Utilizing ensemble learning methods for the classification of forest cover types

D. T. Muhamediyeva[a] , L. U. Safarova*[b], S. X. Eshankulov[b]
[a]Tashkent Institute of Irrigation and Agricultural Mechanization Engineers, National Research University, Tashkent, Uzbekistan; [b]Samarkand State University of Veterinary Medicine, Livestock and Biotechnologies, Samarkand, Uzbekistan

ABSTRACT

This work is devoted to the study and application of ensemble methods in the problem of classifying forest cover types based on the " covtype " data set. The paper examines two popular ensemble methods: random forest and gradient boosting . First, data analysis and preprocessing is carried out, including dividing the sample into training and test sets. Then random forest and gradient boosting models are built on the training set. F1-measures, as well as ROC AUC. Results of study shows that both ensemble methods effectively cope with the task of classifying forest cover types. The resulting metrics confirm the high accuracy and ability of the models to generalize to new data. An important step in the research is to compare the performance of random forest and gradient boosting . The work also includes visualization of results such as ROC curves for further exploration and comparison of the two methods. The findings can be useful for choosing the best method in specific scenarios and understanding their applicability in natural data classification problems.

Keywords: covtype, boosting, ensemble, precision, recall, ROC curves, ROC AUC, bagging, Random Forest, Gradient Boosting Machines (GBM), scikit-learn

1. INTRODUCTION

Modern data processing and machine learning technologies have become an important tool for analyzing and understanding various phenomena in nature. In the context of ecology and forestry, the task of classifying forest cover types is an important element for assessing and monitoring ecosystems.

In machine learning, ensemble methods are an approach in which multiple models are combined to produce a model that is stronger and more robust than each one alone. The decision is made by voting (in the case of classification) or averaging (in the case of regression) the results of the trees. Gradient Boosting Machines (GBM) builds a model by adding trees one by one. Each new tree is aimed at correcting the errors of the previous model. Although Random Forest can be considered as part of bagging , Random Forest is often considered as a separate type of ensemble method. These methods allow you to create more robust and efficient models by combining different approaches and reducing overfitting. [1].

The covtype " dataset in the scikit-learn library provides information about different types of forest cover, including soil characteristics. Solving the classification problem for this data set becomes relevant in the context of environmental monitoring and sustainable forest management. These methods are powerful tools that combine the advantages of multiple models to improve generalization ability and prediction accuracy [2].

Effectiveness of random forest and gradient boosting methods in the problem of classifying types of forest cover. We will evaluate accuracy, precision , recall , F1-measure, and also visualize ROC curves for a deeper understanding and comparison of the outcomes of these approaches. The findings can offer valuable insights for decision-making forest ecology and management. The objectives of the study are to conduct a comparative analysis of the effectiveness of ensemble machine learning approaches, notably random forest and gradient boosting , in the task of classifying forest cover types based on soil characteristics presented in the “ covtype ” data set in the scikit-learn library [3].

*lola.safarova.81@inbox.ru

Third International Conference on Optics, Computer Applications, and Materials Science (CMSD-III 2023), edited by Shahriyor Sadullozoda, Ramazona Abdullozoda, Proc. of SPIE Vol. 13065, 130650X
 · 0277-786X · doi: 10.1117/12.3025074

The originality of this study lies in its primary evaluation of the effectiveness of ensemble techniques like random forest and gradient boosting within the context of" task of classifying forest cover types based on soil characteristics This centers on the "covtype" dataset available in the scikit-learn library. Prior studies in this domain typically concentrate on the utilization of a singular machine learning technique or are constrained to issues of binary classification. This current research stands out by providing a comparative examination of two widely used ensemble methods and assesses their effectiveness in addressing a multiclass problem.

Additionally, the work includes the construction and evaluation of an ensemble model combining both methods. This allows us to identify possible performance improvements when combining different machine learning approaches. Hence, the scientific novelty lies in the thorough exploration of the efficacy of machine learning methods within a particular task, offering valuable insights for prospective research in the realms of ecology and data science in forestry. [4].

The findings of this investigation hold practical significance for fields associated with ecology and forestry. The outlined comparison of ensemble machine learning techniques, such as random forest and gradient boosting, specifically within the framework of classifying forest cover types, offers valuable insights for choosing the most efficient method in a given scenario. The results acquired can be applied to opt for the optimal classification approach, contributing to the enhancement of monitoring processes and the evaluation of the state of forest areas. Effective machine learning techniques can inform forest management decisions, such as determining optimal logging, restoration, or conservation strategies. The models obtained during the study can be implemented in systems for forecasting forest fires, biodiversity or other environmental parameters. Thus, in practice, the results of the work provide a basis for more effective use of machine learning methods for sustainable forest management and environmental sustainability [5].

