Predict the Damage to a Building

Asmita Goswami

17 October 2018

1. Definition

1.1. Project Overview

Earthquake prediction is concerned with the specification of the time, location, and magnitude of future earthquakes within stated limits, and particularly the determination of parameters for the *next* strong earthquake to occur in a region.

The Hackerearth provide this problem with the main objective behind is to present a methodology for estimating the benefits of adding earthquake resistance to buildings. Determining the degree of damage that is done to buildings post an earthquake can help identify safe and unsafe buildings, thus avoiding death and injuries resulting from aftershocks. Leveraging the power of machine learning is one viable option that can potentially prevent massive loss of lives while simultaneously making rescue efforts easy and efficient.

1.2. Datasets

In this dataset consist of the before and after details of nearly one million buildings after an earthquake provided by the hackerearth. This dataset is free to download.
The variables present in the dataset are as follows :-

Variable	**Description**
area_assesed	Indicates the nature of the damage assessment in terms of the areas of the building that were assessed
building_id	A unique ID that identifies every individual building
damage_grade	Damage grade assigned to the building after

	assessment (Target Variable)
district_id	District where the building is located
has_geotechnical_risk	Indicates if building has geotechnical risks
has_geotechnical_risk_fault_crack	Indicates if building has geotechnical risks related to fault cracking
has_geotechnical_risk_flood	Indicates if building has geotechnical risks related to flood
has_geotechnical_risk_land_settlement	Indicates if building has geotechnical risks related to land settlement
has_geotechnical_risk_landslide	Indicates if building has geotechnical risks related to landslide
has_geotechnical_risk_liquefaction	Indicates if building has geotechnical risks related to liquefaction
has_geotechnical_risk_other	Indicates if building has any other geotechnical risks
has_geotechnical_risk_rock_fall	Indicates if building has geotechnical risks related to rock fall
has_repair_started	Indicates if the repair work had started
vdcmun_id	Municipality where the building is located

Table 1 : Dataset description

1.3. Problem Statement

This is a classification problem. Inputs is the building details and the output is the prediction of the extent of damage that has been done to a building after an earthquake based on the variables assessment. The damage to a building is categorized in five grades. Each grade depicts the extent of damage done to a building post an earthquake.

1.4. Matrices

The evaluation matrices used for this hackerearth competition is based on F1 Score with 'weighted' average.

In statistical analysis of binary classification, the **F_1 score** (also **F-score** or **F-measure**) is a measure of a test's accuracy. It considers both the precision *p* and the recall *r* of the test to compute the score: *p* is the number of correct positive results divided by the number of all positive results returned by the classifier, and *r* is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F_1 score is the harmonic average of the precision and recall, where an F_1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. The formula for the F1 score is:

$$F1 = 2 * (precision * recall) / (precision + recall)$$

In the multi-class and multi-label case, this is the average of the F1 score of each class with weighting depending on the **average** parameter.

2. Analysis

2.1. Data Exploration

The prediction of damaged grade is done for the provided building. First some visualization of the data was performed. In this, dataset consist of the before and after details of nearly one million buildings after an earthquake provided by the hackerearth. I have randomly classified data into all the five grades. There is a little preprocessing of the data so before even start training models, I first took a glimpse of the data and saw what the shape is and how they are formatted.

Then I extracted the information for training such as :

area_assesed	631761 non-null object
building_id	631761 non-null object
damage_grade	631761 non-null object
district_id	631761 non-null int64
has_geotechnical_risk	631761 non-null float64
has_geotechnical_risk_fault_crack	631761 non-null int64
has_geotechnical_risk_flood	631761 non-null int64

Has_geotechnical_risk_land_settlement	631761 non-null int64
has_geotechnical_risk_landslide	631761 non-null int64
has_geotechnical_risk_liquefaction	631761 non-null int64
has_geotechnical_risk_other	631761 non-null int64
has_geotechnical_risk_rock_fall	631761 non-null int64
has_repair_started	598344 non-null float64
vdcmun_id	631761 non-null int64

2.2. Exploratory Visualization

After completing the data pre-processing, then I performed some graph visualization for better understanding of the data distribution.

