Classification of Illegal Fishing

Name:	**Rasel Mondal**
Registration No./Roll No.:	2311004
Institute/University Name:	IISER, Bhopal
Program/Stream:	DSE
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction

Illegal fishing poses a serious threat to the delicate balance of marine ecosystems and threatens the livelihoods of countless people who depend on the world's waters. This multifaceted problem is characterized by overfishing, unreported fishing, and unauthorized access to restricted areas, and affects the environment, the economy, and society in general. The consequences of this extend not only to the depletion of marine natural resources but also to species extinction and long-term threats to the health of our oceans, exacerbated by the continued scarcity of marine natural resources.[1]

To address this global challenge, we use Automatic Identification System (AIS) vessel tracking as the cornerstone of our initiative. These tracks provide valuable information about the behavior of ships at sea and provide critical information about the position, movement, and other important characteristics of ships. Using the power of machine learning, we aim to develop a system that can independently identify and classify illegal fishing incidents. By leveraging advanced technology, we aim to strengthen the protection of our seas and strengthen enforcement of fishing regulations, ushering in a new era of sustainable development for our oceans.[2]

1.1 Data Description

A large oceanic dataset with **838,860 rows** and **8 columns** makes up the provided training dataset; each row represents a distinct feature vectors. This dataset is an important source of information for developing machine-learning algorithms that identify and classify illegal fishing. To assist in classifying the observed ship actions, the data comprises three distinct class labels in addition to a number of other numerical factors. The following are those classification marks:

- **-1 (No Class Label):** Instances denoted by this label do not have a particular task assignment. These data points do not relate to actions related to fishing or non-fishing.
- **0 (Not Fishing):** This "0" label indicates that the vessels under observation are not presently involved in any fishing activities.
- **1 (Fishing):** Instances labeled as "1" correspond to vessels actively involved in fishing activities. This class identifies and differentiates fishing-related behaviors from other maritime actions.

2 Methods

2.1 Data Preprocessing

Three of the *speed* and *course* features in the training dataset had six missing values. The mean imputation method was applied in order to deal with the missing values. A basic imputation technique called the mean method involves substituting the mean value of the remaining data points in a feature for any missing values in a dataset. One of the simplest methods to address missing data while maintaining the dataset's overall statistical characteristics is to employ this technique, which is frequently applied when working with numerical data.

2.2 Exploratory Data Analysis (EDA)

To gain a comprehensive understanding of the dataset and its features, an exploratory data analysis (EDA) was conducted.

Class	Number of Instances
-1	802828
0	30237
1	5795

This necessitated an in-depth analysis of the dataset generated by combining the training data and the associated class labels. Evidence-based analysis (EDA) facilitates the identification of insights, trends, and correlations between features, thus allowing for more informed decisions in subsequent modeling operations.

2.3 Model Training

The current classification problem requires that suitable machine learning models be chosen and trained in order to classify the target variables into discrete values (-1, 0, 1). During this training stage, four supervised machine-learning classification models, Decision Tree, Random Forest, Adaptive Boosting, and Logistic Regression were used.
To fine-tune and optimize model performance, the GridSearchCV technique was utilized. GridSearchCV is a method that systematically tests various hyperparameters and selects the optimal combination of hyperparameters by employing either k-fold or stratified k-fold cross-validation. This enables the identification of the most effective configuration for each model, resulting in enhanced classification accuracy and predictive power.

3 Experimental Setup

3.1 Evaluation Criteria

Various evaluation criteria such as precision, recall, and F1 score were used to evaluate the performance of the model. These metrics provide a comprehensive understanding of the model and the ability to correctly identify fishing, not fishing, and no-label cases, accounting for both false positives and false negatives.

3.2 Experimental Setting

Experiments were performed with several Python libraries such as numpy, pandas, matplotlib, scikit-learn, unbalanced learning, etc. The dataset was divided into training (80%) and testing (20%) sets, and models were evaluated using k-fold, stratified k-fold (5-fold) cross-validation to ensure reliable results with two feature selection methods namely SelectFromModel, Recursive Feature Elimination (RFE).

