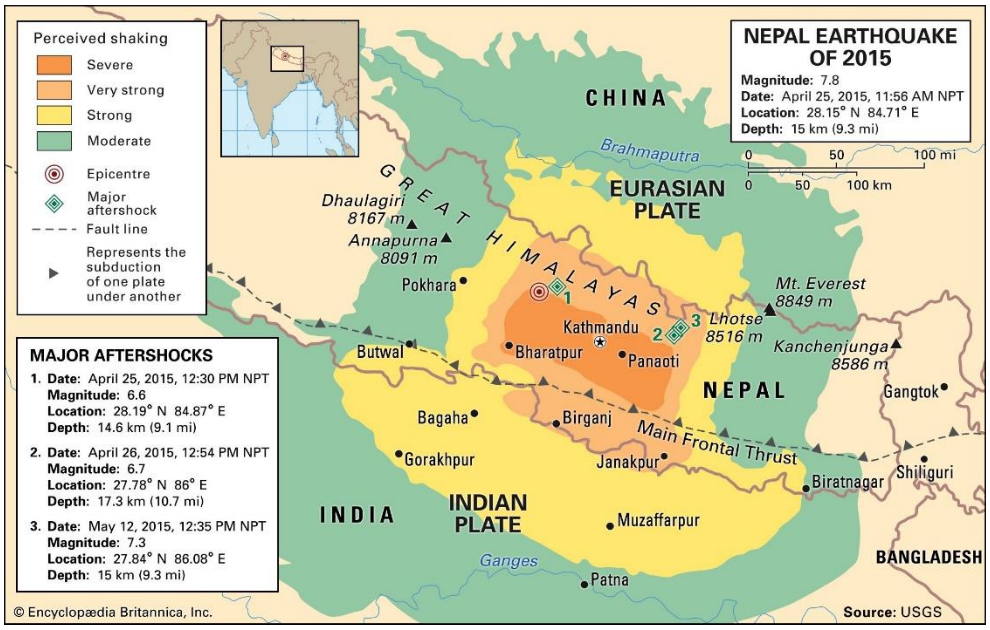
# EARTHQUAKE DAMAGE PREDICTION REPORT

## BUSINESS PROBLEM

Based on aspects of building location and construction, the goal is to predict the level of damage to buildings caused by the 2015 Gorkha earthquake in Nepal. This is a multiclass classification problem.



## THE DATASET

The data was collected through surveys by Kathmandu Living Labs and the Central Bureau of Statistics, which works under the National Planning Commission Secretariat of Nepal. This survey is **one of the largest post-disaster datasets ever collected**, containing valuable information on earthquake impacts, household conditions, and socio-economic-demographic statistics. The task in this project is to predict how badly an individual's house is damaged, given the information about its location, secondary usage, and the materials used during construction.

The dataset contains 2,60,601 observations and has a total of 39 features- 31 numerical and 8 categorical. Categorical variables were obfuscated by random lowercase ascii characters. Another unique characteristic about the dataset is that it has 12 binary features with 0 (no) or 1(yes) as the possible values. These binary features or flag variables were mostly sparse features i.e., most values in the column were zero.

The target variable i.e., damage grade is ordinal in nature and represents the level of damage done to the buildings hit by the earthquake based on 3 grades-

- 1 represents low damage
- 2 represents a medium amount of damage
- 3 represents almost complete destruction

## EDA

Basic Domain Analysis was conducted where each variable was described and a general relationship was established with the target variable. For example, flag variable **'has_superstructure_rc_engineered'** that indicates if the superstructure was made of engineered reinforced concrete is plays an important role in predicting damage levels as building's that use this material are more resistant to earthquake damage. Also, the 'land_surface_condition' type where the building was constructed can be useful in predicting the level of earthquake damage. Areas that are prone to natural disasters often have softer ground material which can move and cave during heavy rain or vibrations

A detailed univariate analysis was conducted and visualized using charts, count plots and histograms using the seaborn library. Some insights from the analysis were that 90% of the observation were below 50 years of age and a few extremely positive values (age = 995) were skewing the distribution. The analysis also revealed that the target variable was imbalanced with 57% of the observations have a damage grade=2, 33% have a damage grade=3 and 10% have damage grade= 1.

Bivariate analysis was also performed using violin plots and histograms that revealed interesting insights in the relationship of each variable with the target variable. For instance, buildings with a plan configuration of type 'd' have the most damage with level 2 being the most frequently occurring damage grade in this type. Buildings that used mud mortar stone for construction showed proportionally higher damage grades (grade 2 or 3) as compared to those that did not use this material. Also, damage grades (1, 2 and 3) are high for buildings with 2 floors. Buildings with 1 floor have lower damages (grade=1) as compared to buildings with 3 floors that show more observations in damage grade = 3.

## DATA PREROCESSING AND FEATURE ENGINEERING

### MISSING VALUES

The dataset has no missing values.

### OUTLIERS

Since tree-based models were used for prediction, handling outliers were not necessary. Such models are robust to outliers because every split slices the dataset and is condition based. Once the dataset is split, we only check the sample proportion in each split region and how much impurity exists thus not considering the absolute value of any observation for any feature. Consequently, the outliers would end up in a particular branch of the tree and wouldn't impact model performance.

