higher at least one point type in YGS, do not have the right to choose a higher train program with their YGS (the transition to higher train examination) scores and to enter LYS (Undergraduate Placement Examination).Candidates who score between 140.00 and 179.99 in YGS can only choose associate's degree programs at vocational schools and open train programs. These candidates do not have the right to enter LYS. Candidates, who received 180.00 points or more, gain the right to enter LYS and can choose associate's degree programs at vocational schools, open train programs and as well as undergraduate degree programs which accept students with YGS points [12].

In the MATLAB statistical software program of the data set, the classification operations were performed by Bagging and Boosting methods.According to the data set, success rates were recalculated and the new calculated value is considered as dependent variable. Accordingly, dependent variable values are assigned so that the above average is equal to 1 and below average is equal to 0.

This study is an evaluation of the research classification and performance of the data set which is created by taking students who placed to universities from 81 provinces, in accordance with Undergraduate Placement Examination between the years 2010-2013 in Turkey, with Bagging and Boosting methods which are Ensemble algorithms.

2. Materials And Methods

2.1. Ensemble systems and their statistical features

The ensemble method, which is an important member of machine learning, has very effective learning algorithms that are both directing and non- directing in obtaining high accuracy results. Ensemble methods do not match the model using a single method. On the contrary, it uses the linear combination of many methods to match the model. In other words, ensemble methods make parameter estimation by creating multiple models and combining them, and improve the results. As a result, ensemble methods stand out as a very effective method to improve the estimation and predictive performance of statistical models. Each model, in which the Ensemble methods provide harmony, is called students or learners. In that case, the ensemble method brings together the information obtained by many students on the same problem and improves the prediction performance of the model. Each learner is called the basic learner. The knowledge of the basic learners is generally derived from the learning data set. This information is provided with the help of an algorithm and this algorithm is called the basic learning algorithm. This algorithm can be a decision tree, an artificial neural network, or another kind of algorithm [13].

Low defective classifiers show a high deviation tendency and vice versa are valid. On the other hand, there is also a deviation-reducing effect of average taking. Therefore, the goal of the ensemble methods is to establish a relatively defined or to create several classifiers with similar mistakes, to gather their data, to determine the average and to reduce deviation [14].

2.2. Bagging (bootstrap aggregating)

The Bagging algorithm, which is a method used to increase accuracy in Leo Breiman's classification and regression estimation, is an efficient and at the same time simple, ensemble-based algorithm [15]. Bagging refers to Bootstrap clustering and is a technique that uses Bootstrap to reduce variance and increases the accuracy of some predictors (can be used at classification and regression) [16].

2.3. Bootstrap

Bootstrap is a sample based statistical method.Bootstrap, which is referred to as the resampling method and used for prediction of accuracy, deals with small sample size [17].In this method, a lot of (non-segregated) training data is randomly subtracted from a single main data set. A Bootstrap training data set is created by randomly selecting the "N" sample by substituting it in an "N" sample data set.Each time a sample is selected and the selection of the selected sample is performed in an equally probabilistic manner. The sample taken is added to the training set again.Thus, in a training set, as there is the possibility of selecting the same sample more than once, it is also possible that the sample is not drawn at all. The possibility of not being selected is in this shape;

$$\left(1-\frac{1}{N}\right)^{N} \approx \exp(-1) \approx 0.368 \quad (1)$$

While 36.8% of the data sets constitute the test set, 63.2% of the date is obtained for the training set [17].

Selection of training and test sets is very important in order to create a safe model.Because if the test set represents the training set well, it is possible to obtain the correct estimate of the performance of the model. The random Bootstrap sample number "B" used to obtain the error predict, the sampling method can be repeated B times, and each of the Bootstrap samples is used to train the model.The models obtained to calculate the model's prediction error repeated B times by applying to the original data set or to the data which is not included in the sample and the Bootstrap error predict is obtained as the mean prediction error on the sample.

When the original sample is used, as the training set and the test set will be similar, the model will make relatively good predictions.Efron and Tibshirani's "0.632 predictor" is used to reduce this situation [18].The "0.632 Bootstrap error predictor" can be written as follows to eliminate overfitting in Bootstrap prediction [19].

$$Error\ /\ accuracy_{boot} = \frac{1}{B}\sum_{i=1}^{B}[(0.632 * test\ error_i) + (0.368 * total\ error)] \quad (2)$$

Here B states the number of random sample which is used to obtain error estimation; $\mathrm{testerror_i}$states the error which is obtained when the model that obtained as a result of i. Bootstrap sample, applied to the test set; and $total\ error_i$ states the error which is obtained when the model that obtained with i. Bootstrap, applied to the original data set.

Classically, Bootstrap is used to generate a limited number of large statistics about N number of samples $Z = \{Z_1, \ldots, Z_N\}$, regarding the population of P.

This idea is to ensure the equality$Z_b^* \subseteq Z$, b = 1,…….,B in B clusters by displacing each N random samples from Z which provides estimates of T (P), from B. These estimates are then converted to the

final estimate average, and thus it is possible to provide variance estimation and confidence intervals.

