Flattening the Recall line using Voting classifier for Forest Cover Type data

K. M. Safin Kamal
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
2020-1-60-235@std.ewubd.edu

Mysha Maliha Priyanka
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
2020-1-60-230@std.ewubd.edu

Alfe sunny
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
2019-3-60-010@std.ewubd.edu

Maimuna akter Liza
Department of Computer Science & Engineering
East West University
Dhaka, Bangladesh
2019-2-60-088@std.ewubd.edu

***Abstract*—This paper focuses on the application of a Voting Classifier to address the challenge of flattening the recall line in the classification of Forest Cover Type data. Forest cover is crucial for biodiversity preservation and climate regulation, and accurate classification of forest cover types is essential for effective forest management. The paper utilizes a dataset containing attributes related to forest cover, and machine learning models such as Random Forests, K-Nearest Neighbors (KNN), Extra Trees, and XGBoost are employed. However, the performance of individual models may vary in terms of recall. To overcome this, a Voting Classifier is introduced, which combines the predictions from multiple models using a majority or weighted vote. The experiments demonstrate the effectiveness of the Voting Classifier in flattening the recall line and enhancing the accuracy of forest cover type classification.**

Keywords—Forest cover, voting classifier, RF, KNN, Extra Tree, XGBoost.

I. INTRODUCTION

The amount of land that is covered by forests is referred to as the forest cover, and it is essential for preserving biodiversity and regulating the climate. Sustainable development requires constant forestry cover management. The assessment and mapping of many types of forest cover, including temperate forests and tropical rainforests, is made possible by remote sensing technologies. Understanding where they are found makes it easier to put effective conservation measures into practice and encourage sustainable land use. In order to combat climate change, preserve wildlife habitats, and guarantee a healthy environment for future generations, it is essential to protect and restore forest cover. In order to address forest cover loss and promote sustainable forest management practices, international cooperation and policy are essential[1].

Researchers and practitioners often use this dataset to develop and evaluate machine learning models, such as decision trees, random forests, or gradient boosting algorithms, to accurately classify or predict the forest cover type based on the provided features.

Detecting the cover forest type dataset properly and understanding its characteristics can provide several benefits in terms of data analysis and modeling. Here are some potential benefits:

1. Ecological Understanding: The dataset provides valuable information about the predominant tree species in different forest areas. Proper detection and analysis of the dataset can contribute to a better understanding of forest ecosystems, including species distribution patterns, habitat suitability, and ecological processes.

2. Forest Management: Accurate classification of forest cover types can assist in effective forest management practices. It can help in identifying areas with specific tree species, which can inform decisions related to timber harvesting, reforestation efforts, biodiversity conservation, and ecosystem preservation.

3. Model Development and Evaluation: The dataset serves as a benchmark for developing and evaluating machine learning models and algorithms. By detecting the dataset properly, researchers can use it to train and test their models, allowing for comparisons and advancements in classification and prediction accuracy.

4. Feature Importance and Analysis: The dataset contains various geographic features that influence forest cover types. Detecting the dataset properly enables the exploration of feature importance, helping to identify the most influential variables in determining forest types. This analysis can provide insights into the ecological factors driving forest composition.

Regarding the measures used to evaluate the performance of models on the cover forest type dataset, commonly used metrics include accuracy, precision, recall, F1 score, and confusion matrix. These measures assess the model's ability to correctly classify each forest cover type and provide insights into its predictive capabilities.

In summary, properly detecting and utilizing the cover forest type dataset can contribute to ecological understanding, assist in forest management decisions, facilitate model development and evaluation, and provide insights into feature importance and analysis in relation to forest cover types.

II. LITERATURE REVIEW

Arvind Kumar and Nishant Sinha [1]presents the results of applying the Random Forests algorithm to classify forest cover types. The authors collected a dataset containing attributes related to forest cover and performed data preprocessing, including feature selection and normalization. The authors utilize the Random Forests algorithm, which is an ensemble learning method based on decision trees. The experimental results showed that the Random Forests algorithm achieved high accuracy in classifying forest cover types.

