Modeling Earthquake Damage

Machine Learning Project
Milestone 2

Presented by:

Omowonuola Molayosi Akintola

Rabina Twayana

Photo Source CircleOfBlue

Université Bretagne Sud ubs

COPERNICUS MASTER IN DIGITAL EARTH

Co-funded by the European Union

1. Recap

01 **Background**

02 **Target Variable**

1. Background

- Survey conducted after the 2015 Gorkha earthquake in Nepal. Trained enumerators visited the affected areas and collected information on the level of damage, structural characteristics (foundation, roof, building materials), household information, age, and use for over 250,000 buildings.

- The primary goal of the survey was to identify beneficiaries eligible for government assistance for housing reconstruction but the data now serves other purposes including ML tasks.

- The main task is to predict the damage level for buildings affected by the earthquake.

Target Variable

The target variable represents the level of damage to the buildings. There are 3 grades

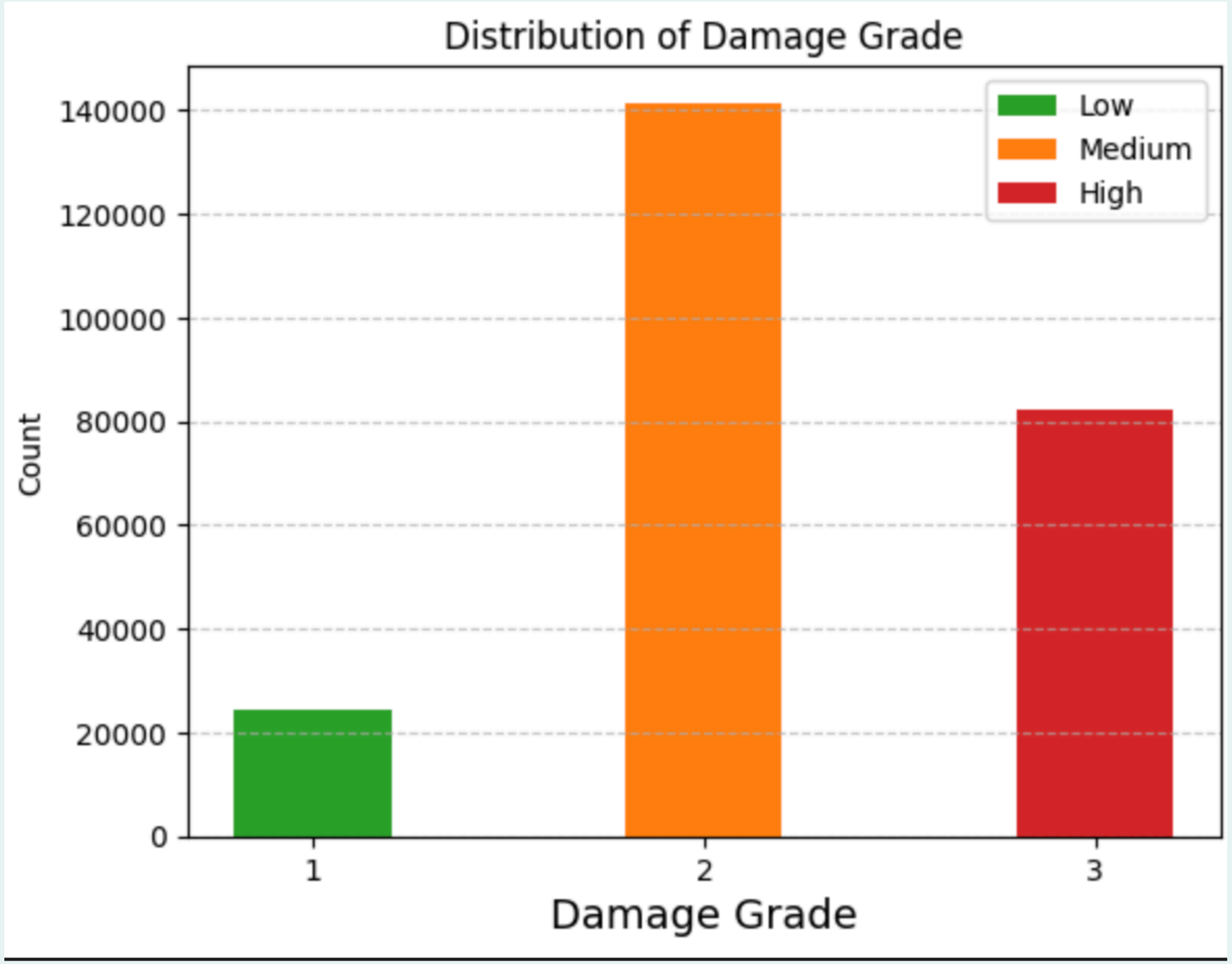
01 Low Damage

02 Medium Damage

03 Almost Complete Destruction

The ML task for this project is considered a multiclassification task, where one of the three damage grades are assigned to the buildings.

Highly Imbalanced Data !!!



2. Milestone 2

01 Modification (Milestone 1)

02 Model Selection

03 Hyperparamater Tuning and Model Training

04 Mode Comparision

05 Insights

1. Modification (Milestone 1)

- Keep the ‘height-percentage’ (high correlated with ‘count floor’), since tree-based models, not affected by the correlated variables.