2.4. Out of bagging

If we take the traindata$Z = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, by creating a model for each $T_{B,K} = 1, \ldots, B$ Bootstrap sample drawn from these data, if it is wanted to find the estimation $(x, T_{B,K})$, the Bagging estimate is defined as;

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{Q}(x, T_{B,K}) \quad (3)$$

In general, each Bootstrap tree may differ from the original and may have a different number of end nodes.This is the average estimate which the x's created in the B tree, gives the prediction of Bagging. In a Bootstrap sample, 37% of the training data remains out of the sample.In an iteration, not drawn part is called "out of bag" data, the drawn part is called "in bag" data [20].

OOB (Out of bag) data is not used to prune or create a tree, however, it provides generalization of Bagging estimates and allows making better predictions on the node error. Comprising the part staying out of the training data,the rate approximately %37, actually is the test samples that are not used.Therefore, instead of using the response values of real train set in regression trees, using OOB estimation provides more accurate regression trees.

2.5.Boosting

Boosting is defined as a repetitive approach from a group of weak classifiers to create a strong classifier that result in clearly better results than random estimation, randomly gives less training error.Boosting differs from the Bagging approach in an important point by using SMV (Simple Majority Voting) on condition of combining the ensemble group of weak classifiers. The Bagging method is the iteration of the selected samples training data with Bootstrap to train the single classifiers.This means that each sample has an equal chance to be included in each training data set.In the Boosting method, however, the training data set for each final classifier focuses on samples that have been misclassified by previously produced classifiers.Therefore, while a larger weight value is assigned to the correct classification, the lower weight value is assigned to the one that has not been classified well [21]. Designed for binary class problems, the Boosting method creates three sets of weak classifiers at a time.The first classifier h_1 is trained on the random subset of the available training data, similar to the Bagging method.The second classifier h_2 is trained on a different subset of the original data set, half of which is misclassified and half correctly defined by h_1.Such a subset of train is called the "most informative", which gives the decision h_1. H_3 is trained with examples where the third classifier h_1 and h_2 are incompatible. These three classifiers are then combined through the "three-way majority vote". Provided that each classifier has a least expected ε <0.5 error rate from the classifier based on the binary classifier problem, Schapire has proved that these three ensemble classifiers limited the train error with $g(\varepsilon) < 3\varepsilon^2 - 2\varepsilon^3$(here, ε is the error of any of the three classifiers) [14].

2.6. Boosting process for classification

Input: For $x_i \in X$as $D_i = (x_i, y_i)$ and $y_i \in \{-1, +1\}$ the data set shall be $D = \{D_1, D_2, \ldots, D_N\}$.

Output: $H: X \to \{-1, +1\}$to be a classifier;

To get $D^*_1\, D_1^* - i$ from D, samples $L_1 < N$ are selected randomly and without displacement.

Providing H_1classifier, H_2 is concluded by running WL over $D^*_1, D_1^* - i$

In order to obtain D_i^*, the samples $L_2 < N$ are selected from Dwith half of the samples that are misclassified by H_1.

$H_2$is concluded by running WL over D_2^*.

All samples in Z are selected from H_1 and H_2 dispute and D_3^* is generated.

By running WL on D_3^* , H_3 classifier is obtained.

The final classifier is produced as a large majority vote.

$$H(x) = sign\left(\sum_{b=1}^{3} H_b(x)\right) \quad (4)$$

As it can be seen in the above algorithm; the training set is randomly divided into three parts so that there will be D_1^*, D_2^*, and D_3^* without replacing each other. For the given example, if the first two classifiers (H_1 and H_2) agree on the class label, this is the final decision for the example in question.It is expressed with D_3^* which is used to determine H_3, a set of examples they cannot compromise.Schapire has shown that this method of detection is strong. Furthermore, the error can be further reduced by repeated use of this approach. That is, each learner can be obtained spontaneously through the Boosting method.

2.7. AdaBoost

After their first individual study on the Boosting Algorithm, Schapire and Freund proposed the Adaptive Boosting (AdaBoost) algorithm [22]. Boosting algorithms vary according to the way they measure compliance deficiencies and how they select the observation weights in the next steps. Original Boosting algorithms such as AdaBoost have been used to develop binary classification problems [23].

2.8. AdaBoost algorithm used for binary classification

Input: For $x_i \in X$, $D_i = (x_i, y_i) y_i \in \{-1, +1\}$ as the maximum number of classifiers, the data set is $D = \{D_1, D_2, \ldots, D_N\}$

Output: $H: X \to \{-1, +1\}$to be a classifier;

The program is prepared for use. Weights are set according to$w_i^1 = \frac{1}{N}, i \in \{1, \ldots, N\}\ and\ m = 1$.

While $m \leq M$, it is run for poor students over Z by using w_i^1.

$H_m: X \to \{-1, +1\}$classifier is presented.

$err_m$is calculated.

Weighted error of H_m is found as$err_m = \sum_{i=1}^{N} w_i^{(m)} h\left(-y_i H_m(x_i)\right)$.

Calculated as $\alpha_m = \frac{1}{2} log\left(\frac{1-err_m}{err_m}\right)$.

For each examples it is *i=1,...,N;* the weight is updated as $v_i^{(m)} = w_i^{(m)} exp\left(-\alpha_m y_i H_m(x_i)\right)$.