4 Results and Analysis

Table 1: Performance of Different Classifiers using 3 labels without K-fold

Classifier	Precision	Recall	Accuracy	F-measure
Adaptive Boosting	0.9068	0.9401	**0.99**	**0.9226**
Logistic Regression	0.3190	0.3333	**0.96**	**0.3260**
Random Forest	0.9453	0.9444	**1**	**0.9448**

Table 2: Performance of Different Classifiers with 3 labels using SelectKBest

Classifier	Precision	Recall	Accuracy	F-measure
Adaptive Boosting	0.9462	0.9439	**1**	**0.9450**
Decision Tree	0.9387	0.9471	**1**	**0.9429**
Random Forest	0.9574	0.9532	**1**	**0.9553**

Table 3: Performance of Different Classifiers with 3 labels using GridsearchCV

Classifier	Precision	Recall	Accuracy	F-measure
Adaptive Boosting	0.9411	0.9371	**1**	**0.9391**
Decision Tree	0.9455	0.9269	**1**	**0.9359**
Random Forest	0.9301	0.9631	**1**	**0.9463**

Table 4: Performance of Different Classifiers with 2 labels using GridSeaechCV without k-fold

Classifier	Precision	Recall	Accuracy	F-measure
Adaptive Boosting	0.9060	0.8610	**0.94**	**0.8814**
Decision Tree	0.8914	0.8938	**0.94**	**0.8926**
Random Forest	0.8939	0.9321	**0.95**	**0.9115**

Table 5: Performance of Different Classifiers using with 2 labels using GridSearchCV with K-fold

Classifier	Precision	Recall	Accuracy	F-measure
Adaptive Boosting	0.9442	0.9501	**0.97**	**0.9471**
Decision Tree	0.9452	0.9547	**0.97**	**0.9499**
Random Forest	0.9638	0.9603	**0.98**	**0.9620**

4.1 Analysis

- 3-class classification problem was converted to a 2-class classification problem by dropping one class, namely, -1, because this class label indicated the absence of any class assignment so had very little or no significance.
- Random Forest exhibits outstanding performance on Dataset 2, which has 5 features and 2 classes. It attains impressive precision, recall, F1-score, and accuracy, showcasing a balanced performance across these metrics. The model effectively identifies and classifies instances from both classes without bias. The optimal parameters for training the model include using the criterion 'entropy', setting the maximum depth to 50, employing a minimum samples leaf of 1, specifying a minimum samples split of 2, utilizing 100 estimators, and fixing the random state

at 0. This configuration underscores the model's ability to capture intricate patterns within the dataset, leading to its exceptional performance in classification tasks with a binary class distribution.

- Decision Tree excels on Dataset 5, characterized by 5 features and 3 classes, as it attains remarkable precision, recall, F1-score, and accuracy. The model's robust performance across these metrics indicates its ability to effectively distinguish and categorize instances among the three classes without showing bias towards any particular class. The optimal configuration for training this model includes using the criterion 'entropy', setting the maximum depth to 50, employing a minimum samples leaf of 1, specifying a minimum samples split of 2, utilizing 100 estimators, and fixing the random state at 0. This configuration demonstrates the model's ability to capture complex relationships within the dataset, leading to its exceptional performance on classification tasks with multiple classes and diverse feature sets.

- It is observed that selecting 5 features is giving the best results in terms of all evaluation criteria, irrespective of the number of classes.

5 Conclusion

In conclusion, our analysis of classification models, focusing on Random Forest and Decision Tree algorithms in combating illegal fishing, reveals valuable insights. Transforming a 3-class problem into a 2-class scenario enhances interpretability. Random Forest excels on Dataset 2, demonstrating balanced and impressive performance in identifying instances in a binary class distribution. Its optimal parameters highlight its ability to capture intricate patterns crucial for effective illegal fishing classification. Conversely, the Decision Tree excels on Dataset 5, showcasing remarkable precision, recall, F1-score, and accuracy in handling multiple classes and diverse features. The consistent superiority of models with 5 features underscores the importance of feature selection for optimal results, suggesting their efficacy in addressing the challenges of combating illegal fishing activities.

References

[1] P. Sowjanya P.V.S. Keerthana K. Tejaswini B. Padmaja, K. Mounika. Catching illegal fishing using random forest and linear regression models. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 10, 2022.

[2] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.