### HANDLING CATEGORICAL VARIABLES

Majorly, One Hot Encoding was used to transform the categorical features into a numeric form. It was also noted that despite the 3 geo level id features being in a numeric form, they were actually categorical variables and had to be treated accordingly. The different values in these features were just numeric labels of locations with different specificity. The cardinality of features increases from geo level id 1 to 3 as the specificity of the building's location increases. For such high cardinality features target encoding was used which takes the mean of the response variable of each group and assigns all observations in that group that value.

## FEATURE ENGINEERING

Certain numerical features had a positively skewed distribution namely ‘age’, ‘height percentage’ and ‘area percentage’. This means that there were some observations with very high values as compared to the rest of the data in these features. This is problematic since such skewed data hurts the performance of nearly every prediction model, since it interferes with the loss function of the model.

In order to tackle this excessive rightward skewness problem, the log of the values of these features was taken so that the major disparity in observation values could be reduced without changing the order or meaning of the data. Since the age feature had 0 values and the natural log of 0 does not exist, 1 was added to each value and then the log was taken.

## FEATURE SELECTION

A heatmap of correlation values of each feature and the target variable was constructed which revealed that there were no highly correlated features. The ‘building_id’ feature was dropped since it was a unique value column that did not provide any useful information in prediction of the target variable.

## FEATURE SCALING

Tree based models don’t require feature scaling hence this was not performed.

## MODEL SELECTION

After pre-processing, the data was fit to 2 models- Random Forest Classifier and XGBOOST. Distance based algorithms like KNN and SVM was not selected since such algorithms don’t perform well when the number of features is high or are sparse as the dimensional complexity increases. Moreover, KNN would be more time consuming and computationally intensive on this dataset as the number of observations were more than 2,00,000.

For the models, the data was split into training and testing set using sklearn’s train test split method.

## EVALUATION

The following evaluation metrics were calculated for each model-

1. Recall- it is a measure of from the total number of positive results how many positives were correctly predicted by the model. It is literally is how many of the true positives were recalled (found). Recall= $TP/ (TP+FN)$
2. Precision- it is a measure of amongst all the positive predictions, how many of them were actually positive. Precision=$TP/ (TP+FP)$
3. F1 score- is defined as the harmonic mean of Precision and Recall. It sums up the predictive performance of a model by combining two otherwise competing metrics — precision and recall.
4. Accuracy- it’s defined as the total number of correct classifications divided by the total number of classifications. It tells us the fraction of predictions our model got right.

F1 score was used as the primary evaluation metric since both precision and recall were important in this case. Since this was a multiclass problem, either a micro or macro averaged f1 score had to be taken.

A macro-average will compute the metric independently for each class and then take the average hence treating all classes equally, whereas a micro-average will aggregate the contributions of all classes to compute the average metric. The main difference between macro and micro averaging is that macro **weighs each class equally** whereas **micro weighs each sample equally**. Thus, macro averaging is the preferred metric when the data is perfectly balanced whereas micro-average is preferable if the classes are imbalanced. Since univariate analysis revealed that our target variable was imbalanced, micro averaged f1 score was selected as the evaluation metric.

https://androidkt.com/micro-macro-averages-for-imbalance-multiclass-classification/#:~:text=A%20macro%2Daverage%20will%20compute,to%20compute%20the%20average%20metric

**CROSS VALIDATION**

5- fold stratified cross validation was performed to ensure that the percentage of samples for each class was preserved in every fold. This technique is preferred when we have an imbalanced dataset to avoid disproportionate representation of target variable classes in the training and testing sets.

A **Feature Importance** graph was also plotted which revealed that the geo level id features have the highest feature importance which is understandable since the location of the building (whether close to the earthquake's epicentre or far) would play a crucial role in predicting the level of damage. The feature importance also shows that nearly all binary variables had a low feature importance score, meaning they were providing the model with little to no predictive information. In order to reduce the number of dimensions and improve model performance the binary features that had a score of less than 0.01% were dropped from the dataset. However, dropping these features had no impact on the model's performance.

Hyperparameter tuning using Randomized Search CV was performed on the Random Forest model and the following best estimator was selected: -

RandomForestClassifier (max_depth=20, min_samples_leaf=3, min_samples_split=3, n_estimators=300)

The model evaluation scores were as follows-

| **Model** | **Random Forest Classifier** | **Random Forest after Hyperparameter Tuning** | **XGBOOST** |
|---|---|---|---|
| **F1 score (micro avg)** | 0.7365 | 0.7511 | 0.7463 |
| **Accuracy** | 0.7365 | 0.7511 | 0.7463 |
| **Avg cross validation score** | 0.7389 | 0.7538 | 0.7473 |
| **Std of cross validation score** | 0.001 | 0.001 | 0.001 |

From the above table we can see that both Random Forest and XGBOOST have very similar scores with both the model's performing well on the dataset.

**Random Forest Classifier after hyperparameter tuning was selected as the final model with a f1 score of 0.7511 and an average cross validation score of 0.7538**.