H. Sjöqvist, M. Längkvist, and F. Javed [2] presents an analysis of fast learning methods for classifying forest cover types. The objective of the study is to analyze and compare different fast learning methods for accurately classifying forest cover types. The authors investigate various fast learning algorithms, including Decision Trees, Random

Forests, Gradient Boosting, and Support Vector Machines (SVM). The Random Forest algorithm demonstrates superior accuracy compared to the other algorithms.

Blackard, J. A., & Dean, D. J. [3] compared the accuracies of artificial neural networks (ANNs) and discriminant analysis in predicting forest cover types based on cartographic variables. The study aims to assess and compare the prediction accuracies of two different modeling approaches, artificial neural networks (ANNs) and discriminant analysis, in predicting forest cover types using cartographic variables as input. A separate study had been conducted for the Colorado State Forest using remotely sensed data. That study achieved a classification accuracy of 71.1% for broad land cover classes

Yu Wang, Han Liu, Lingling Sang, and Jun Wang [4] focused on the application of ensemble learning techniques to analyze forest cover and landscape patterns using multi-source remote sensing data. The study addresses the importance of accurately assessing forest cover and landscape patterns for effective forest management and conservation. They used many machine learning approach like XGBoost, extra Tree, ligh GBM.

Literature Gap: Despite the extensive exploration of machine learning models for forest cover type classification in the reviewed papers, there are several literature gaps that need to be addressed.

1) Recall Evaluation: The literature gap lies in the absence of comprehensive analysis and reporting of recall scores for different models, hindering a complete understanding of their performance.

2) K-Nearest Neighbors (KNN): None of the reviewed papers consider the application of the K-Nearest Neighbors algorithm for forest cover type classification. Its absence in the reviewed papers creates a gap in understanding the potential benefits and performance of KNN in this specific domain.

3) Voting Classifier: The reviewed papers do not explore the utilization of a voting classifier, which combines the predictions of multiple individual classifiers.

4) Dataset Preprocessing: Although the reviewed papers mention the importance of data preprocessing but did not provide any detailed information on the specific preprocessing techniques applied to the forest cover datasets.

III. METHODOLOGY

In this project, we used the Python programming language to implement and train a machine learning-based model. The dataset in discussion is a compilation of forest cover data. These data will serve as a tool for technique learning and will produce useful information. Following Fig- shows the project's working process.

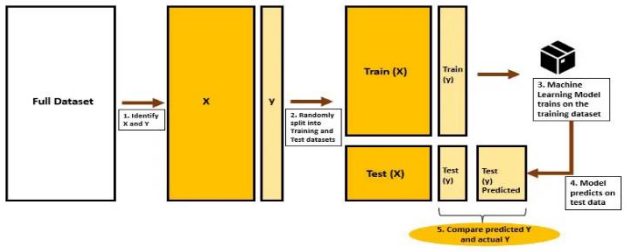

Figure 1: working process

A. *Dataset description*

This dataset contains tree observations from four areas of the Roosevelt National Forest in Colorado. All observations are cartographic variables (no remote sensing) from 30-meter x 30-meter sections of forest. There are over half a million measurements total. This dataset includes information on tree type, shadow coverage, distance to nearby landmarks (roads etcetera), soil type, local topography and cover type. The dataset comprehends 54 cartographic variables plus the class label, 10 of which are quantitative features while the remaining 44 correspond to 2 qualitative variables one-hot encoded. In this work our goal is to classify the cover type based on predictor variables of each observation.

