**Richter's Predictor: Modelling Earthquake Damage**

How artificial intelligence and predictive analysis can help in faster damage recovery from earthquake

**Overview**

**General Overview of the data**

**Data Source:** “train\_values.csv” , “train\_labels.csv”

Inhouse data was collected through surveys by the Central Bureau of Statistics that work under the National Planning Commission Secretariat of Nepal. It is rumoured that 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.

This is a classification problem for which we will be using classical ML techniques to predict from the classes for the given test dataset.

## \*\*Steps to follow:\*\*

- Import the necessary libraries

- Download the dataset

- get all the datasets to the data frame

\_\_\_

- Get the basic info about the columns

- Get the statistical description about the variables

- Get the correlation matrix to view the relationship between the explanatory variables and explained variable and among explanatory variables themselves.

\_\_\_

- Data pre-processing

1. get rid of missing values

2. encode the categorical variables

3. remove useless variables

\_\_\_

- Select the evaluation score - as needed by the competition problem statement

- Split data into training and validation set

- Run different classification models to see which could work best

\_\_\_

- Get the best model and train on the whole training set

- Get the predictions from the test set and replace the values in the submission file

- Make the first submission and view the scores

\_\_\_

- Tune hyperparameters on the best models to further improve the accuracies

- Do the feature selection and feature engineering

- train the best model with best hyperparameters on the whole training set

## Machine learning Theory

Machine Learning model is system that has been trained from features to recognize the pattern and give out a label as an output. In the training set the model tend to learn a general theme around the data and based on the kind of model chosen, aligns the weights to several features in a way to predict the target variable.

Based on aspects of building location and construction, our goal is to predict the level of damage to buildings caused by the 2015 Gorkha earthquake in Nepal. The data mainly consists of information on the buildings' structure and their legal ownership. Each row in the dataset represents a specific building in the region that was hit by Gorkha earthquake.

**Problem description**

Predict the ordinal variable damage\_grade, which represents a level of damage to the building that was hit by the earthquake. There are 3 grades of the damage:

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

**Features**

The dataset mainly consists of information on the buildings' structure and their legal ownership. Each row in the dataset represents a specific building in the region that was hit by Gorkha earthquake.

There are 39 columns in this dataset, where the building\_id column is a unique and random identifier. The remaining 38 features are described in the section below. Categorical variables have been obfuscated random lowercase ascii characters. The appearance of the same character in distinct columns does not imply the same original value.

**Description**

**geo\_level\_1\_id, geo\_level\_2\_id, geo\_level\_3\_id (type: int):**

geographic region in which building exists, from largest (level 1) to most specific sub-region (level 3). Possible values: level 1: 0-30

level 2: 0-1427

level 3: 0-12567

**count\_floors\_pre\_eq (type: int):**

number of floors in the building before the earthquake.

**age (type: int):**

age of the building in years.

**area\_percentage (type: int):**

normalized area of the building footprint.

**height\_percentage (type: int):**

normalized height of the building footprint.

**land\_surface\_condition (type: categorical):**

surface condition of the land where the building was built. Possible values: n, o, t.

**foundation\_type (type: categorical):**

type of foundation used while building. Possible values: h, i, r, u, w.

**roof\_type (type: categorical):**

type of roof used while building. Possible values: n, q, x.

**ground\_floor\_type (type: categorical):**

type of the ground floor. Possible values: f, m, v, x, z.

**other\_floor\_type (type: categorical):**

type of constructions used in higher than the ground floors (except of roof). Possible values: j, q, s, x.

**position (type: categorical):**

position of the building. Possible values: j, o, s, t.

**plan\_configuration (type: categorical):**

building plan configuration. Possible values: a, c, d, f, m, n, o, q, s, u.

**has\_superstructure\_adobe\_mud (type: binary):**

flag variable that indicates if the superstructure was made of Adobe/Mud.

**has\_superstructure\_mud\_mortar\_stone (type: binary):**

flag variable that indicates if the superstructure was made of Mud Mortar - Stone.

