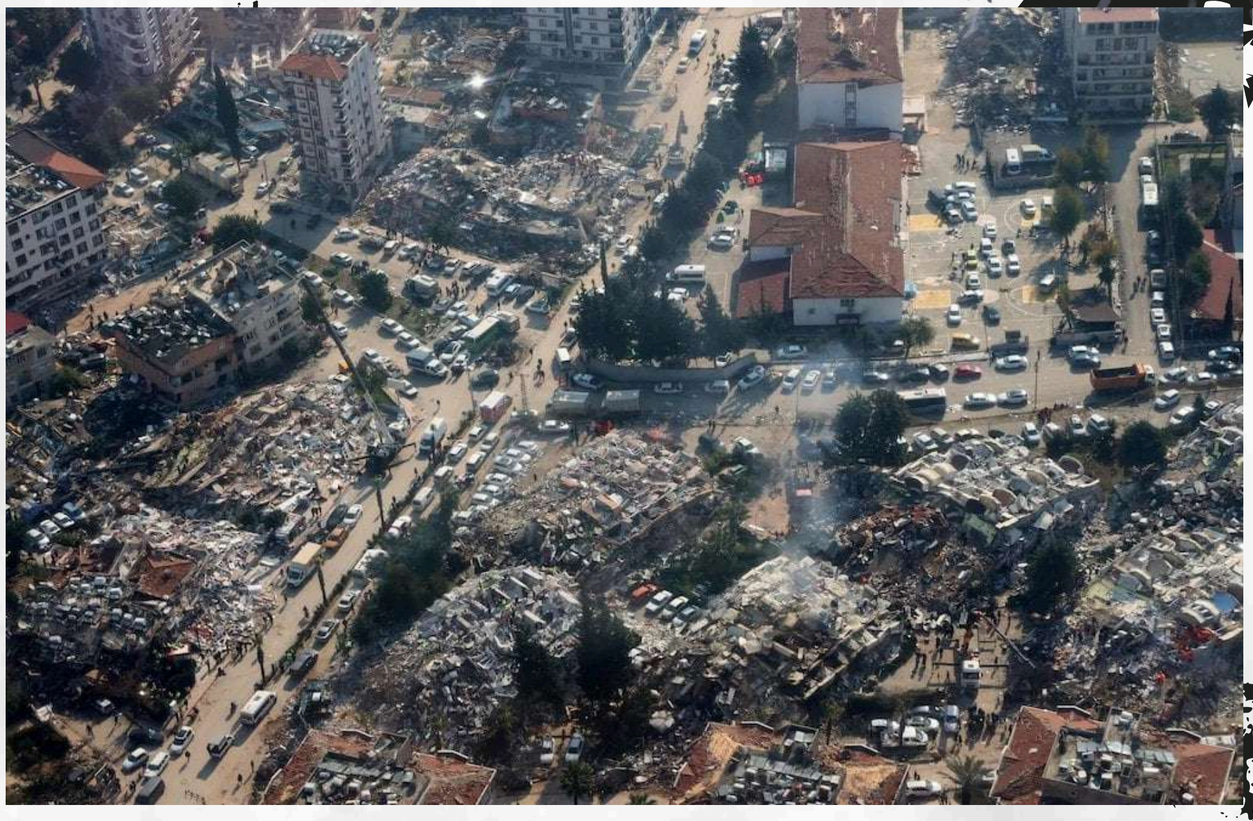
EARTHQUAKE EVENT CLASSIFICATION USING MACHINE LEARNING

Bilal Erçin | Baki Turhan

INTRODUCTION

Earthquakes are unpredictable natural disasters with significant impacts. By using machine learning to classify earthquake events based on seismic data, we can better understand their severity and improve disaster preparedness.



OBJECTIVE

This project aims to analyze patterns in earthquake attributes to support risk assessment and early response efforts.

DATASET OVERVIEW

- **Number of Rows:** 7,717
- **Number of Columns:** 23

Target Variable: **Alert**

Represents the alert level for earthquakes.

- Green
- Yellow
- Orange
- Red

DATASET OVERVIEW

Feature Variables:

Numerical Features:

- mag (magnitude),
- depth
- latitude
- longitude

Feature Enginnering

Created new features:

- **impact_score**: Multiplication of magnitude (mag) and depth (depth) to represent earthquake impact.
- **log_depth**: Logarithmic transformation of depth.