We briefly present the information regarding the meaning of some Columns and its admissible values:

Table 1: Columns and description

Columns	**Description**
Elevation	in meters
Aspect	in degrees azimuth
Slope	in degrees.
Distance_To_ Hydrology	in meters to nearest surface water features.
Horizontal_Distance To_Roadways	in meters to the nearest roadway.
Hillshade	index at day time, summer solstice. Value out of 255.
Horizontal_Distance _To_Fire_Point	in meters to nearest wildfire ignition points.
Wilderness_Area#	wilderness area designation
Soil_Type#	Soil type designation.
Cover_type (Target)	Forest cover (scaled 1 to 7)

In "Cover_Type", which is our target column, forest cover type designation, its possible values are between 1 and 7, mapped in the following way:

1. Spruce/Fir
2. Lodgepole Pine
3. Ponderosa Pine
4. Cottonwood/Willow
5. Aspen
6. Douglas-fir
7. Krummholz

We can see that highest number tree are type 2 (Lodgepole Pine) and lowest is type 4 (Cottonwood/Willow). There are four wilderness area: Neota (area 2) probably has the highest mean elevational value of the 4 wilderness areas. Rawah (area 1) and Comanche Peak (area 3) would have a lower mean elevational value, while Cache la Poudre (area 4) would have the lowest mean elevational value.

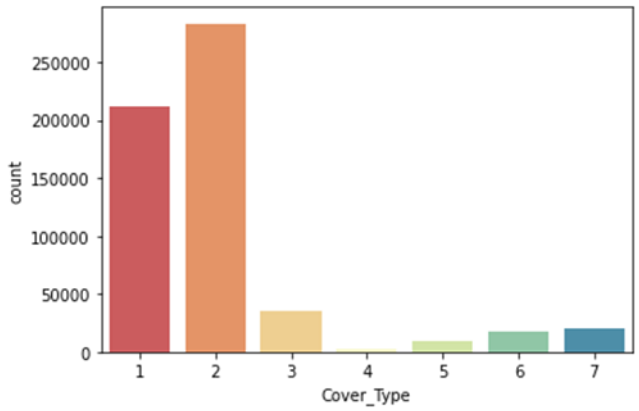

Figure 2: distribution of cover type

Soil_Type feature is encoded as 40 one-hot encoded and Wilderness_Area feature is one-hot encoded to 4 binary columns (0 = absence or 1 = presence).

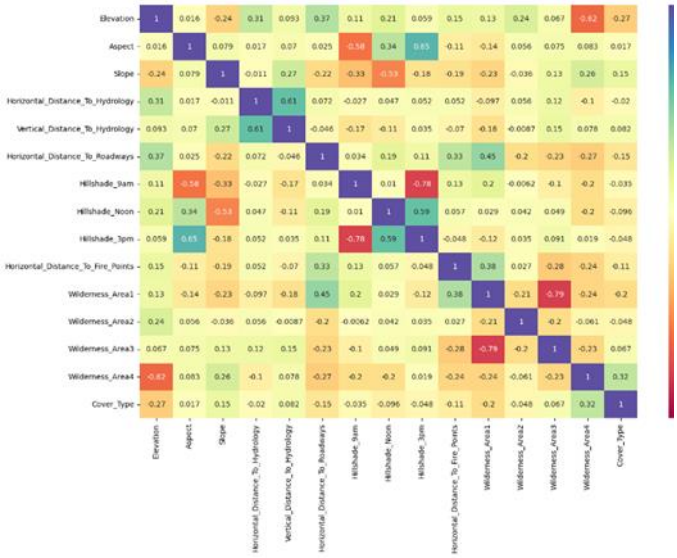

Figure 3: correlation between features

B. *Dataset preprocessing*

We reversed the one-hot encoding of categorical variables into label encoding to obtain a database better suited for tree-based models fitting and qualitative features plotting. Here we can see soil_type# is one-hot encoded to 40 binary columns (0 = absence or 1 = presence). So, we have reversed the on-hot encode so that we get a stable and clean dataset. Before reverse the shape of the dataset was (581012, 55) but after reverse the shape became (581012, 16).

