Classification of Illegal Fishing

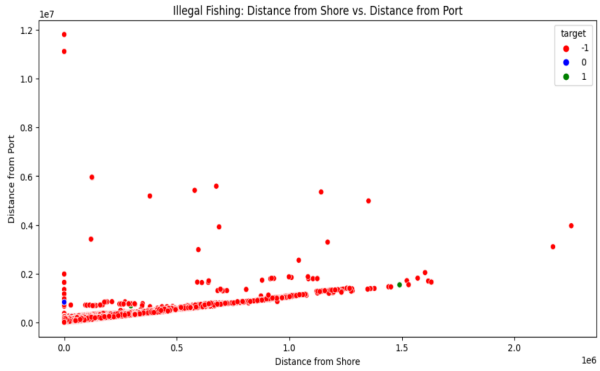
Name:	**Ishan Samir Thoke**
Registration No./Roll No.:	21328
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	August, 2023
Date of Submission:	November 19, 2023

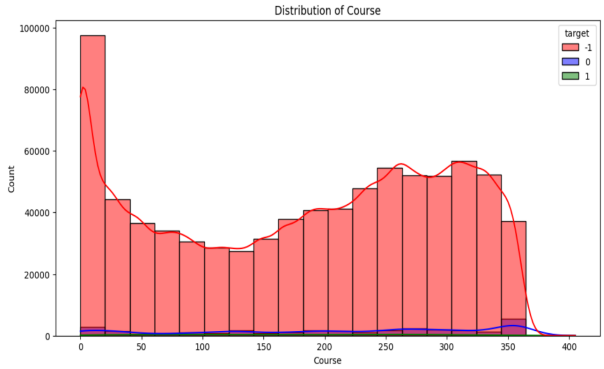
1 Introduction

Problem Statement: The objective is to develop supervised machine learning framework to identify illegal fishing.

Our dataset contains 8 features : *mmsi, timestamp, distance_from_shore, distance_from_port, speed, course, latitude, longitude.* Data points (8,38,860 of them) are classified into 3 classes : *-1, 0, 1.*

We plot the features : 'distance from shore vs distance from port' and 'Distribution of Course'

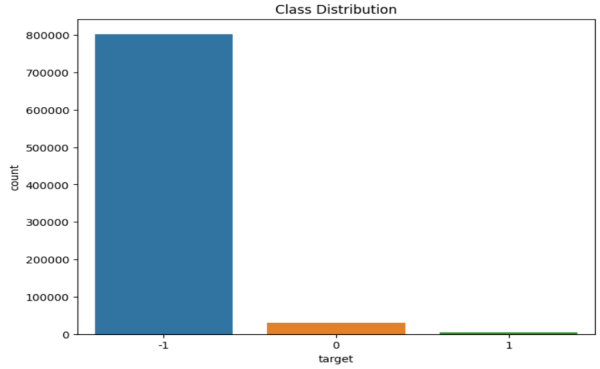


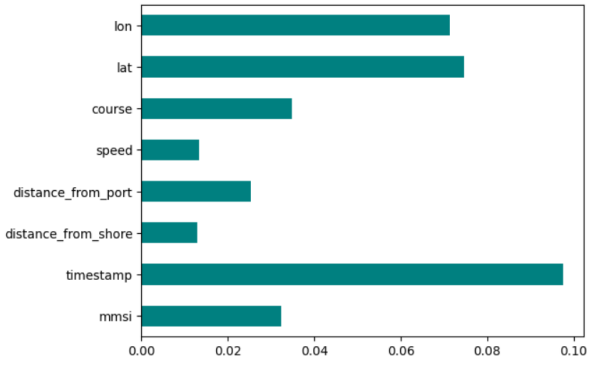


Plan of Action

Feature Selection and Class Imbalance
Classification Techniques
Performance Evaluation

The plot on the left depicts the class distribution (evidence of class imbalance) in the training data and the one on the right is the plot obtained by running feature selection





2 Methods

In this section, we elaborate on the methods employed in our study, including the proposed techniques and the exploration of various methods. We also discuss the concept of parameter tuning and relevant references.