The weights are returned to normal again. i=1,…, N and $w_i^{(m+1)} = v_i^{(m)} / S_m$ to be like this, iteration is applied so it will be $S_m = \sum_{j=1}^{N} V_j$, $m \leftarrow m + 1$

The process is finished.

Last classifier,

$$H(x) = sign\left(\sum_{j=1}^{M} \alpha_j H_j(x)\right) \quad (5)$$

The function $h: \mathbb{R} \to \{0,1\}$ used in the algorithm is the Heaviside function.This function is defined as $h(x) = 1$, if it is $x \geq 0$, if it is $x < 0$ it is defined as $h(x) = 0$. As a result both $y_i$and $H_m(x_i)$ receives the value{-1, +1}; if $y_i \neq H_m(x_i)$ is so, then it is $h\left(-y_i H_m(x_i)\right) = 1$, but if it is $y_i = H_m(x_i)$, it is $h\left(-y_i H_m(x_i)\right) = 0$. Err_m, m. is the weighted error rate of its classifier.

2.9. Overfitting problem in boosting method

Running the AdaBoost method with more than necessary to include iterations (stopping too late) can facilitate over-fitting.Because the complexity of the last section increases.On the other hand, stopping the algorithm prematurely not only results in a high error in the training data, but also results in a weaker estimate of the new data (underfitting). In the content of the AdaBoost method, although the algorithm may be over-adaptive, it is often seen to be resistant to over fitting [24], [25], [26].

3. Results And Dıscussıon

3.1. Classification results with bagging method

The success rates of the learners were grouped into two groups in order to better evaluate Bagging and Boosting classification performances. According to this, the provinces that are above the average are coded as 1, and the provinces below the average are coded as 0 and dependent variables are created. The ensemble methods, Bagging and Boosting algorithms, were run accordingly. In order to evaluate the prediction abilities of the Bagging and Boosting algorithms, the data set is divided into training and testing. For this purpose, two different approaches were conducted: 1) a ten-folded cross validation on the whole data and 2) assaying the data between years 2010-2012 as training data and the data of year 2013 as testing data.

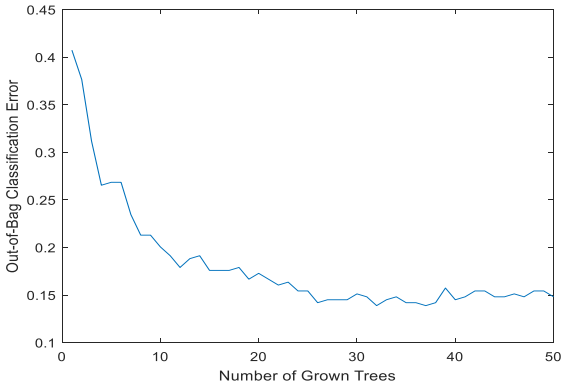

Figure 1. Out - of - Bag Classification Error

Small error values and error variance are considered as one of the important measure in evaluating performance of supervised machine learning methods such as Bagging and Boosting. As the number of trees is increased, it is evaluated by handling the downward trend of the error (Figure 1). As the number of trees increases, the classification error is expected to reduce as long as there is no over-fitting problem. In our

study, while the number of trees was 32, it was seen that the error was the lowest. However, in general terms, it is observed that there is no major change in the classification error after the 25th tree and it is turned into a partially stable state. In deciding the number of trees, the number of trees where the OOB error makes the most important drop is considered. In this study, the most important drop formation occurred in the 10th tree (Figure 1).

In practical applications, hundreds of trees can be enlarged with ensemble models. For example, as the optimum number of leaves is determined by using 50 trees for a better result, the number of properties can be estimated by creating a larger ensemble model with 100 trees.

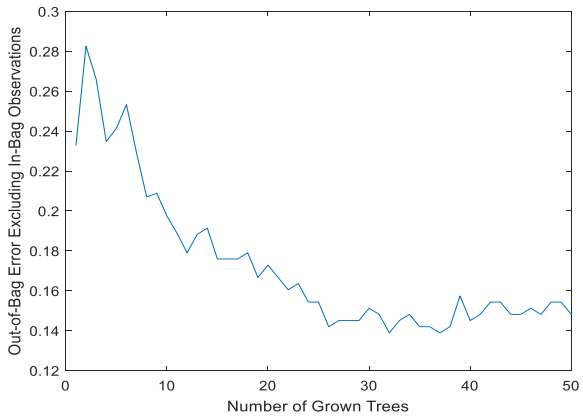

Figure 2. Classification Error Out of In-Bag Observation

In the Bagging method, the classification error results out of In-Bag observation are not significantly different from the OOB classification error results when looking at the out-of-bag error after repeated observations are removed. In particular, there were no major changes in the classification error after the 25th tree. The most important change was observed in the 10th tree (Figure 2).

The OOB observations show the TreeBagger property, describing the observations for which trees are out of the bag. Using this feature can control the function of observation in all training data which is "in bag" for all trees. The curl starts at about 2/3, which is the only fraction selected by Bootstrap iteration and reduces to 0 after about 10th tree. When the repeated observations in the Bagging method were examined, although the classification error started to reduce until the 25th tree, the important reduce in the error went on till 11th tree (Figure 3).