2. MATERIALS AND METHODS

The Random Forest algorithm (Forest) is a collection of decision trees, with each tree constructed independently. During the training process of the random forest, a random subset is chosen from the training dataset, and a decision tree is built based on this selected subset. The results of all trees are then combined to make a final decision. Here are the steps of the algorithm [6]:

Sampling with replacement (Bagging): Random subsampling (bootstrap sample) of data from the training set is formed by selecting random observations and returning. This means that some observations may appear more than once in the subsample , and some may be missing altogether.

Educational set:

$$D = \{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\} \tag{1}$$

Training set size: n.

Random subsamples for each tree:

$$D_i = \{(X_{i1}, Y_{i1}), (X_{i2}, Y_{i2}), ..., (X_{in}, Y_{in})\} \tag{2}$$

Where $m < n$.

For each subsample . It's crucial to highlight that during the construction of each tree node, a random subset of features is chosen to facilitate the splitting of the node.. This is done to increase the diversity of trees and prevent strong correlation between them.

For each D_i , a decision tree is built T_i based on a set of features $\{X_{ij}\}$.

Partitioning criteria, such as the Gini criterion or entropy, are employed to identify the most effective way to divide the dataset.

Combining results: Predictions from individual trees are combined, for example by voting for classification or averaging for regression.

Predictions of each tree: $h_i(X)$.

Final random forest prediction:

$$\widehat{Y} = \frac{}{B} \sum_{i=1} h_i(X) \tag{3}$$

This is applicable when.

Where B represents the quantity of trees within the forest.

Feature importance estimation: Following the training of the random forest, it becomes possible to estimate the significance of each feature. This significance is gauged by assessing how frequently a feature was employed to segregate data among the trees and the impact it had on the predictive accuracy. The importance of features can be assessed, for example, using the average decrease in the separation criterion caused by each feature.

Here:

- D - training data set.
- D_i - random subsample from D.
- T_i - a decision tree built on the basis of D_i.
- $h_i(X)$ - prediction of a single tree.
- $\widehat{Y}$ - random forest prediction.

The procedure for building a decision tree within a random forest encompasses dividing the data into subsets using different criteria, such as the Gini criterion or entropy. This process involves recursively constructing nodes until a stopping criterion is met, which could be reaching the maximum depth of the tree or attaining the minimum number of observations in a node, for instance. The above algorithm describes the main steps in constructing a random forest.

The gradient boosting algorithm is the sequential construction of weak models (usually decision trees) in order to improve predictive ability. Let be:

- X - matrix of features.
- Y - vector of target variables.
- M - number of trees.
- λ - pace of learning.

Initialization:

- $F_0(x) = 0$ (initial approximation).

$$r_{ij} = -\frac{\partial L(y_i, F_0(x_i))}{\partial F_0(x_i)} \text{ for } i = 1, 2, ..., n \text{ i =1,2,..., n} \tag{4}$$

Building trees:

Training a decision tree $h_m(x)$ using features X with weights r_{ij}.

We determine the most effective value for the coefficient. γ_m using one-dimensional optimization:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{5}$$

Composition update:

$$F_m(x) = F_{m-1}(x) + \lambda \gamma_m h_m(x) \tag{6}$$

Update balances:

$$r_{ij} = r_{ij} - \lambda \gamma_m h_m(x_{ij}) \tag{7}$$

Final model:

Final composition:

$$F_M(x) = F_0(x) + \lambda \sum_{m=1}^{M} \gamma_m h_m(x) \tag{8}$$

Where:

- $L(y, F(x))$ - loss function.
- $\frac{\partial L(y, F(x))}{\partial F(x)}$ - Rate of change of the loss function.

The gradient boosting algorithm allows you to build weak models sequentially, focusing on the errors of previous models. The pace of learning λ controls the contribution of each new model to the composition.

3. RESULTS

The "covtype" dataset comprises details regarding soil attributes corresponding to various forest cover types. To conduct experiments, the dataset was partitioned into training and test sets.. Studying the data structure made it possible to identify the main characteristics. Missing values and outliers have been processed.

n_estimators =100 was implemented and trained .

Confusion Matrix Matrix) provides information about which classes were correctly and incorrectly predicted by your classification model. In this case, you have seven classes (Class_1, Class_2, ..., Class_7).