- Keep only the “has_secondary_use” feature and remove its related sub-categories.
- Apply one-hot-encoding to categorical features directly in the MCA

“has_secondary_use_agriculture",
"has_secondary_use_hotel",
"has_secondary_use_rental",
"has_secondary_use_institution",
"has_secondary_use_school",
"has_secondary_use_industry",
"has_secondary_use_health_post",
"has_secondary_use_gov_office",
"has_secondary_use_use_police",
"has_secondary_use_other"

Total Sum: 29713

The has_secondary_use captures nearly all the information represented by the sub-category since the totals are extremely close.

"has_secondary_use"

Total Sum: 29156

2. Model Selection

1. DummyClassifier(Baseline Model)

- Simple baseline model that ignores input features and makes predictions using simple rules
- Strategy: “stratified”, Target class is imbalanced and has equal importance
- The “stratified” strategy generates random predictions that follow the class distribution as per the training set.

2. RandomForest Classifier

- A supervised learning algorithm,.
- Builds multiple decision trees with the bagging (bootstrap aggregation) method with random selection of features
- Random forest doesn’t overfit since the averaging lowers the overall variance.

3.HistGradientBoostingClassifier

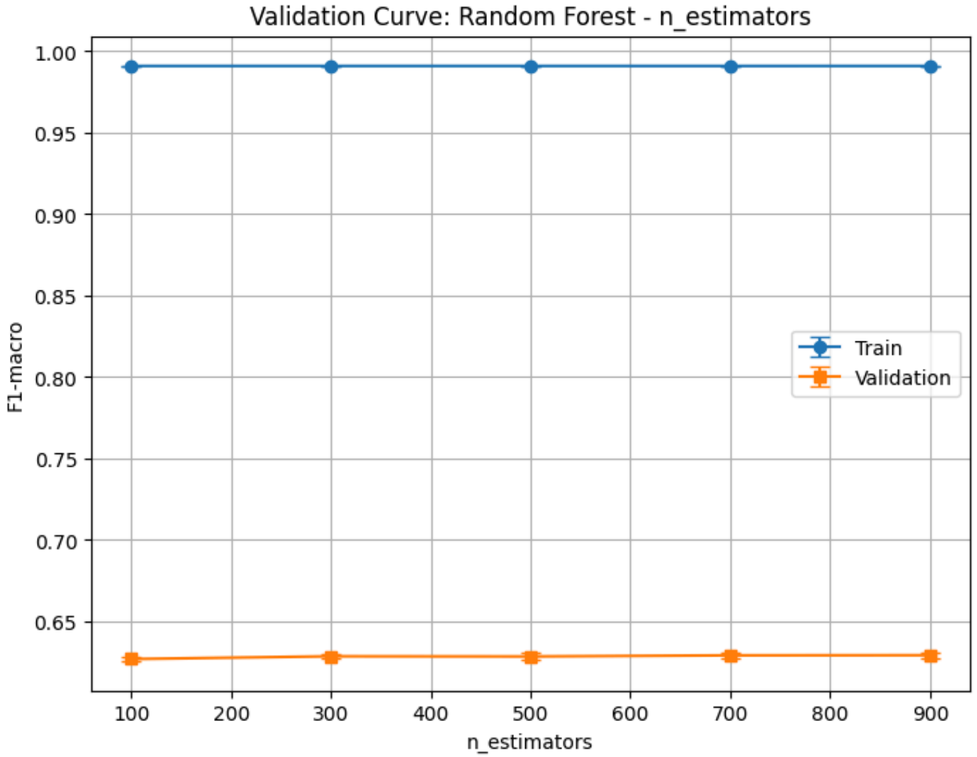
- A supervised learning algorithm.
- Builds decision trees in a boosting framework using histogram binning, which makes it faster and more memory-efficient.