Class Imbalance Handling with SMOTE

Problem Identified:

The target variable **Alert** exhibited class imbalance, where certain classes (e.g., "green") had significantly more samples than others (e.g., "yellow", "orange", "red"). This imbalance could lead to biased models that favor the majority class, reducing the predictive power for minority classes.

Solution Applied (SMOTE):

SMOTE (Synthetic Minority Over-sampling Technique) was applied to address this issue.

SMOTE works by:

- Generating synthetic samples for the minority classes rather than duplicating existing ones.
- Creating new data points along the line segments connecting existing minority class samples.
- Ensuring that the dataset becomes balanced, providing the model with an equal opportunity to learn patterns from all classes.

Outcome:

After applying SMOTE, the target variable's class distribution became balanced, enabling fair representation of all classes during model training.

Data Splitting and Scaling

Data Splitting:

- The dataset was divided into:
  - Training Set (80%): Used to train the machine learning models.
  - Testing Set (20%): Held back for evaluation purposes to assess the generalization capability of the trained models on unseen data.

This split ensures the model is tested on data it has never seen during training, providing a realistic estimate of its performance in real-world scenarios.

Feature Scaling (StandardScaler):

- Why Scaling is Needed:
  - Features like depth and magnitude can have vastly different ranges. For example:
    - depth might range from 0 to hundreds of kilometers.
    - magnitude might range from 1 to around 10. Such differences can negatively impact models sensitive to feature magnitudes.

Outcome:

After scaling, all numerical features had a mean of 0 and a standard deviation of 1, making the dataset more suitable for training machine learning models.


Feature Correlation Heatmap
mag
depth
latitude
longitude
impact_score
log_depth
1.00 0.14 -0.20 0.18 0.19 0.18
0.14 1.00 -0.07 -0.01 0.99 0.82
-0.20 -0.07 1.00 -0.00 -0.07 -0.06
0.18 -0.01 -0.00 1.00 -0.00 0.07
0.19 0.99 -0.07 -0.00 1.00 0.81
0.18 0.82 -0.06 0.07 0.81 1.00
mag
depth
latitude
longitude
impact_score
log_depth
1.0
0.8
0.6
0.4
0.2
0.0

CLASSIFICATION

Hyperparameter Tuning

Purpose of Tuning:

- Hyperparameters control the behavior of algorithms (e.g., tree depth, number of estimators). Tuning these parameters ensures that the model generalizes well to unseen data.

Methodology:

- We used **GridSearchCV**, a systematic approach to hyperparameter optimization:
  a. Defined a parameter grid specific to each model.
  b. Conducted exhaustive searches over the parameter combinations using cross-validation (5-fold CV).
  c. Evaluated performance based on weighted F1 score, suitable for imbalanced datasets.

Benefits of Using GridSearchCV

1. Automated and Systematic
2. Consistent Evaluation
3. Improved Generalization
4. Optimal Model Performance
5. Scalability

Applied Machine Learning Algorithms

We employed three widely-used tree-based machine learning algorithms for classification:

1. **Decision Tree Classifier:**
   - **Why Selected:**Decision trees are interpretable and fast to train. They serve as a good baseline model to understand feature splits and interactions.
   - **Strengths:** Handles both numerical and categorical data effectively and captures non-linear relationships.
   - **Potential Drawbacks:** Overfitting may occur especially in deep trees.

'MAX_DEPTH': 10

'MIN_SAMPLES_LEAF': 1

'MIN_SAMPLES_SPLIT': 2

Classification Report for DecisionTree:

	precision	recall	f1-score	support
green	0.95	0.87	0.91	1419
orange	0.85	0.74	0.79	1539
red	0.76	0.85	0.81	1457
yellow	0.73	0.80	0.76	1491
accuracy			0.81	5906
macro acg	0.82	0.82	0.82	5906
weighted avg	0.82	0.81	0.82	5906

Applied Machine Learning Algorithms

2.**Random Forest Classifier:**

- **Why Selected:**Random Forests are ensembles of decision trees, reducing the risk of overfitting while maintaining high flexibility.
- **Strengths:** Robust against noise, handles missing data well, and provides feature importance.
- **Potential Drawbacks:** Slower to train compared to a single decision tree, but efficient compared to boosting algorithms.