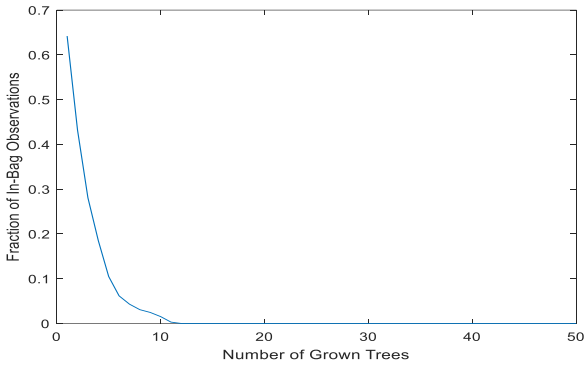

Figure 3. Error in Classification in In-Bag Observation

The prediction ability depends on important features rather than trivial features. When the data set is examined for each property, it shows which features are more effective in the classification. With the *OOBPermutedVarDeltaError* command in MATLAB, it stores an average increase in mean squared error mean on all trees, and divide this value by the standard deviation taken over the trees for each variable. Thus, it determines the contribution of variables to classification. This larger value means the more important variable. Then, an arbitrary cut-off point (such as 0.6) is determined as a threshold, and those above this threshold are determined as the most important variable. Using the most powerful features is an important strategy for increasing the predictive power of the Bagging algorithm. Here, the features coded from 1 to 11 are expressed as follows. 1st feature; number of schools, 2nd feature; number of teachers, 3rd feature; number of students, 4th feature; the number of students taking YGS, 5th feature; schooling rate, 6th feature; unemployment rate, 7th feature; employment rate, 8th feature; the number of illiterate, 9th feature; number of university graduates, 10th feature; the number of students placed by taking 180 score and above in YGS, and 11th feature; the number of students who scored 180 and above (Figure 4).

The number of students taking 180 points and above, which was described as the 11th feature in the classification, was more effective than the other features. After this feature, the 3rd feature "number of students" and the 2nd feature "number of teachers" are the features that are effective in the classification. The

least effective features in the classification are the "employment rate", which is expressed as the 7th feature, and after that, the variable of the number of schools, which is expressed as the 1st feature (Figure 4).

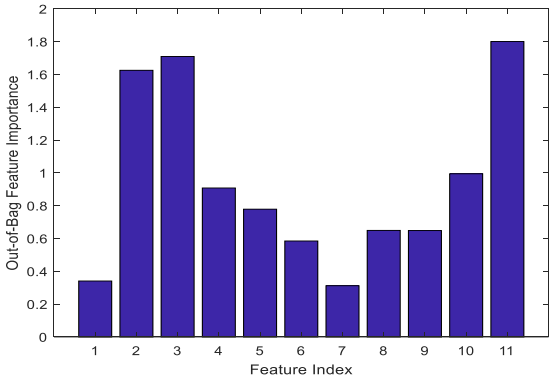

Figure 4. Effect of Properties in Classification

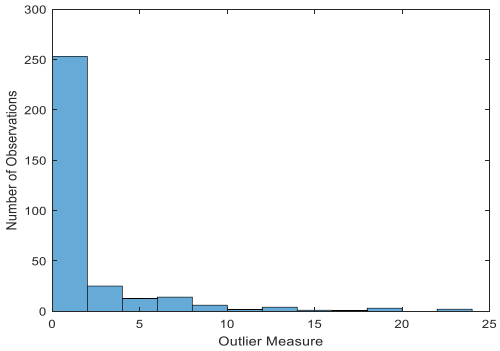

Figure 5. The Relationship between Numbers of Observations and Outliers

The Bagging algorithm is less sensitive than the Boosting algorithm against outliers. When the relationship between the outliers and the number of observations was examined, it was seen that that the majority of the values in the learning data set are defined as zero (0), i.e. outliers, there were only a few outliers, and these values did not affect the results in general (Figure 5).

In the scaled chart, again, outliers are attracting attention. When classifying; red color stamps are coded as 0, blue color stamps are coded as 1 and classification is done. When the graph is examined, it is seen that the classification is done correctly. In a classification where uncertainty reaches a trivial condition, red and blue stamps are expected to be clustered as two separate groups. Furthermore, a margin area between these two colors is expected to take place."0" has a high degree of accuracy. This is because in the coordinate axis there is accumulation and clustering in the region. Even "1" seems to be a bit disorganized; it is clustered in the region where it belongs to. A portion of the "0" is located on the graph in the "1" region. This indicates the existence of entropy partly. In the recall analysis, the classification performance of the Bagging algorithm was found to be 85.4%. The fact that "0" and "1" are nested in some places and out of its own set means that the classification is not 100% accuracy. This indicates the presence of outliers (Figure 6).

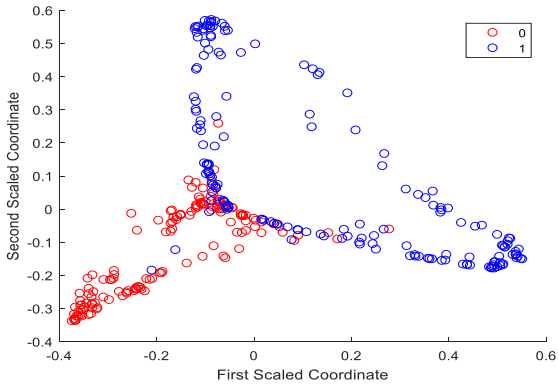

Figure 6. Scaled Results According to First and Second Coordinates.