**has\_superstructure\_stone\_flag (type: binary):**

flag variable that indicates if the superstructure was made of Stone.

**has\_superstructure\_cement\_mortar\_stone (type: binary):**

flag variable that indicates if the superstructure was made of Cement Mortar - Stone.

**has\_superstructure\_mud\_mortar\_brick (type: binary):**

flag variable that indicates if the superstructure was made of Mud Mortar - Brick.

**has\_superstructure\_cement\_mortar\_brick (type: binary):**

flag variable that indicates if the superstructure was made of Cement Mortar - Brick.

**has\_superstructure\_timber (type: binary):**

flag variable that indicates if the superstructure was made of Timber.

**has\_superstructure\_bamboo (type: binary):**

flag variable that indicates if the superstructure was made of Bamboo.

**has\_superstructure\_rc\_non\_engineered (type: binary):**

flag variable that indicates if the superstructure was made of non-engineered reinforced concrete.

**has\_superstructure\_rc\_engineered (type: binary):**

flag variable that indicates if the superstructure was made of engineered reinforced concrete.

**has\_superstructure\_other (type: binary):**

flag variable that indicates if the superstructure was made of any other material.

**legal\_ownership\_status (type: categorical):**

legal ownership status of the land where building was built. Possible values: a, r, v, w.

**count\_families (type: int):**

number of families that live in the building.

**has\_secondary\_use (type: binary):**

flag variable that indicates if the building was used for any secondary purpose.

**has\_secondary\_use\_agriculture (type: binary):**

flag variable that indicates if the building was used for agricultural purposes.

**has\_secondary\_use\_hotel (type: binary):**

flag variable that indicates if the building was used as a hotel.

**has\_secondary\_use\_rental (type: binary):**

flag variable that indicates if the building was used for rental purposes.

**has\_secondary\_use\_institution (type: binary):**

flag variable that indicates if the building was used as a location of any institution.

**has\_secondary\_use\_school (type: binary):**

flag variable that indicates if the building was used as a school.

**has\_secondary\_use\_industry (type: binary):**

flag variable that indicates if the building was used for industrial purposes.

**has\_secondary\_use\_health\_post (type: binary):**

flag variable that indicates if the building was used as a health post.

**has\_secondary\_use\_gov\_office (type: binary):**

flag variable that indicates if the building was used fas a government office.

**has\_secondary\_use\_police (type: binary):**

flag variable that indicates if the building was used as a police station. has\_secondary\_use\_other (type: binary): flag variable that indicates if the building was secondarily used for other purposes.

**Our Target**

We are predicting the level of damage from 1 to 3(Low,Medium,High). The level of damage is an ordinal variable meaning that ordering is important. This can be vied as a classification or Regression Problem

**performance metrics**

To measure the performance of our algorithms, we have used the F1 score which balances the precision and recall of a classifier

F1 - performance on a binary classifier

But since we have three possible labels we used a variant called the micro averaged F1 score.

In Python, we can easily calculate this loss using sklearn.metrics.f1\_score with the keyword argument average='micro'

**Modelling**

Model Micro avg./f1 score - 0.56 , Logistic Regression - 0.59, KNC - 0.65 , Decision Tree - 0.66, Random Forest - 0.72, Xgb – 0.112

**Feature Importance**

No Feature index Importances

1 geo\_level\_3\_id 26.67 2 geo\_level\_2\_id 20.12 3 Age 8.8 4 geo\_level\_2\_id 8.6 5 ground\_floor\_type\_v 5.20 6 roof\_type\_x 3.8 7 count\_floors\_pre\_eq 3.23 8 has\_super\_structure\_mud\_mortar\_stone 3.21 9 foundation\_type\_i 3.17 10 height\_percentage 2.71

**Conclusion**

This modelling proves that seismic damage prediction using Machine Learning models is possible. Nevertheless, limitations concerning the prediction accuracy are present.