



Project Proposal

Earthquake Damage Machine Learning Classification

EPSILON AI
Mohamed Gamal

28 Muhammed Aly Street.
Cairo

01154183666
Ezzat22hegazy@gmail.com



Logo
Name

Objective:

The goal of this project is to develop a machine learning model for classifying earthquake damage grades of buildings based on various features. The target variable, "damage_grade," ranges from 1 to 5, indicating the severity of damage.

Data Overview:

The dataset contains information on various attributes of buildings before and after an earthquake. Here are some key columns:

1. Building Information:

- **building_id**: Unique identifier for each building.
- **district_id**, **vdcmun_id**, **ward_id**: Geographic identifiers.
- **count_floors_pre_eq**, **count_floors_post_eq**: Number of floors before and after the earthquake.
- **age_building**: Age of the building.
- **plinth_area_sq_ft**: Area of the building in square feet.
- **height_ft_pre_eq**, **height_ft_post_eq**: Height of the building before and after the earthquake.

2. Building Structure:

- **land_surface_condition**: Condition of the land surface.
- **foundation_type**: Type of foundation.
- **roof_type**: Type of roof.
- **ground_floor_type**, **other_floor_type**: Floor types.
- **position**: Position of the building.
- **plan_configuration**: Configuration of the building.

3. Superstructure Information:

- Various binary columns indicating the presence of different types of superstructures.

4. Post-Earthquake Information:

- **condition_post_eq**: Condition of the building after the earthquake.
- **damage_grade**: Target variable indicating the severity of damage.

Methods:

1. Data Preprocessing:

- Handle missing values in **position** and **plan_configuration**.
- Address any inconsistencies or outliers.

2. Feature Engineering:

- Create new features if needed based on domain knowledge.
- Encode categorical variables.

3. Model Selection:

- Utilize machine learning algorithms suitable for multi-class classification.

- Consider algorithms such as Random Forest, XGBoost, or neural networks.

4. **Training and Validation:**

- Split the dataset into training and validation sets.
- Train the model on the training set and validate on the validation set.

5. **Evaluation:**

- Evaluate the model's performance using appropriate metrics (e.g., accuracy, F1 score).
- Fine-tune the model for better results.