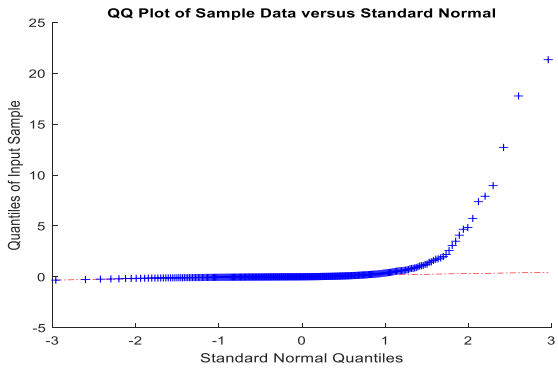

Figure 7. Graph of Errors Distribution

When the distribution graph of the errors is examined, it is expected that the values will change between -3 and +3 in the "0" axis in order to reflect the Gaussian distribution clearly. It is seen that the error values of the observed bias are partially skewed to the right (Figure 7). As the graph should continue in a linear manner, its follow-up a path between 0 and 2, opposite to what is expected is due to the presence of outliers. However, the existence of these outliers does

not prevent the error from being close to zero (Figure 7).

In addition, how MSE (Mean Squared Error) has changed according to the number of leaves was examined (Figure 8). In general, the number of leaves is selected as 5 for regression and one third of the input information is selected randomly. In the next step it is confirmed, by comparing the optimal number of leaves. The graph shown in red brings out the lowest values of the mean squared error average. The effect of the number of 5 to 100 leaves on the mean squared error mean was evaluated with different colors in the figure. When the graph is examined, it is seen that the average of mean squared error is obtained the lowest in 5 leaves. Although the number of leaves has increased to 5, 10 and 20, the ideal number of leaves should be between 5 and 20 leaves (Figure 8).

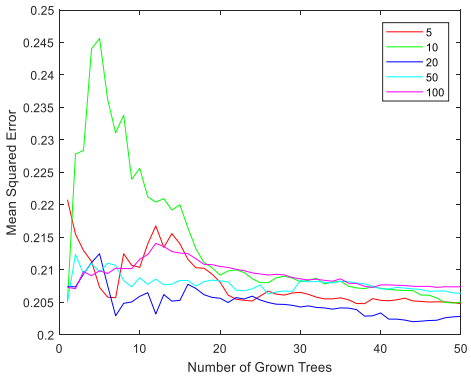

Figure 8. The relationship between the number of leaves and the Mean Squared Error

3.2. Classification by boosting method

While classification was made with Boosting method, three Boosting methods, AdaBoost, LP Boosting and Total Boosting were used, it was examined which one produce more ideal results. The graphs obtained as a result of classification made with Boosting methods are given (Figure 9, Figure 10, and Figure 11). Considering the number of ensembles commonly in the three graphs, there is a difference after 50. Cross-validation error in the test section, and change of the ensemble amount are seen (Figure 9). The algorithm with the smallest cross-validation error value is the best of the Boosting algorithms. Different loss functions are used to determine the cross validation error value and different results can be achieved in the same data depending on the loss function. The most commonly used loss function is the mean of J (.) Mean squared error and is calculated from the difference between the value and the estimated value. In the regression and classification based Boosting algorithms, cross validation errors in the data set used for testing is desired to be small.

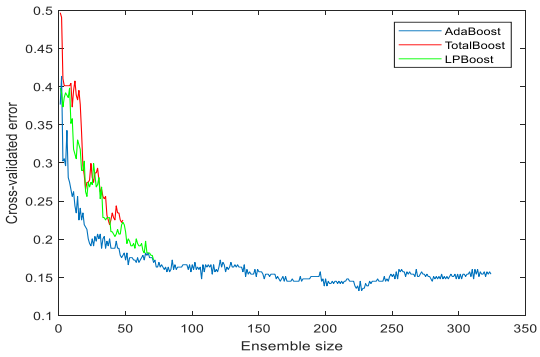

Figure 9. The relationship between the ensemble size and the amount of the cross validation error

When the errors between the train data and the ensemble size were analyzed by three different Boosting methods, it was observed that AdaBoost gave better results both in the test section data and in the training data. When the lost function values produced by AdaBoost algorithm are examined, it is seen that these values reduce below 0.15 now and then and change mostly between 0.2 and 0.15. In the AdaBoost training data set, it can sometimes be observed that the lost function is higher. This condition should meet the normal unless there is overfitting. Overfitting has started in the Total Boost and LP Boost methods, but danger of overfitting is not encountered due to the presence of the parameter controlling overfitting in the AdaBoost method (Figure 10).

When the distribution of the parameters according to the three different algorithms used in the Boosting method is examined, it is seen that the AdaBoost method is better than the other methods when the number of iterations is taken essentially (Figure 11).