Random Forest

Hyperparameter: n_estimator, max_depth, min_sample_leaf, min_sample_split, class_weight, max_features

Tuning Strategy: Stratified K-Fold Cross validation

- Train model (default parameters)
- Tune only n_estimators
- Tune the remaining hyperparameters
  - GridSearch CV
  - Optuna
- Train the final model using the best hyperparameters and evaluate performance



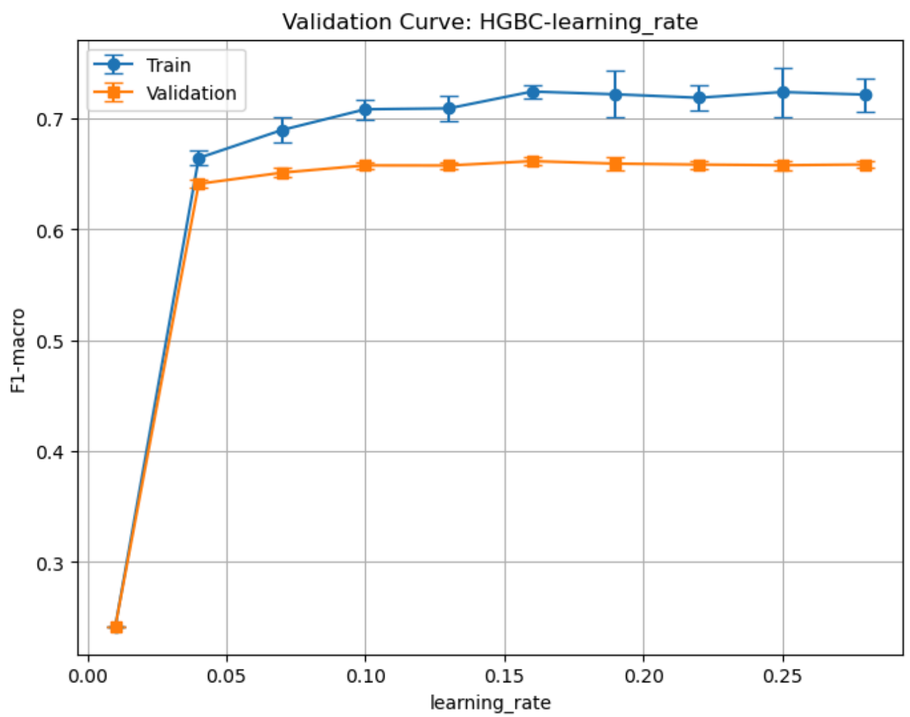
Hyperparameters	Default	Tuned (GridSearch)	Tuned (Optuna)
n_estimator	100	100	100
max_depth	sqrt	20	20
max_features	None	0.8	0.79
min_sample_leaf	2	4	4
min_sample_split	1	5	12
class_weight	None	balanced	balanced

HistGradientBoosting Classifier

Hyperparameters: learning rate, max_depth, min_samples_leaf, max_features, class_weight, max_iter

