An algorithm for predicting damage done to buildings due to 2015 Gorkha earthquake in Nepal

Research Question

The purpose of this study was to identify the level of damage done to the buildings due to 2015 Gorkha earthquake in Nepal from multiple related factors like age, area, height of the building, structure of the building, etc.

The damage level is divided into 3 classes:

1: low damage

2: moderate damage

3: high amount of damage

We have to classify various buildings into these 3 classes based on the related factors.

Methods

Sample:

The sample included N=260601 values and 39 features. 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.

Measures:

The damage level due to earthquake was measured for each building based on various predictors.

The predictors included:

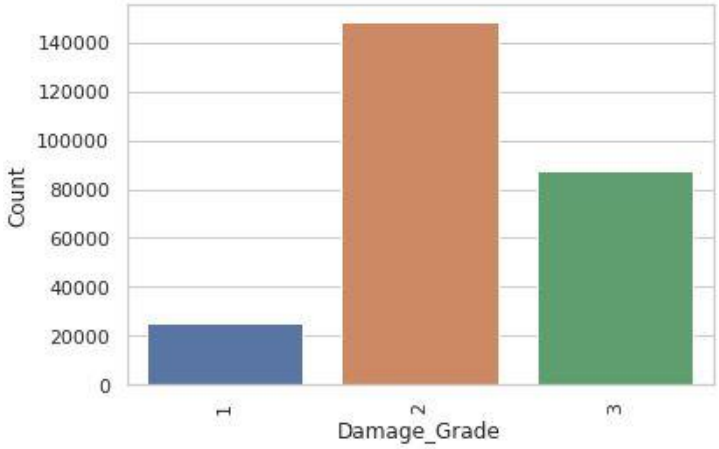
- Age
- Area percentage
- Height percentage
- Foundation type
- Roof type
- Superstructure type
- Number of floors
- Number of families
- Floor types

Analysis:

The various variables were analysed individually as well as in relation with the target variable.

The frequency distributions of categorical variables were examined.

The target variable was analysed visually as shown below.



Univariate Analysis

Various explanatory variables are examined visually, as shown below.

Bar chart, distance plots, etc were used to display various features.

Bivariate Analysis

Chi-square test was used to analyse the relationship between the explanatory variables and the target variable.

Then, feature engineering was used to keep only the related variables into the data set and remove all the un-related variables.

The categorical variables are encoded into binary variables to be able to better predict the damage level.

Classification Algorithm

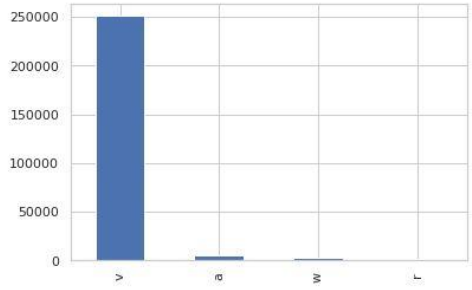
Classification algorithm Decision Tree Classification and Random Forests is used to predict the level of damage caused to buildings due to earthquake.

Results

Univariate Analysis:

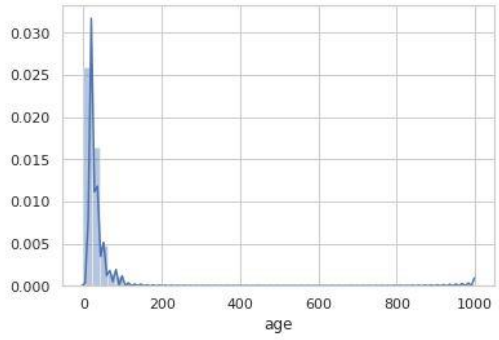
Univariate Analysis

```
In [376]: res_train['legal_ownership_status'].value_counts().plot.bar()

Out[376]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2a78fadf60>
```