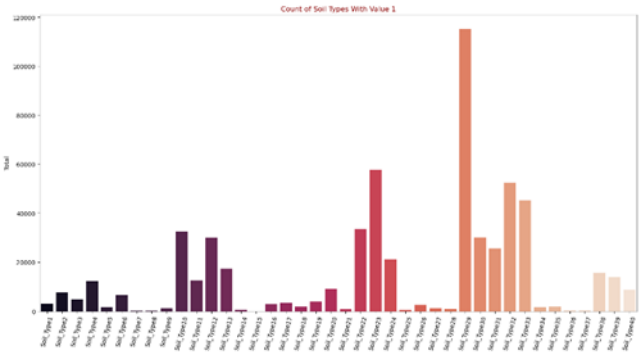

Figure 4: distribution of soil type

C. *Models Specification:*

We have used total 6 Machine learning models. Here we are discussing about them:

- Decision tree:

Decision tree is a widely used supervised machine learning algorithm for classification and regression tasks. It partitions the data recursively based on different features, aiming to create subsets that are similar in terms of the target variable. The algorithm selects the best feature to split the data at each step using metrics like information gain or Gini impurity, which measure the reduction in uncertainty or impurity achieved by the split.

The construction of the tree continues until certain stopping criteria are met, such as reaching a maximum depth or no further improvement in reducing impurity. Once the tree is built, it can be used to make predictions by traversing the tree from the root node to a leaf node based on the input features' values. Decision trees have advantages such as interpretability and handling of categorical, numerical, and missing values. However, they can overfit the data if the tree becomes too complex. Techniques like pruning, random forests, or gradient boosting can address this issue.

- Random forest:

Random Forest is an ensemble algorithm that combines multiple decision trees for classification and regression tasks. It creates independent trees using different subsets of data and features. Each tree predicts an outcome, and the final prediction is determined by majority voting or averaging. The algorithm's randomness reduces overfitting and enhances generalization by introducing variation in training. Random Forests can handle high-dimensional data, large datasets, and both categorical and numerical features. They are less prone to overfitting compared to individual decision trees and provide insights into feature importance. Overall, Random Forests are powerful and versatile algorithms used in finance, healthcare, image classification, and other domains.

- K-Nearest Neighbors:

K-Nearest Neighbors (KNN) is a simple yet powerful algorithm used for both classification and regression tasks in machine learning. It is a non-parametric and instance-based learning algorithm, meaning that it doesn't make any assumptions about the underlying data distribution and stores the entire training dataset in memory for making predictions.

KNN works by finding the K nearest neighbors to a new data point based on a distance metric, and then making predictions based on the majority vote (classification) or averaging (regression) of the neighbors' labels or target values. However, KNN has several advantages, including simplicity, interpretability, and effectiveness in certain types of datasets. However, it also has some limitations, such as the computational cost of searching for nearest neighbors in large datasets and the need for appropriate feature scaling. Overall, KNN is a versatile algorithm that can be applied to various machine learning tasks.

- XGBoost:

XGBoost (eXtreme Gradient Boosting) is a popular and powerful machine learning algorithm that belongs to the ensemble learning category. It is widely used for both classification and regression tasks and has proven to be highly effective in various domains, including Kaggle competitions and real-world applications [5].

XGBoost initializes the model with an initial prediction and iteratively trains a sequence of decision trees to minimize the specified loss function. Each tree is added to the ensemble by learning the optimal structure using gradient descent, combining the predictions of all trees to make the final prediction. XGBoost offers several advantages, including high prediction accuracy, scalability, and efficient parallel processing. It also supports various advanced features, such as handling missing values, handling categorical variables, and early stopping to prevent overfitting.

- Extra tree:

Extra Trees, or Extremely Randomized Trees, is an ensemble algorithm that builds a collection of decision trees. It is similar to Random Forest but introduces additional randomness during the tree construction process. This randomness makes Extra Trees faster, more efficient, and less prone to overfitting. The predictions are obtained by averaging the outputs of all trees for regression or through majority voting for classification. Extra Trees are especially useful for high-dimensional and noisy data, handling both categorical and numerical features, and being robust against outliers. However, they may not provide feature importance measures[4]. So, Extra Trees offer computational efficiency, reduced overfitting, and are well-suited for challenging datasets.