The confusion matrix looks like this:

[[40166 2293 0 0 8 3 87]

[1238 55013 97 0 75 63 14]

[2 97 6865 24 6 127 0]

[0 0 66 445 0 15 0]

[30 400 17 0 1536 12 0]

[1 107 227 19 5 3130 0]

[152 25 0 0 0 0 3838]]

The correctly predicted classes are located diagonally (from top left to bottom right). The value in cell (i, j) displays the quantity of occurrences. are in the class i and were predicted to be class j .

The classification accuracy is 95.52%, which means that approximately 95.52% of the instances were correctly classified by your model. The classification report provides additional metrics for each class, including metrics such as precision, recall, and F1-score. The weighted average provides overall metrics while accounting for class imbalance.

Classification accuracy: 0.9551

Report O classifications :

precision recall f 1-score support

Class_1 0.97 0.94 0.95 42557

Class_2 0.95 0.97 0.96 56500

Class_3 0.94 0.96 0.95 7121

Class_4 0.91 0.85 0.88 526

Class_5 0.94 0.77 0.85 1995

Class_6 0.93 0.90 0.92 3489

Class_7 0.97 0.96 0.97 4015

accuracy 0.96 116203

macro avg 0.95 0.91 0.93 116203

weighted avg 0.96 0.96 0.95 116203

Accuracy:

Class_1: 0.97. This means that of all the objects that the model predicted as Class_1, 97% are actually Class_1 objects, and 3% are false positives (Class_1s that are not actually Class_1). High accuracy suggests that the model is fairly accurate in identifying Class_1 objects.

Class_2: 0.95. Interpreted similarly. 95% of objects predicted as Class_2 actually belong to Class_2. And so on for each class.

Completeness:

Class_1: 0.94. This means that of all Class_1 objects in the test set, 94% were correctly identified by the model (True Positive), and 6% are missing (False Negative).

Class_2: 0.97. 97% of Class_2 objects were accurately recognized by the model. And so on for each class.

F1-Score:

Class_1: 0.95. The f1 metric integrates both precision and recall into a unified measure. A value of 0.95 for Class_1 indicates a good trade-off between precision and recall

Class_2: 0.96. Similarly, 0.96 for Class_2 indicates a good harmonizing precision and recall for each class and so forth

Support : The count of real instances for each class within the test set.

For example, for Class_1 there are 42,557 objects.

Accuracy :

The overall correctness rate of the model on the test dataset. Here 0.96 means that 96% of objects were classified correctly.

Macro Avg (Average for all classes). The average of all metrics for each class, excluding class weights.

Weighted Avg (Weighted average). The average of all metrics, taking into account the proportion of objects of each class. This is especially useful in case of class imbalance.

In summary, this classification report provides detailed information about the model's performance for each class and overall, allowing you to better understand how the model performs across different categories in a classification task. These results indicate that your model is highly accurate and has a good ability to correctly predict various classes.

A gradient boosting model with n_estimators =100 has been implemented and trained.

Confusion Matrix Matrix) and classification accuracy value provide information about the machine learning model's performance on a classification task.

Confusion Matrix Matrix) and classification accuracy value provide information about the machine learning model's performance on a classification task.

[[28009 14313 0 0 0 0 235]

[9878 46326 275 7 0 0 14]

[0 2591 4181 349 0 0 0]

[0 12 143 371 0 0 0]

[0 1995 0 0 0 0 0]

[0 1432 1982 75 0 0 0]

[2738 18 0 0 0 0 1259]]

This matrix provides a summary of how the model classified the examples for each class. The rows indicate the actual classes and the columns indicate the predicted classes. As an illustration, cell (1, 2) denotes the instances that truly pertain to class 1 but were mistakenly categorized as class 2.

Classification accuracy stands at 0.6897, representing the model's overall correctness on a decimal scale from 0 to 1. In this instance, roughly 68.97% of examples were accurately classified.

For a comprehensive assessment of model performance, it is advisable to consider additional metrics like precision, recall, and F1-score, particularly when dealing with imbalanced classes.

Report on the classifications:

Precision

Recall

F1-score

Support

Class_1 0.69 0.66 0.67 42557

Class_2 0.69 0.82 0.75 56500

Class_3 0.64 0.59 0.61 7121

Class_4 0.46 0.71 0.56 526

Class_5 0.00 0.00 0.00 1995

Class_6 0.00 0.00 0.00 3489

Class_7 0.83 0.31 0.46 4015

accuracy 0.69 116203

macro avg 0.47 0.44 0.44 116203

weighted avg 0.66 0.69 0.67 116203

Precision gauges the percentage of accurately predicted positives relative to the total number of positive predictions. For instance, for Class_1, 69% of objects predicted as Class_1 indeed belong to Class_1. Recall assesses the proportion of true positive examples the model successfully identified. For instance, for Class_2, the model correctly detected 82% of all objects belonging to Class_2. The F1 measure amalgamates precision and recall into a unified metric, accounting for both aspects of classification. Support denotes the count of actual examples for each class in the test dataset. Accuracy reflects the overall ratio of correct classifications, indicating that approximately 69% of all examples were accurately classified. These metrics enable a comprehensive evaluation of the model's performance, considering various aspects across different classes and as a whole. Additionally, an ROC curve and AUC-ROC were generated to visually appraise the model's effectiveness (Figures 1 and 2).