'MAX_DEPTH': 10

'N_ESTIMATORS': 100

'MIN_SAMPLES_SPLIT': 2

Classification Report for RandomForest:

	precision	recall	f1-score	support
green	0.98	0.91	0.95	1419
orange	0.88	0.87	0.87	1539
red	0.89	0.91	0.90	1457
yellow	0.84	0.89	0.86	1491
accuracy			0.90	5906
macro acg	0.90	0.90	0.90	5906
weighted avg	0.90	0.90	0.90	5906

Applied Machine Learning Algorithms

3.**Gradient Boosting Classifier:**

- **Why Selected:** Gradient Boosting builds an ensemble of weak learners iteratively, focusing on reducing errors of prior models.
- **Strengths:** Excels in predictive accuracy, especially on imbalanced datasets.
- **Potential Drawbacks:** Computationally expensive and requires careful tuning to avoid overfitting.

'LEARNING_RATE': 0.1

'MAX_DEPTH': 5

'MIN_SAMPLES_SPLIT': 2

'N_ESTIMATORS': 100

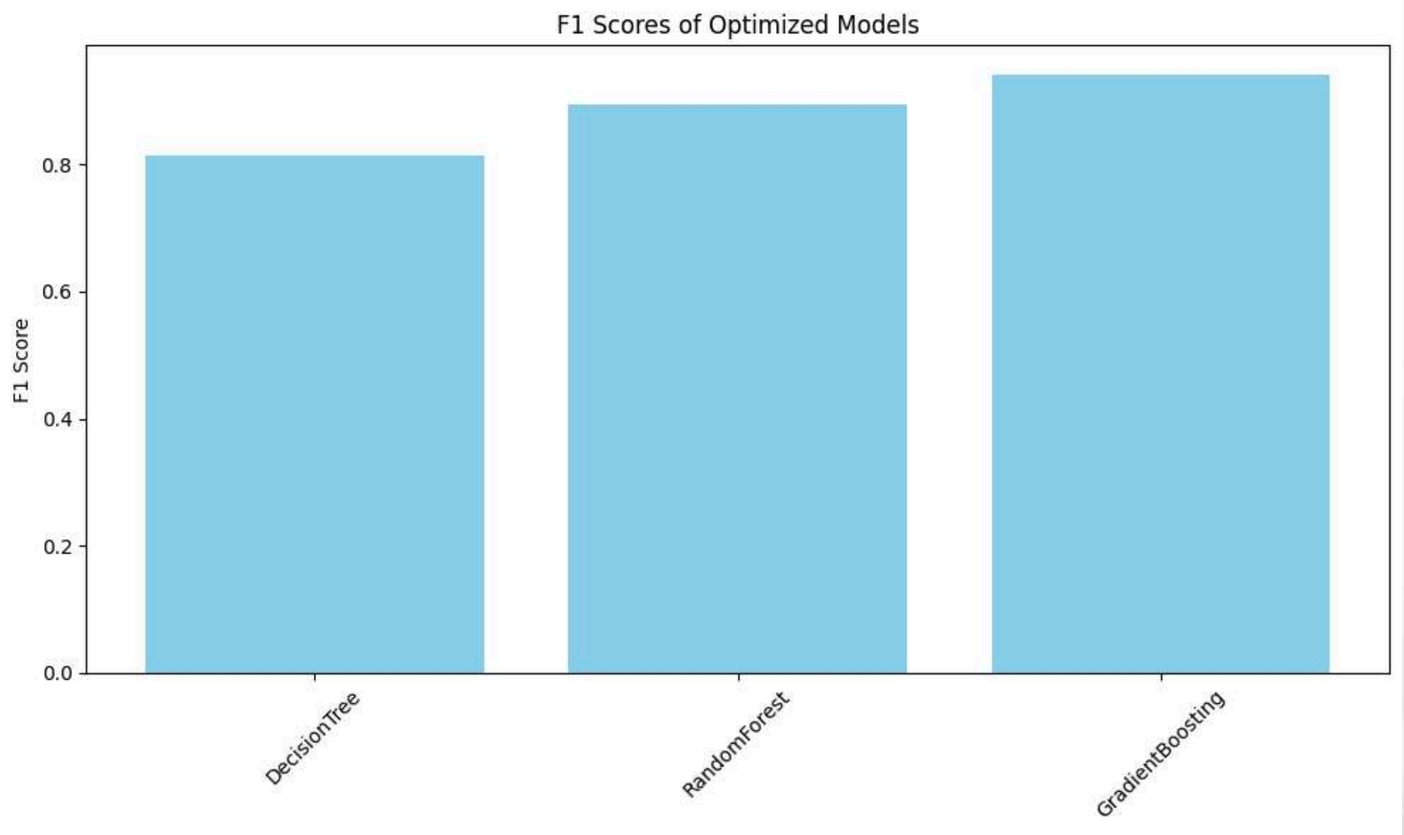
Classification Report for GradientBoosting:

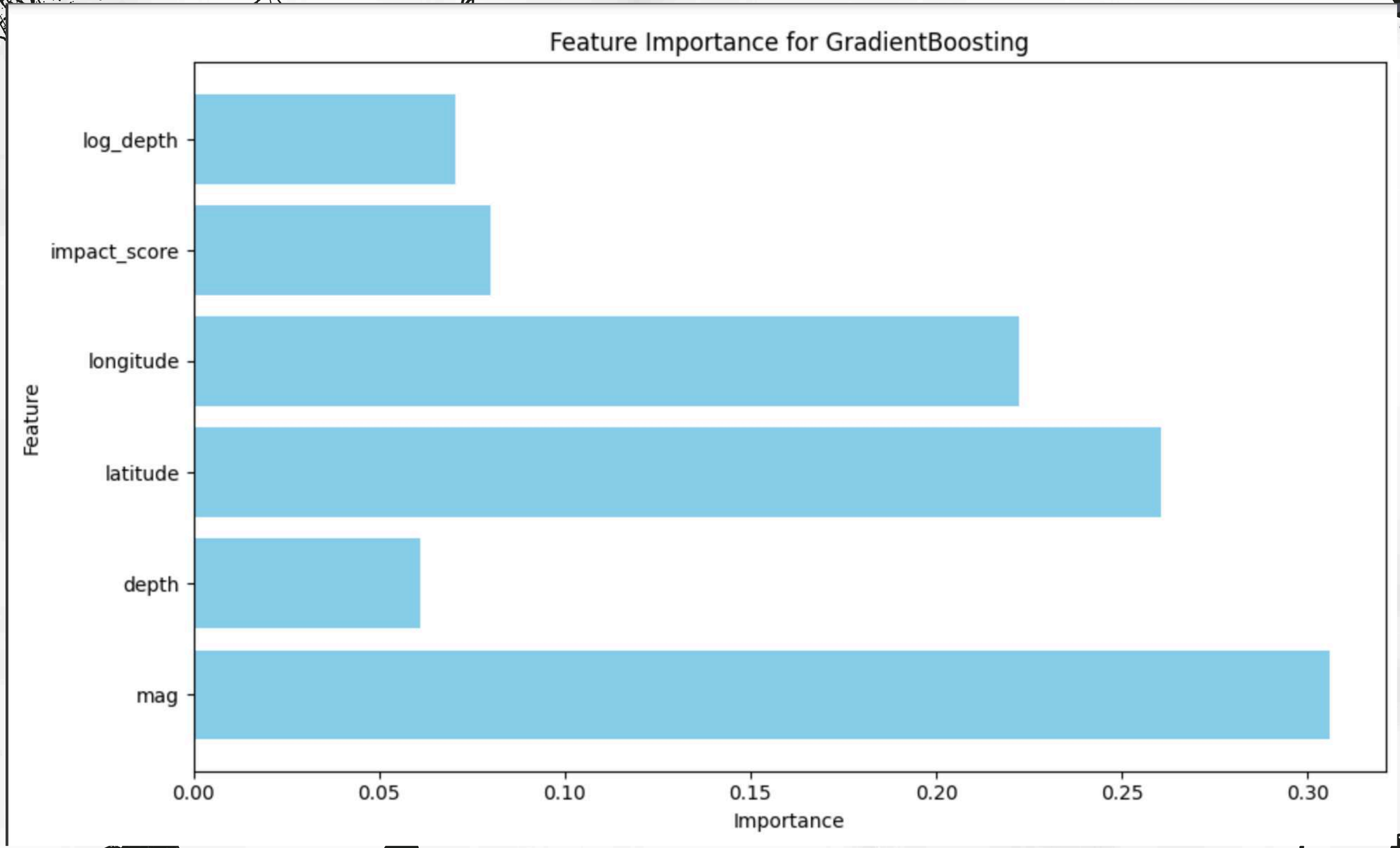
	precision	recall	f1-score	support
green	0.98	0.95	0.96	1419
orange	0.93	0.92	0.93	1539
red	0.93	0.97	0.95	1457
yellow	0.92	0.93	0.92	1491
accuracy			0.94	5906
macro acg	0.94	0.94	0.94	5906
weighted avg	0.94	0.94	0.94	5906