- Voting classifier:

A Voting Classifier is an ensemble learning technique in machine learning where multiple models, often of different types, are combined to make predictions. It works by aggregating the predictions from each individual model and using a majority vote or weighted vote to determine the final prediction.

Voting Classifiers can be used for both classification and regression tasks. They leverage the idea that combining multiple models can lead to better overall performance by capturing different aspects of the data and reducing individual model biases. This ensemble technique can improve the accuracy, robustness, and generalization ability of the model.

Voting Classifiers can incorporate a variety of models such as decision trees, support vector machines, logistic regression, or any other model that supports the concept of predicting class labels or probabilities. It is particularly useful when individual models have complementary strengths and weaknesses.

The voting classifier is our main focus. We used other 5 models in this voting machine.

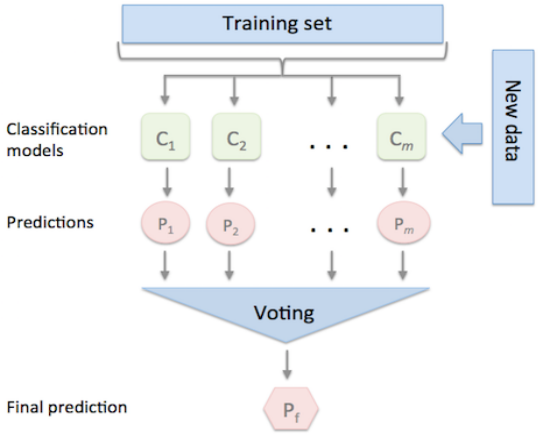

Figure 5: work process of voting classifier

IV. RESULTS AND DISCUSSION

After the training and testing the models we have got those result:

Table 2: Class Wise Recall Score for Each model

Classes /label	RF	Decision Tree	KNN	XG Boost	Extra Tree	Voting Classi-fier	Voting Classifier (Weighted Recall)
1	0.94	0.93	0.91	0.84	0.94	0.78	0.78
2	0.96	0.94	0.94	0.90	0.97	0.97	0.97
3	0.95	0.92	0.89	0.91	0.96	0.86	0.86
4	0.84	0.81	0.72	0.85	0.84	0.84	0.84
5	0.75	0.82	0.76	0.60	0.79	0.92	0.92
6	0.88	0.85	0.78	0.82	0.90	0.96	0.96
7	0.93	0.94	0.92	0.90	0.95	0.98	0.98

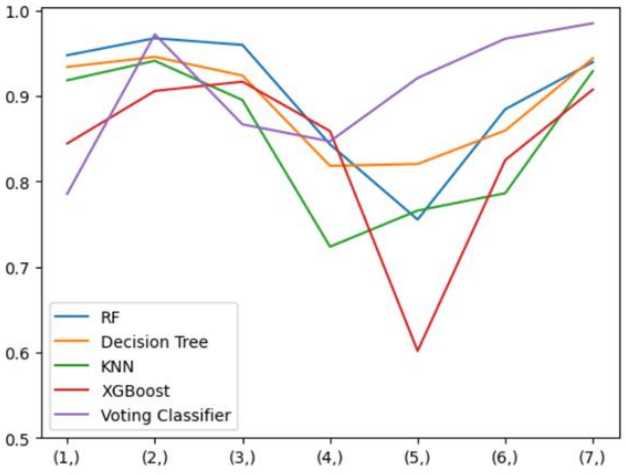

Figure 6: Class Wise Recall Score for Each model

Based on the provided table, we have recorded the recall values for different machine learning models over multiple classes. Recall, also known as true positive rate or sensitivity, measures the ability of a model to correctly identify positive instances from all actual positive instances. Let's analyze the results and provide a broad report:

1) Random Forest: The Random Forest model achieves high recall values across the classes, ranging from 0.918 to 0.948. It shows consistently good performance in correctly identifying positive instances.