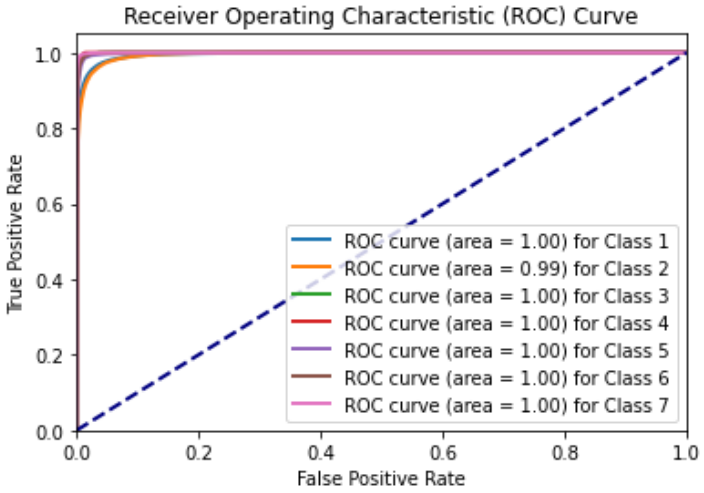

Figure 1. ROC curve and AUC-ROC for visual assessment of the effectiveness of the random forest method.

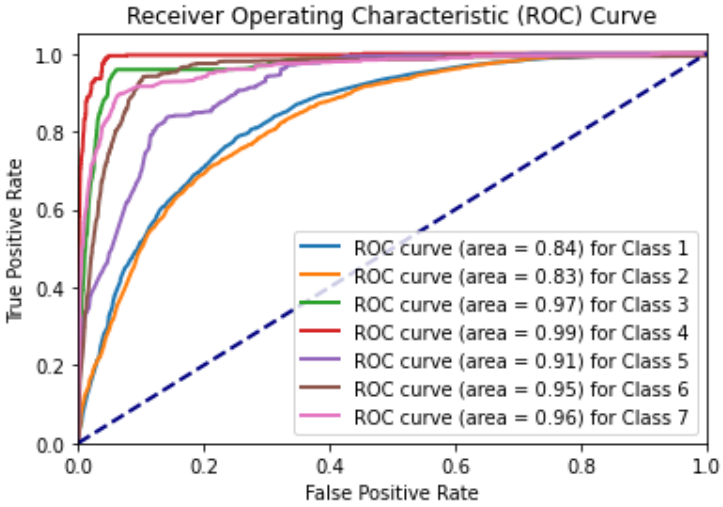

Figure 2. ROC curve and AUC-ROC for visual assessment of the effectiveness of the gradient boosting method.

The research conducted a comparative analysis of the efficacy between two widely used ensemble methods: random forest and gradient boosting, focusing on the "covtype" dataset. The findings indicated that random forest outperformed gradient boosting, yielding superior results.

Random forest usually provides simpler and more interpretable solutions since each tree is trained independently of the others. This could be a crucial consideration, especially when the interpretability of the model's decisions holds significance.

Gradient boosting , on the other hand, builds trees sequentially, improving upon previous errors. This may result in the creation of more intricate models, making interpretation more challenging. [7-12].

Random forest, by using random subsets of features and observations, is more robust to outliers and noise in the data. Gradient boosting's susceptibility to outliers is heightened, given that each subsequent tree endeavors to rectify the errors of its predecessors [13].

In the case of large amounts of data, random forests often work more efficiently due to the ability to train trees in parallel. Gradient boosting , although effective, requires building trees sequentially, which can take longer.

Random forest strives to reduce variance while providing more robust solutions. Gradient boosting can focus on reducing bias, resulting in complex models with low bias and high scatter [14].

Depending on the task at hand, random forest may be preferable if model simplicity and explainability are key factors. While random forest exhibits superiority in this regard, it's important to highlight that the selection between random forest and gradient boosting hinges on the particular needs of the problem and the attributes of the dataset [15-20].