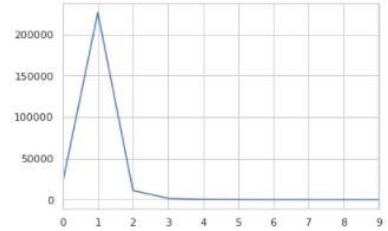
The variable ‘legal ownership status’ is visually examined, the above figure shows that most of the buildings have legal ownership status of type ‘v’.

```
In :  #Age of Building
      sns.distplot(res_train['age'])

In [377]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2a78f9cfd0>
```



The above figure shows the visual description of ‘age’ of the buildings.

```
In :  print(res_train['count_families'].value_counts())
      res_train['count_families'].value_counts().sort_index().plot.line()

1    226115
0     20862
2     11294
3      1802
4       389
5       104
6        22
7         7
9         4
8         2
Name: count_families, dtype: int64

Out[379]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2a78e78ef0>
```



From the above figure, we can see that most of the buildings have only 1 family living in them.

Bivariate Analysis:

Chi-square test is used to find the association between explanatory variables and target variable.

```
ChiSquareTest(cat,res_train)
```

```
land_surface_condition
dof=4
probability=0.950, critical=9.488, stat=449.671
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)

foundation_type
dof=8
probability=0.950, critical=15.507, stat=48547.161
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)

roof_type
dof=4
probability=0.950, critical=9.488, stat=30251.419
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)

ground_floor_type
dof=8
probability=0.950, critical=15.507, stat=36430.849
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)
```

The above figure shows the chi-square test of various categorical variables in relation with the target variable.

For each variable, significance level is calculated and based on the p-value, it is decided on which variables the damage level depends.

```
ChiSquareTest(cat_binary,res_train)
```

```
has_superstructure_adobe_mud
dof=2
probability=0.950, critical=5.991, stat=1470.498
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)

has_superstructure_mud_mortar_stone
dof=2
probability=0.950, critical=5.991, stat=29276.035
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)

has_superstructure_stone_flag
dof=2
probability=0.950, critical=5.991, stat=1147.811
Dependent (reject H0)
```

```
has_secondary_use_use_police
dof=2
probability=0.950, critical=5.991, stat=1.586
Independent (fail to reject H0)
significance=0.050, p=0.452
Independent (fail to reject H0)

has_secondary_use_other
dof=2
probability=0.950, critical=5.991, stat=72.082
Dependent (reject H0)
significance=0.050, p=0.000
Dependent (reject H0)
```

The above output is of the chi-square test of binary variables to find their association with the target variable. It is shown that the target variable is independent of the variable 'has_secondary_use_police'. So, we can remove this variable from our train and test data set.

Feature Engineering:

Next, we transformed the categorical explanatory variables so that they can be fit into the classification model.

The approach was to examine the frequency of various values in a particular variable and keep encode the most frequently occurring values.

Feature Engineering

```
]: def calculateDistribution(cat, res_train):
     for c in cat:
       print(c)
       print((res_train[c].value_counts())/ len(res_train[c]))
       print(" ")
```

```
]: calculateDistribution(cat,res_train)
```

```
land_surface_condition
t    0.831758
n    0.136331
o    0.031911
Name: land_surface_condition, dtype: float64

foundation_type
r    0.841117
w    0.058012
u    0.054720
i    0.040595
h    0.005556
Name: foundation_type, dtype: float64

roof_type
```

e_other	IsV	IsT	IsR	IsN	IsF	IsQ	IsS	IsD
	1	1	1	1	1	1	0	1
	1	0	1	1	0	1	1	1
	1	1	1	1	1	0	0	1
	1	1	1	1	1	0	1	1
	1	1	1	1	1	0	1	1

The values are encoded into binary labels.