RESULT

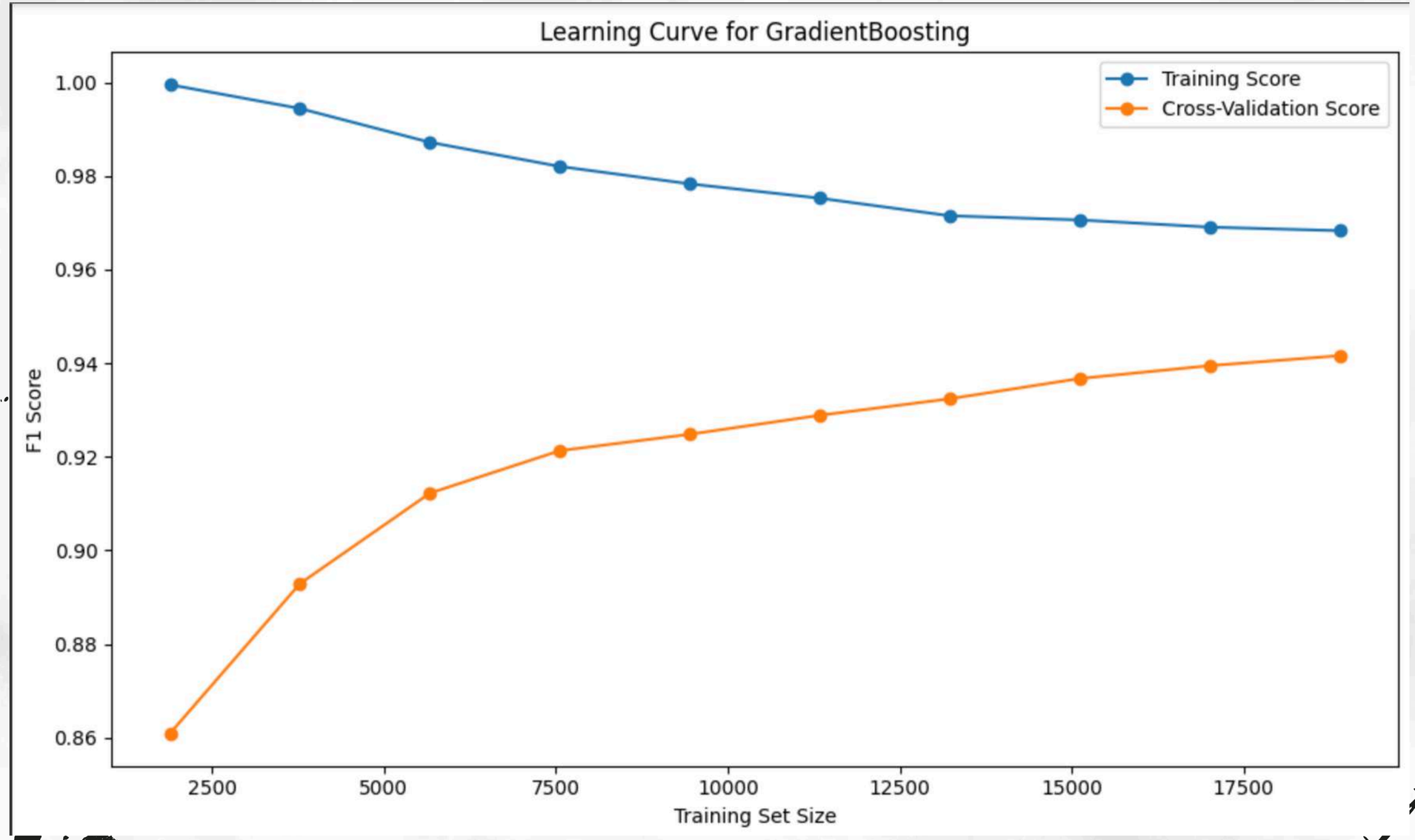
The **Gradient Boosting Classifier** was selected as the best model because:

1. It achieved the highest F1 score and accuracy.
2. It effectively handled class imbalances, improving predictions for classes.
3. Its iterative refinement process captured complex patterns in the data.




Feature Importance for GradientBoosting
log_depth
impact_score
longitude
latitude
depth
mag
Feature
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance

LEARNING CURVES ARE USED TO DIAGNOSE **OVERFITTING** OR **UNDERFITTING** BY VISUALIZING MODEL PERFORMANCE ON THE TRAINING AND VALIDATION SETS AS THE TRAINING DATA SIZE INCREASES.



THANK YOU

06.02.2023