Proposed Methods

- **Feature Selection**: Features with high mutual information scores are deemed more relevant to the task of identifying illegal fishing events.
- **Class Imbalance Handling**: We use the Synthetic Minority Over-sampling Technique with Edited Nearest Neighbors (SMOTEENN) to oversample minority classes and undersample the majority class, achieving a more balanced dataset.
- **Model Selection and Tuning**: We explore a range of classification models, including Decision Trees and Random Forests, and fine-tune their hyper-parameters to optimize their performance.

Parameter Tuning

Parameter tuning is a critical aspect of optimizing the performance of machine learning models. We follow a systematic approach to fine-tuning hyper-parameters [1, 2].
We employ techniques such as grid search and random search to search through hyper-parameter combinations and select the best configuration for our models. This ensures that our models are well-calibrated and capable of making accurate predictions.
By systematically exploring different methods and optimizing parameters, we aim to develop robust models that can effectively detect illegal fishing events from vessel tracking data.

Exploration of Methods

While we have discussed our proposed methods in detail, it is important to note that we also explored existing classification techniques (e.g., SVM [3], k-means, ensemble techniques, etc.) for experimental analysis. Our focus remains on the techniques and approaches that we have customized and fine-tuned to address the specific challenges posed by the task of identifying illegal fishing events. These methods aim to provide a more tailored and effective solution to the problem at hand.

3 Experimental Setup

In our experimental setup, we utilize state-of-the-art models of various models like *Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, K-Nearest Neighbours and Support Vector Machine*
A range of evaluation criteria is employed to assess the performance of these models, it's primary metrics include: *Precision, Recall and F-Measure*

To optimize the state-of-the-art models, we focus on tuning significant parameters or **hyper-parameters**. In particular, we fine-tune parameters such as **maximum depth, minimum samples for splitting, and minimum samples at leaf nodes** for **Decision Tree models**. For **Random Forest models**, we explore parameters including the **number of estimators, maximum depth, and minimum samples for splitting and leaf nodes**.

Our experimental setup is designed to comprehensively evaluate model performance, emphasizing **precision, recall, and F1-score**, and harnessing the flexibility of **scikit-learn**[1] to implement and optimize state-of-the-art models. We then choose the best model and thereby make predictions on the test data to obtain the required labels.

[1] https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

4 Results and Discussion

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Precision	Recall	F-measure
Logistic Regression	0.32	0.33	0.33
Decision Tree	0.94	0.94	0.94
Random Forest	0.95	0.93	0.94
Gradient Boosting	0.90	0.80	0.85
K-Nearest Neighbors	0.96	0.95	0.96
Support Vector Machine	0.32	0.33	0.33

Table 2: Using Feature Selection and SMOTEENN

Classifier	Precision	Recall	F-measure
Logistic Regression	0.32	0.33	0.33
Decision Tree	0.76	0.97	0.84
Random Forest	0.79	0.98	0.86
Gradient Boosting	0.52	0.92	0.60
K-Nearest Neighbors	0.46	0.80	0.51
Support Vector Machine	0.92	0.96	0.94

Table 3: Confusion Matrices of Different Classifiers

Actual Class	Predicted Class		
	-1	0	1
-1	160489	0	0
0	6128	0	0
1	1155	0	0

Logistic Regression

Actual Class	Predicted Class		
	-1	0	1
-1	157575	2114	800
0	52	5993	83
1	13	53	1089

Decision Tree

Actual Class	Predicted Class		
	-1	0	1
-1	158375	1376	738
0	27	5986	115
1	10	34	1111

Random Forest

Actual Class	Predicted Class		
	-1	0	1
-1	145927	11591	2971
0	3	5618	507
1	1	71	1083

Gradient Boosting

Actual Class	Predicted Class		
	-1	0	1
-1	141844	14740	3905
0	195	4644	1289
1	36	284	871

K N N

Actual Class	Predicted Class		
	-1	0	1
-1	160489	0	0
0	6128	0	0
1	1155	0	0

S V M

Table 1: Decision Tree and Random Forest showcase robust classification, outperforming other models. Gradient Boosting shows slightly lower recall, indicating potential challenges in identifying certain instances. KNN performs well across all metrics. SVM and Logistic Regression, demonstrates lower performance metrics.