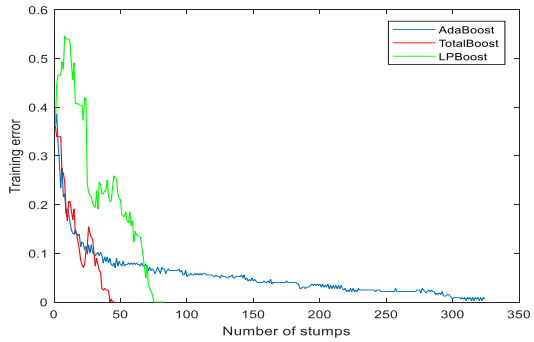

Figure 10. The relationship between stumps and training error

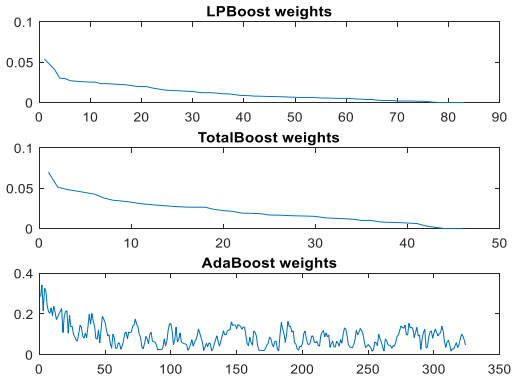

Figure 11. Distribution of parameters according to algorithms

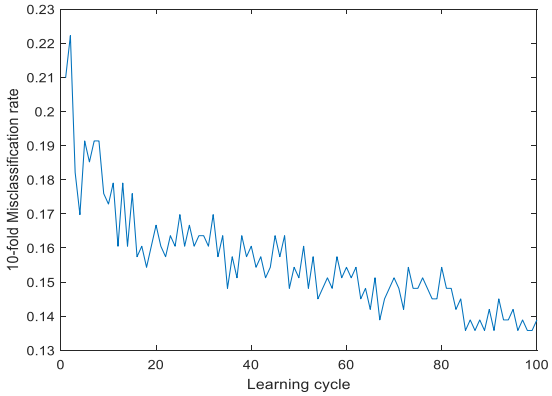

Figure 12. The learning cycle with 10-fold misclassification rate

When the misclassification rate of learning cycle consisting of 100 iterations is examined by dividing into 10 segments, one is the test and the remaining nine is training data, the data is divided into 10 segments, When calculating the cross validation in the classification, the data is divided into 10 segments. While nine of these segments are designated as training data, one is used as test data. Then another segment is used as test data and the remaining nine segments are calculated as training data. This process continues until all segments are calculated as test data. In this way, the results are obtained by calculating the average of ten cross errors. At the end of 100 iterations, the reduce in error is observed. In the 100th iteration, the process was ended here because the error converged zero (Figure 12).

3.3. Comparison of bagging and boosting results

Accuracy, precision, recall and f-measure were used to demonstrate the performance of the proposed methods in the study. These success measures are calculated as follows.

$$(Accuracy) = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$(Precision) = TP/(TP + FP) \quad (7)$$

$$(Recall) = TP/(TP + FN) \quad (8)$$

$$(F\text{-}Measure) = 2(Recall * Precision) / (Recall + Precision) \quad (9)$$

In these equations T, F, P and N; respectively expresses true, false, positive and negative. For example, TP shows the number of positive samples correctly classified; and FN shows the number of incorrectly classified negative samples.

Accuracy: It is the most popular and simple method used to determine success and this is defined as the ratio of the number of samples classified correctly (TP + TN) to the total number of samples (TP + TN + FP + FN).

Precision: It gives the degree of accuracy of the classifier result. It is the ratio of the number of positively affected samples (TP) to the total of positively classified samples (TP + FP).

Recall: It is the ratio of positively labeled samples (TP) to the total number of samples (TP + FN) that are really positive.

F-Measure: It is calculated by using precision and recall metrics. It is used to optimize the system towards precision or recall.

These results classification procedures were performed according to the 10-fold cross validation test. The data set is divided into 10 parts, classification process is performed by using respectively each of these 10 segments as test set, others as training set. At the end of the process, the results of 10 classification processes are taken as general success.

Table 1. The Confusion Matrix obtained from the Boosting AdaBoost algorithm.

		Below the average (0)	Above the average (1)	%
Output Class	Below the average (0)	122	27	% 81.88
	Above the average (1)	23	152	% 86.85
		% 84.13	% 84.91	% 84.56

Table 2. Confusion matrix obtained from the Bagging algorithm.

		Below the average (0)	Above the average (1)	%
Output Class	Below the average (0)	122	27	% 81.88
	Above the average (1)	26	149	% 85.14
		% 82.43	% 84.65	% 83.64

Table 3. The true positive (TP), false positive (FP), accuracy, precision, recall and f-measure obtained from Boosting AdaBoost algorithm.

Class	TP rate	FP rate	Precision	Recall	F-Measure
Below the average (0)	0.819	0.131	0.841	0.819	0.830
Above the average (1)	0.869	0.181	0.849	0.869	0.859
Average	0.846	0.158	0.846	0.846	0.846

Table 4. True positive (TP), false positive (FP), accuracy, precision, recall and f-measure obtained from the Bagging algorithm.