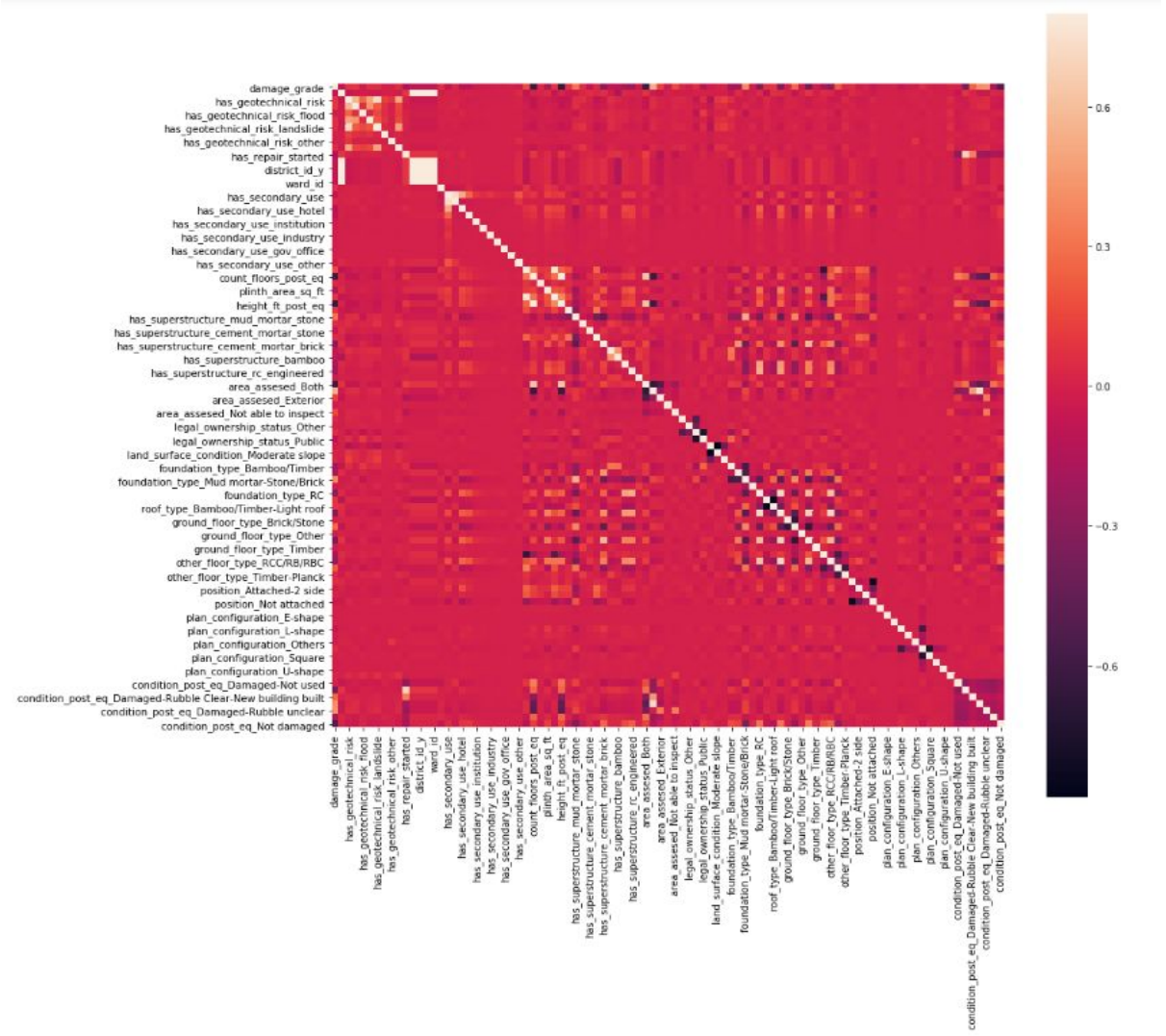

Figure 1 : Heatmap of Training Data

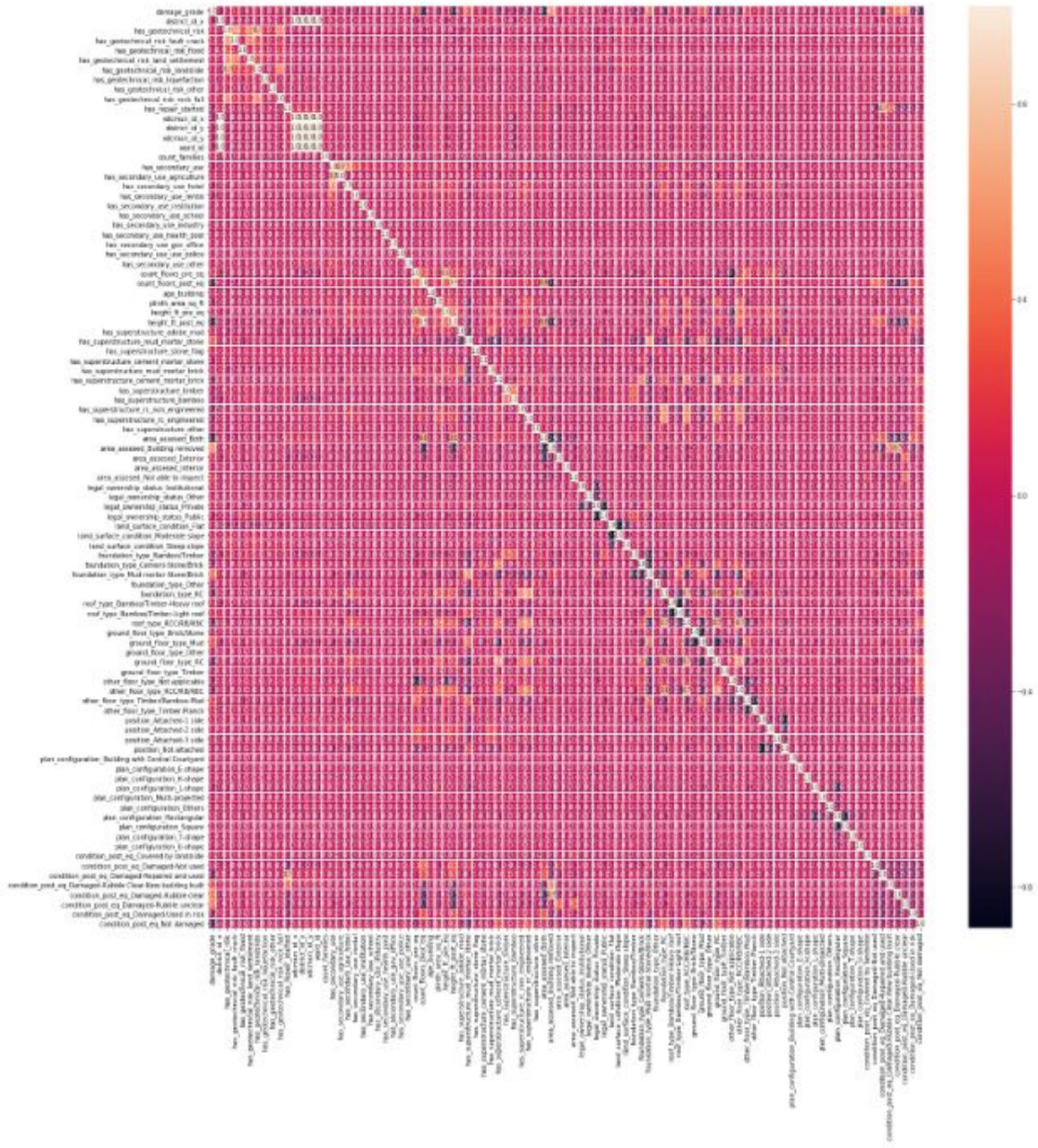

Figure 2: Heatmap of Training Data

2.3. Algorithms and Techniques

To train models, we choose 3-4 different models to compare. We compared logistic regression, decision trees , random forest and nearest-neighbors since this is a classification problem.

Logistic Regression

