Problem description

We're trying to predict the [ordinal variable](https://www.ma.utexas.edu/users/mks/statmistakes/ordinal.html) 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

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.

Feature 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\_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.

# Approach to the problem:

The dataset contains 4 numerical variables and rest all are categorical variables.

Since the dataset is massive and we have many features, we may have to select and analyse which features are important to our target variables.

Since, we have to predict the damage\_grade, the scene is pretty clear that we have to apply classification analysis.

EDA and Data Processing:

1. The dataset has no missing values.
2. The dataset has high number of categorical / flag variables.
3. We build a countplot to check the distribution of target variable, to see whtehr is imbalanced is not. It is not clear but later in the modelling we will take to avoid it, if there is a presence.

Feature selection:

This data requires feature selection as we have a large number of features and data points. Hence, we will do statistical tests for feature selection.

We do chi-square tests for comparing categorical data with categorical data or in other words to see if our categorical features have any impact on the severity of the damage/ damage\_grade.

We do kruskal tests to check relationship between numerical data and the categorical damage\_grade to check if the numerical effects the categorical data.

We then remove the unwanted or irrelevant features to remove unwanted noise in the dataset.

Splitting and modelling the data:

Since the categorical variables have low number of categories and that too they have ordinal data, we can go ahead with label encoding as the chances of bias will be low since the number of categories in each feature are less.

We can go by dummification, but that will be a nightmare because geo\_level\_id 1,2,3 are all ordinal data. It will create additional categories.

Getting the First result after base modelling:

We apply random forest and gradient boosting since they are good at tacking data imbalance.

Since, the model is going to be evaluated on F1 score, then accuracy score is not a credible metric for evaluation.

For a good F1 score, we need to be very careful of the data imbalance and true positives and false negatives.

So, for this we check the confusion matrix, and see that the data is imbalanced.

So, we apply smote to balance the data and get a better F1 score.

Conclusion:

I got a score of 0.70 on this dataset, which was ranked 406 out of 1670 participants. But there is a lot of room for improvement.