Then we encode our train and test datasets using one hot encoding.

```
: res_train_one_hot=pd.get_dummies(res_train_copy)
  res_test_one_hot=pd.get_dummies(res_test_copy)
```

```
: res_train_one_hot.head()
```

17]:

	count_floors_pre_eq	age	area_percentage	height_percentage	has_superstructure_adobe_mud	has_sup
0	2	30	6	5	1	
1	2	10	8	7	0	
2	2	10	5	5	0	
3	2	10	6	5	0	
4	3	30	8	9	1	

5 rows × 34 columns

Classification Algorithm:

The Decision Tree classifier is used for modelling and training data set is fit into the model.

Then, the damage level is predicted using the test data set.

Random Forest is then used to predict the damage level at a better level.

The accuracy score of random forest classifier for the training set is 71.82 % as shown below.

```
|: rf.score(res_train_one_hot, y_train)*100
```

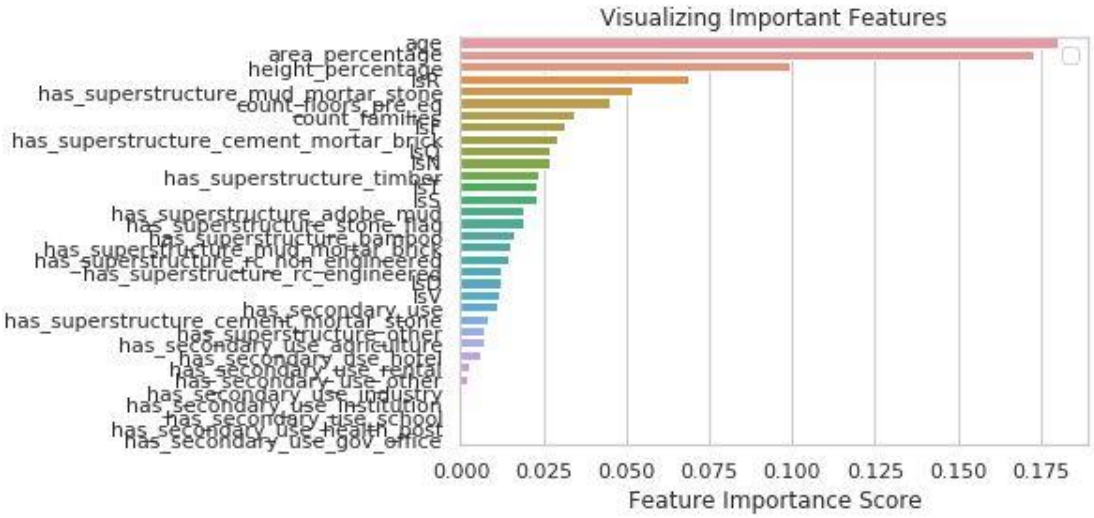
26]: 71.81745273425658

```
|: feature_imp = pd.Series(rf.feature_importances_,index=res_train_one_hot.columns).sort_values(ascending=False)
  feature_imp.head()
```

27]:
```
age                                 0.180340
area_percentage                     0.172775
height_percentage                   0.099048
IsR                                 0.068500
has_superstructure_mud_mortar_stone 0.051692
dtype: float64
```

The feature importance is also examined for the explanatory variables with respect to the random forest classification. The most important features as per the output are:

- Age
- Area
- Height
- Whether the foundation type of building is ‘r’ or not
- Whether the superstructure is made of mud mortar stone or not



The model prediction for the test data set is shown below

433]:

	building_id	damage_grade
0	300051	3
1	99355	2
2	890251	2
3	745817	1
4	421793	3

Conclusions

This project used Random Forest Classifier to predict the damage done to various buildings to the 2015 Gorkha earthquake in Nepal. It also examined the most important features that decide the level of damage caused to buildings.

5 most important features of the train data set are responsible for predicting the damage level of buildings in the Random Forest Classifier.

The accuracy for training data set in the model is around 71.82 %. This suggests that the developed model is quite good for the given sample.

In this particular study, I could not identify the accuracy score of the test data set as I did not have the original values of the predicted target variable.

This model may not be the best model. The future studies may look into other important factors and gather the original values of the predicted variable to test the accuracy of the model.