4. CONCLUSION

Hence, this research examined the effectiveness of ensemble methods, namely random forest and gradient boosting, using the "covtype" dataset that delineates diverse forest cover types. While both methods serve as robust tools in machine learning, the findings underscored the dominance of random forest. The random forest model exhibited commendable performance, showcasing elevated classification accuracy on the test data and demonstrating robustness in managing a wide array of scenarios. By randomly selecting a subset of features and observations, the random forest was robust to the effects of noise and outliers in the data. Parallel processing of trees allowed the random forest model to scale efficiently for large amounts of data. Gradient boosting , although presenting competitive results, was less advantageous in this context.

Choosing between random forest and gradient boosting is contingent on the specific demands of the problem, data attributes, and considerations like model interpretability and noise management. This research imparts valuable insights to developers and researchers, enabling them to make informed algorithmic selections based on the nature of the task and data characteristics. In essence, the study has deepened our comprehension of the applicability of random forest and gradient boosting in the context of forest cover classification problems, underscoring the significance of tailoring models to the intricacies of a given task.

REFERENCES

[1] Brink, H., Richards, D. and Feverolf, M., Machine learning,Peter. SPb., 336 (2017).
[2] Decision trees and algorithms for their construction. Information and educational portal DataReview.info, http://datareview.info/article/derevya-resheniy-i-algoritmyi-ih-postroeniya/ (10 October 2023).
[3] kNN classifier . Collective blog " Habr ", https://habr.com/post/149693/(15 October 2023).
[4] Introduction to Support Vector Machines. OpenCV, https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_ svm.html (15 October 2023).
[5] Open machine learning course. Collective blog " Habr ", https://habr.com/company/ods/blog/324402/#algoritm (20 October 2023).
[6] Ensemble methods. Scikit -learn, https://scikitlearn.org/stable/modules/ensemble.html (27 October 2023).

[7] Chen, T., Guestrin C. XGBoost: A Scalable Tree Boosting System, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco California USA: ACM, 785–794 (2016).
[8] Turimov Mustapoevich, D., Muhamediyeva Tulkunovna, D., Safarova Ulmasovna, L., Primova, H. and Kim, W., Improved Cattle Disease Diagnosis Based on Fuzzy Logic Algorithms. Sensors, 23(4), 2107 (2023).
[9] Primova, H. A., Mukhamedieva, D. T. and Safarova, L., Application of Algorithm of Fuzzy Rule Conclusions in Determination of Animal's Diseases Journal of Physics: Conference Series, 2224(1), 012007 (2022).
[10] Safarova, L., Formation of informative signs for predicting the disease of highly productive cows with non-communicable diseases Journal of Physics: Conference Series, 1901(1), 012049 (2021).
[11] Primova, H., Yalgashev, O. and Safarova, L., Solution of the multi-criterial routing problem in telecommunication networks. International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities, ICISCT 2019, 9012045 (2019).
[12] Primova, H., Sotvoldiev, D. and Safarova , L., Approaches to solving the problem of risk assessment with fuzzy initial information 12th International Scientific and Technical Conference "Dynamics of Systems, Mechanisms and Machines", Dynamics 2018, 8601485 (2018).
[13] Dyakonov, A. G., C stacking (Stacking) and blending (Blending), 2017, https :// dyakonov . org / 2017 /03/10/ c stacking -and - blending - blending / (23 October 2023).
[14] Merkov, A. B., Pattern recognition: Construction and training of probabilistic models, –LENAND, Moscow (2020).
[15] Breiman, L., Random Forests, Machine Learning, 45, 1, 5–32 (2001).
[16] Mukhamedieva, D. T. and Safarova, L. U., Main problems and tasks of intellectualization of information processing system: International Journal of Innovative Technology and Exploring Engineering, 8(9 Special Issue 3), 158–165 (2019).
[17] Primova, H. and Safarova, L., The predictive model of disease diagnosis osteodystrophy cows using fuzzy mechanisms logic; AIP Conference Proceedings, 2365, 050005 (2021).
[18] Muhamediyeva, D., Safarova, L. and Tukhtamurodov, N., Neutrosophic Sets and Their Decision-Making Methods on the Example of Diagnosing Cattle Disease; International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities, ICISCT 2021 (2021).
[19] Muhamediyeva, D. T., Safarova, L. U. and Tukhtamurodov, N., Early diagnostics of animal diseases on the basis of modern information technologies AIP Conference Proceedings, 2817, 020038 (2023).
[20] Muhamediyeva, D. T., Safarova, L. U. and Tukhtamurodov, N., Building a fuzzy sugeno model for diagnosing cattle diseases on the basis of developing a knowledge base; AIP Conference Proceedings, 2817, 020037 (2023).