Class	TP rate	FP rate	Precision	Recall	F-Measure
Below the average (0)	0.819	0.149	0.824	0.819	0.822
Above the average (1)	0.851	0.181	0.847	0.851	0.849
Average	0.836	0.166	0.836	0.836	0.836

The above results have shown that the Boosting method has produced better results than the Bagging method in all performance measure marginally. For example, an average success rate of 83.6% was observed by the Bagging classification method.

When Bagging and Boosting comparisons were made, it was made by taking 1000 Bootstrap for Bagging. However, in such applications where the data set is not large, a more accurate comparison should be made by increasing the number of Bootstraps.

Moreover, we have conducted a ROC analysis to evaluate the prediction abilities of algorithms. The area under the ROC curve (AUC) calculated as 85.1% and 83.9% for Boosting and Bagging, respectively. In addition, it is observed that it supports the results of clustering analysis of the classification made with the Boosting method, when the two groups were formed according to the values below and below the average. For example, while the first observation value is included in the above-average group with a probability of 88%, the 11% represents the group that is below the mean. This situation indicates that the Boosting method verifies the clustering analysis.

4. Conclusion

In this study, the classification of the students who showed success at the Higher Train Transition Exam between the years 2010-2013 and received the score of 180 and above, placed to universities with Bagging and Boosting methods was handled. While creating the data set, 81 provinces were taken as basis and it was evaluated by clustering analyzesseparately for

four years.R computer program was used for clustering analysis and MATLAB computer program was used to classify with Bagging and Boosting methods.

When considering the results obtained, it was understood that the decision tree of AdaBoost method was a more successful method according to other ensemble methods. By selecting from AdaBoost method data set randomly,it determines with which faulty sample the classification procedures are performed instead of making new classifications. For this reason, it has shown higher success because it reduces the error in each iteration and is less affected by overfitting than other methods.Studies show that the Bagging method is clearly preferred according to the Boosting method when unbalanced data is noisy.This is because the Boosting method reduces the performance and focuses more on noise samples.In the case of over noise samples and unbalanced data, the Bagging Method exhibits a better performance than the Boosting method [27]. In the implementation phase of the Bagging algorithm,when sampling with nonparametric Bootstrap, it was confirmed by many studies that it is a common result [28].Bagging showed a lower performance than the Boosting method.The reason for this is that the Boosting method can re-train itselfover the samples that error made [29].In addition, Isikhan examined the performance of Regression Trees and he has seen that Bagging and Boosting methods play a better healing role on the test set performance of the regression tree [30].When the performance of the Bagging and Boosting algorithms are compared, it is determined the Boosting algorithm predicts better than the Bagging algorithm in terms of mean and standard error.

The increasing in the number of trees makes the algorithm more stable [31]. However, algorithm performance converges to a better result in accurate classification, as the number of trees increases,the algorithm will run more slowly and the process load will increase.In this study, however, the lowest value of the error is observed in 32 trees, due to the lack of a significant change in the classification errorby 25th tree,it will be sufficient to use 25 trees to alleviate the workload.In addition, the situation that can be clearly identified as the drop, in which the error is mostlyreduced, is in the10th tree.

The results obtained with the Bagging algorithm showed that in case of few outliers, these outliersin question are not very effective on the result. Similar results are drawing attentionwhen the data is scaled.Many studies reported similar results to these findings [32], [28], [33].Because the methods using Boosting algorithms show weak classification performance on adjusted variables according to variables that bears outliers and covariates.In contrast, the Bagging algorithm shows poor classification performance due to deviation from the sample [32].

When the results are examined, the average of mean squared errorreduces gradually as the number of trees and the degree of interaction increases.With or without excessive adaptation, reduce is seen in the average of mean squared error as the number of trees increases.

As a result, when our own study and other studies are examined, the experimental results in various data sets showed that Bagging algorithms are more effective against outliers in various base learners; Boosting algorithms are more effectiveagainst prediction bias.Despite these advantages, there are some weaknesses of the ensemble methods.For example, the interpretation of the results of Bagging and Boosting is still a problem standing in front of us.However, convergence problems are experienced in very large data sets. In addition, problem of overfitting is also in question with the Boosting algorithm in large data sets.

References

[1] Koyuncugil, A. S., Özgülbaş, N., İMKB'de İşlem Gören KOBİ'lerin güçlü ve zayıf Yönleri : Bir CHAID Karar Ağacı uygulaması. *Dokuz Eylül Üniversitesi İİBF Dergisi.* 23(1) (2008) 1-22.

[2] Hand, D.,Manilla, H., Smyth, P., Principles of Data Mining. MIT, USA, (2001) 546

[3] Augusty, S. M.,Izudheen, S., EnsembleClassifiers A Survey: Evaluation of Ensemble classifiers and data level methods to deal withim balanced data problem in protein- protein interactions. *Review of Bionformatics and Biometrics*, 2 (1) (2013) 1-9.

[4] Lee, S. L.A., Kouzani, A. Z., Hu, E. J., Random forest based lung nodule classification aided biclustering. *Computerized Medical Imaging and Graphics*,34 (2010) 535-542.

[5] Tartar, A., Kılıç, N., Akan, A., Bagging support vector machine approaches for pulmonary nodule detection. IEEE International Conference on Control, Decision and Information Technologies.Tunisia, (2013) 047-050.

[6] Zeng, X. D.,Chao, S., Wang, F., 2010.