2) Decision Tree: The Decision Tree model also performs well with recall values ranging from 0.819 to 0.944. It demonstrates good performance, although slightly lower compared to the Random Forest model.

3) KNN: The KNN model has recall values ranging from 0.724 to 0.941. It performs reasonably well but shows some variability across the classes.

4) XGBoost: The XGBoost model achieves recall values ranging from 0.602 to 0.906. It shows good performance overall, but with some variability across the classes.

5) ExtraTree Classifier: The Extra Tree model achieves recall values ranging from 0.793 to 0.972. It demonstrates good performance, similar to the Random Forest and Decision Tree models.

6) Voting Classifier: The Voting Classifier model, without any specified weights, achieves recall values ranging from 0.786 to 0.967. It combines multiple models and shows good performance.

7) Voting Classifier (Weighted): The Voting Classifier model with weighted votes achieves recall values ranging from 0.786 to 0.97. There's no difference between Classifier based on Majority voting and Weighted Voting .

A. Low Recall Vs High Recall Over Classes:

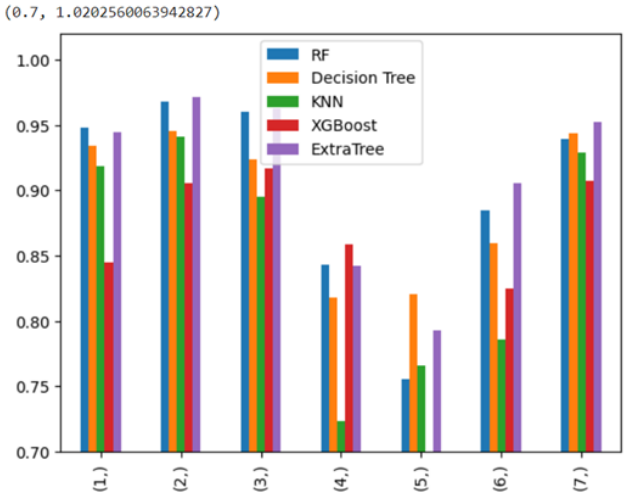

Figure 7: Class Wise Recall Score for Each model

a) Low Recall Scores:

1) Class 4: Across most models, Class 4 consistently has relatively low recall scores. It ranges from 0.723 to 0.858 in different models.

b) High Recall Scores:

1) Class 2: Class 2 generally has high recall scores across most models. It ranges from 0.818 to 0.945, showing good performance in correctly identifying positive instances.

2) Class 3: Similar to Class 2, Class 3 also demonstrates high recall scores in most models. It ranges from 0.724 to 0.941, indicating good performance in correctly identifying positive instances.

3) Class 6: Class 6 shows consistently high recall scores across most models, ranging from 0.786 to 0.967.

4) Class 7: Similar to Class 6, Class 7 also demonstrates high recall scores in most models. It ranges from 0.785 to 0.985, indicating good performance in correctly identifying positive instances.

B. Performane Analysis of voting Classifer

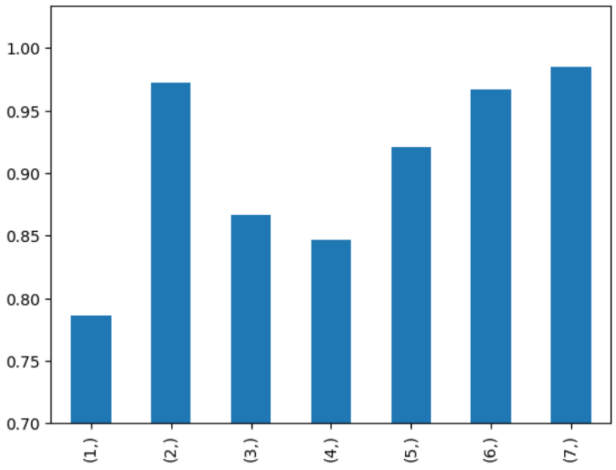

Figure 8: Class Wise Recall Score for Voting Classifier

From the provided table, it can be observed that the Voting Classifier consistently achieves better recall scores in the low recall classes compared to individual models such as RandomForest, Decision Tree, KNN, XGBoost, ExtraTree, and others. This indicates that the Voting Classifier's ensemble approach, which combines the predictions of multiple individual models, is effective in improving the recall performance for these classes.