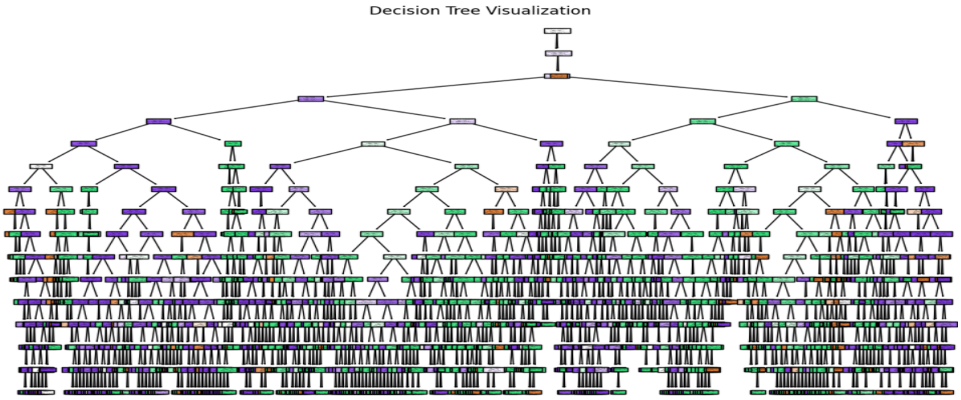
Table 2: Decision Tree's precision drops, but recall and F-measure remain acceptable. Random Forest shows improvements across all metrics, indicating positive effects of feature selection and sampling. Gradient Boosting improves in precision, while recall and F-measure remain comparatively lower. K-Nearest Neighbors experiences significant drops in all metrics. SVM and Logistic Regression's performance remains sub-optimal.

Table 3 provides the confusion matrices for different classifiers. The confusion matrices offer insights into the specific class-wise performance of each model.

Overall, Decision Tree and Random Forest consistently perform well, while the impact on other models varies.

Parameter Tuning : Decision Tree

In addition to evaluating the performance of various classifiers, we conducted a hyperparameter tuning analysis using **GridSearchCV** for the Decision Tree classifier. The optimal hyperparameters, determined based on accuracy, are as follows: *a maximum depth of 16, minimum samples per leaf set to 1, and minimum samples for split set to 2.*



Parameter Tuning : Random Forest

For the Random Forest classifier, we conducted a hyperparameter search using **RandomizedSearchCV** to optimize its performance. The optimal hyperparameters, which maximize accuracy, are as follows: *80 estimators, minimum samples per split set to 5, minimum samples per leaf set to 1, 'sqrt' for the maximum features, a maximum depth of 20, and bootstrap set to False.* These hyperparameters, identified through a randomized search process, contribute to the impressive performance.

5 Conclusion

In conclusion, our study comprehensively evaluated multiple machine learning classifiers for the given classification problem. Notably, the Decision Tree & Random Forest model, with optimized hyperparameters obtained through GridSearchCV & RandomizedSearchCV respectively, demonstrated superior performance, achieving a remarkable results. The feature selection and SMOTEENN techniques further improved the precision, recall, and F-measure for various classifiers.

Future work could delve into exploring advanced ensemble methods, incorporating more sophisticated feature engineering, and expanding the dataset for a more robust analysis. These endeavors aim to enhance model generalization and contribute to the continuous evolution of effective classification strategies in the domain.

GitHub: `https://github.com/Ishan-Thoke/ECS308-DSML-Classification-of-Illegal-Fishing`

References

[1] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[2] Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. *The Elements of Statistical Learning.* Springer, second edition, 2008.

[3] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.