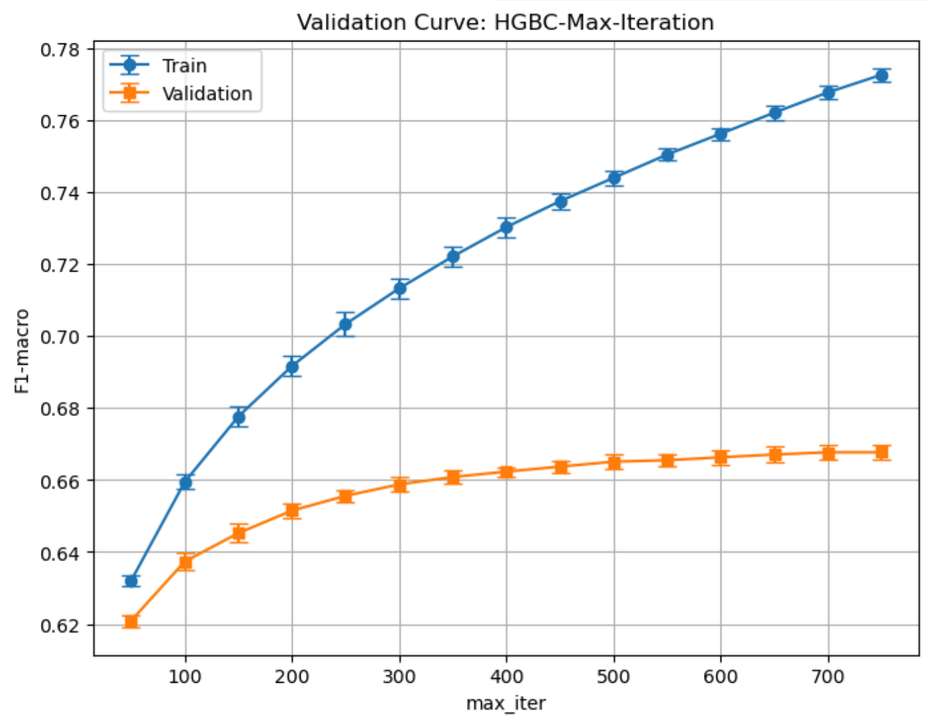
Tuning Strategy: Stratified K-Fold Cross validation

- Train model (Default Parameters)
- Tune learning rate (early_stopping enabled)
- Tune remaining hyperparameters (early_stopping enabled)
  - GridSearchCV
  - Optuna
- Tune max_iter (early_stopping disabled)
- Train the final model using the best hyperparameters and evaluate performance



Best learning_rate= 0.1

Best max_iter = 350



HistGradientBoosting Classifier

Best Hyperparameters

Hyperparameters	Default	Tuned (GridSearchCV	Tuned (Optuna)
learning_rate	0.1	0.1	0.1
max_depth	None	6	8
min_samples_leaf	20	60	157
max_features	1.0	0.7	0.84
class_weight	None	None	None
max_iter	100	350	350

5. Model Comparison

Metric Comparision: Macro F1-score

Dataset	**DummyClassifier (Baseline)**	**Random Forest**			**HistGradientBoostingClassifier**		
		Default	GridSearch CV	Optuna	Default	GridSearch CV	Optuna
Train	0.3318	0.9866	0.7921	0.7783	0.6613	0.7125	0.7164
Test	0.3361	0.6319	**0.6713**	0.6705	0.6414	0.6649	**0.6662**

Model Recommendation: HistGradietBoostingClassifier

Hyperparameter Tuning Computational Time Comparision
(24 GB RAM, 7 core)

GridSearchCV		**Optuna**	
RF	HGBC	RF	HGBC
126 min (324 combinations)	58 min (160 combinations)	14 min (25 trials)	20 min (25 trials)

5. Model Comparison

Feature Importance based on Mean Decrease in Accuracy (MDA)

S.N	Random Forest	HistGradientBoostingClassifier
1	geo_level_1_id	geo_level_1_id
2	geo_level_2_id	geo_level_2_id
3	MCA_0	MCA_0
4	has_superstructure_mud_mortar_stone	geo_level_3_id
5	geo_level_3_id	has_superstructure_mud_mortar_stone

5. Insights

Advantages

- Non-parametric models
- Tree based model work well with imbalanced data
- Handles correlated features well, Robust to outlier
- Generalization improves after hyperparameter tuning

Limitation

- Sensitive to hyperparameters
- GridSearchCV based hyperparameter tuning is time consuming

Next Steps:

- Clarifying categorical feature meaning with data
- Explore feature engineering (eg: Vulnerability indices)
- Test other models (SVM, XGBoost,CatBoost)
- Consider data splitting based on geo_level_id, and access model performance varies across different regions

THANK YOU