Specifically, for the low recall classes (such as Class 4 and Class 5), the recall scores of the Voting Classifier are generally higher compared to the recall scores of the individual models. This suggests that the Voting Classifier is able to leverage the strengths of different models and make more accurate predictions for these challenging classes.

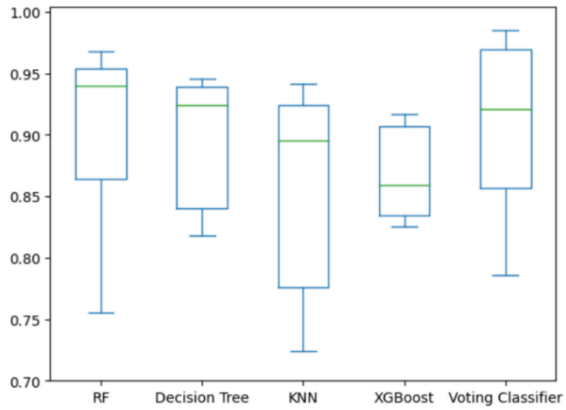

Figure 9: Recall Improvement for Voting Classifier

The Voting Classifier's ability to outperform individual models in terms of recall for low recall classes can be attributed to the ensemble's ability to reduce bias and variance, improve generalization, and capture diverse patterns present in the data. By combining the predictions of multiple models, the Voting Classifier is able to make more robust and reliable predictions, particularly for classes that individual models may struggle with.

Overall, the Voting Classifier demonstrates its effectiveness in improving the recall scores for low recall classes, showcasing the benefits of ensemble learning in handling challenging classification tasks.

V. COLCUSION

In summary, the analysis of recall values for different machine learning models reveals that Random Forest, Decision Tree, ExtraTree Classifier, and Voting Classifier perform well in identifying positive instances. KNN and XGBoost models also show reasonable performance. However, Class 4 consistently has low recall scores, while Class 2, Class 3, Class 6, and Class 7 demonstrate high recall scores. The Voting Classifier, with its ensemble approach, consistently improves recall performance, highlighting the benefits of combining multiple models. Overall, these findings emphasize the importance of model selection and ensemble techniques in achieving accurate positive instance identification of the forest cover type dataset.

REFERENCES

[1] Kumar, A., Sinha, N. (2020). Classification of Forest Cover Type Using Random Forests Algorithm. In: Kolhe, M., Tiwari, S., Trivedi, M., Mishra, K. (eds) Advances in Data and Information Sciences. Lecture Notes in Networks and Systems, vol 94. Springer, Singapore.

[2] Sjöqvist, H., Längkvist, M., & Javed, F. (2020). An Analysis of Fast Learning Methods for Classifying Forest Cover Types. Journal of Intelligent Systems, 29(3), 691-709.

[3] Blackard, J. A., & Dean, D. J. (1999). Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Remote Sensing and GIS Program, Department of Forest Sciences, 113 Forestry Building, Colorado State University, Fort Collins, CO 80523, USA.

[4] Wang, Y., Liu, H., Sang, L., & Wang, J. (2022). Characterizing Forest Cover and Landscape Pattern Using Multi-Source Remote Sensing Data with Ensemble Learning. Remote Sensing, 14(21), 5470.

[5] H. Xu, G. Pang, Y. Wang and Y. Wang, "Deep Isolation Forest for Anomaly Detection," in IEEE Transactions on Knowledge and Data Engineering.

[6] Guo, G., Wang, H., Bell, D. A., & Bi, Y. (2004). KNN Model-Based Approach in Classification. ResearchGate.

[7] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2019). A Comparative Analysis of XGBoost. ResearchGate. Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2019). A Comparative Analysis of XGBoost. ResearchGate.