Optimization of Bagging Classifiers Based on SBCB Algorithm. Proceedings of the ninth International Conference on Machine Learning and Cybernetics.11-14 July (2010) Qingdao. 262-267.

[7] Biggio, B.,Corona, I., Fumera, G., Giacinto, G., Roli, F., Bagging Classifiers for Fighting Poisoning Attacks in Adversarial Classification Tasks. Springer Verlag Berlin Heidelberg, (2011) 350-359.

[8] Breiman, L., Using iterated bagging to debias regressions. *Machine Learnings,* 45(3) (2001) 261-277.

[9] Banfield, R. E.,Hall, L. O., Bowyer, K. W., Kegelmeyer, W. P., Ensemble diversity measures and their application to thinning**.** *Information Fusion*, 6(1) (2005) 49–62.

[10] Alfaro, E.,Gamez, M., Garcia, N., Adabag: An R package for classification with Boosting and Bagging. *Journal of Statistical Software*, 54(2) (2013) 1-35.

[11] Kumari, G. T., A Study of Bagging and Boosting approaches to develop meta- classifier. Engineering Science and Technology: *An International Journal (ESTIJ)*, 2(5) (2012) 850-855.

[12] Anonim, Öğrenci Seçme ve Yerleştirme Sistemi Yükseköğretim Programları ve Kontenjanları Kılavuzu.http://www.osym.gov.tr. (2013)

[13] [Zhou, Z. H., Ensemble Methods: Foundations and Algorithms.Chapman & Hall/CRC Machine Learning &Pattern Recognition Series. Boca Raton, FL, United States of America. (2012) 236.

[14] Zhang, C.,Ma, Y., Ensemble Learning, Chap. 1. Ensemble Machine Learning(Editor: R. Polikar). (2012) 1-17.

[15] Coşgun, E.,Limdi, N.A., Duarte C.W., High dimensional pharma cogenetic prediction of a continuous trait using machine learning techniques with application to warfar indose prediction in African American. *Bioinformatics*, 27(10) (2011) 1384-1389.

[16] Breiman, L., Bagging predictors. *Machine Leraning*, 24 (2) (1996) 123-140.

[17] Efron, B.,Tibshirani, R., An Introduction to the Bootstrap.Chapman and Hall. London. (1993) 430.

[18] Grubinger, T.,Kobel, C., Pfeiffer, K.P., Regression tree construction by bootstrap: Model search for DRG-systems applied to Austrian health-data. *BMC Medical Informatics and Decision Making*, 10 (9) (2010) 1-11.

[19] Song, M.,Breneman, C.M., Bi, J., Sukumar, N., Bennett, K.P., Cramer, S.M., Prediction of protein retention times in anion exchange chromatograph ysystems using support vector regression. *Journal of Chemical Information and Computer Sciences*, 42(6) (2002) 1347-1357.

[20] Prasad, A.M., Iverson, L.R., Liaw, A., Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems, 9 (2006) 181–199.

[21] Schapire, R. E., The strength of weak learnability. *Machine Learning*, 5 (2) (1990) 197–227.

[22] Schapire, R. E.,Freund, Y., Boosting: Foundations and Algorithms. MIT Press, Cambridge, London, England. (2012) 528.

[23] Elith, J.,Leathwick, J.R, Hastie, T., A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4) (2008) 802-813.

[24] Grove, A.J.,Schuurmans, D., Boosting in the Limit: Maximizing the Margin of Learned Ensembles. In: Proceeding of the AAAI-98. John Wiley&Sons Ltd, (1998)692-699.

[25] Ratsch, G.,Onoda, T., Müller, K. R., Soft Margins for AdaBoost. *Machine Learning*, 42 (3) (2001) 287-320.

[26] Bühlmann, P.,Hothorn, T., Boosting algorithms: Regularization, prediction and model fitting (with Discussion). *Statistical Science*,22 (2007) 477-522.

[27] Khoshgftaar, T. M., Hulse, J. V., Napolitano, A., Comparing Boosting and Bagging Techniques with Noisy and Imbalanced Data. *IEEE Transactions on Systems Man and Cybernetics*, 41 (3) (2011) 552-568.

[28] Chen, Z., Lin, T., Chen, R., Xie Y., Xu, H., Creating diversity in ensembles using synthetic neighborhoods of training samples. *Journal Apllied Intelligence*, 47 (2) (2017) 570-583.

[29] Kotsiantis, S. B., Bagging and Boosting variants for handling classification problems: a survey. *Cambridge University Press.* 29 (1) (2014) 78-100.

[30] Işıkhan, S., Mikrodizilim Gen İfade Çalışmalarında Genelleştirme Yöntemlerinin Regresyon Modelleri Üzerine Etkisi , PhD Thesis,.

Hacettepe Üniversity, Ankara (2014)

[31] Dietterich, T., An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning,* 40(2) (2000) 139–157.

[32] Davidson, I., Fan, W., When Efficient Model Averaging Out- Performs Boosting and Bagging. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases.Berlin, Germany, (2006) 477-486.

[33] Arsov, N.,Pavlovski, M., Basnarkov, L., Kocarev, L., 2017. Generating highly accurate prediction hypotheses through collaboratative ensemble learning. Scientific Reports, 7(44649) (2017) 1-34.