In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model; it is a form of binomial regression. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name.

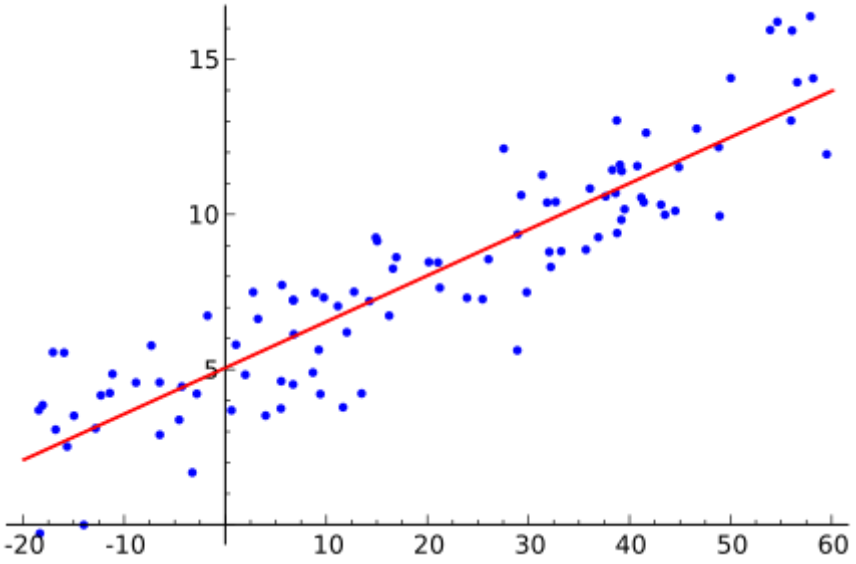

Figure 3: Regression Analysis

Decision Trees

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

In decision analysis, a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values(or expected utility) of competing alternatives are calculated.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

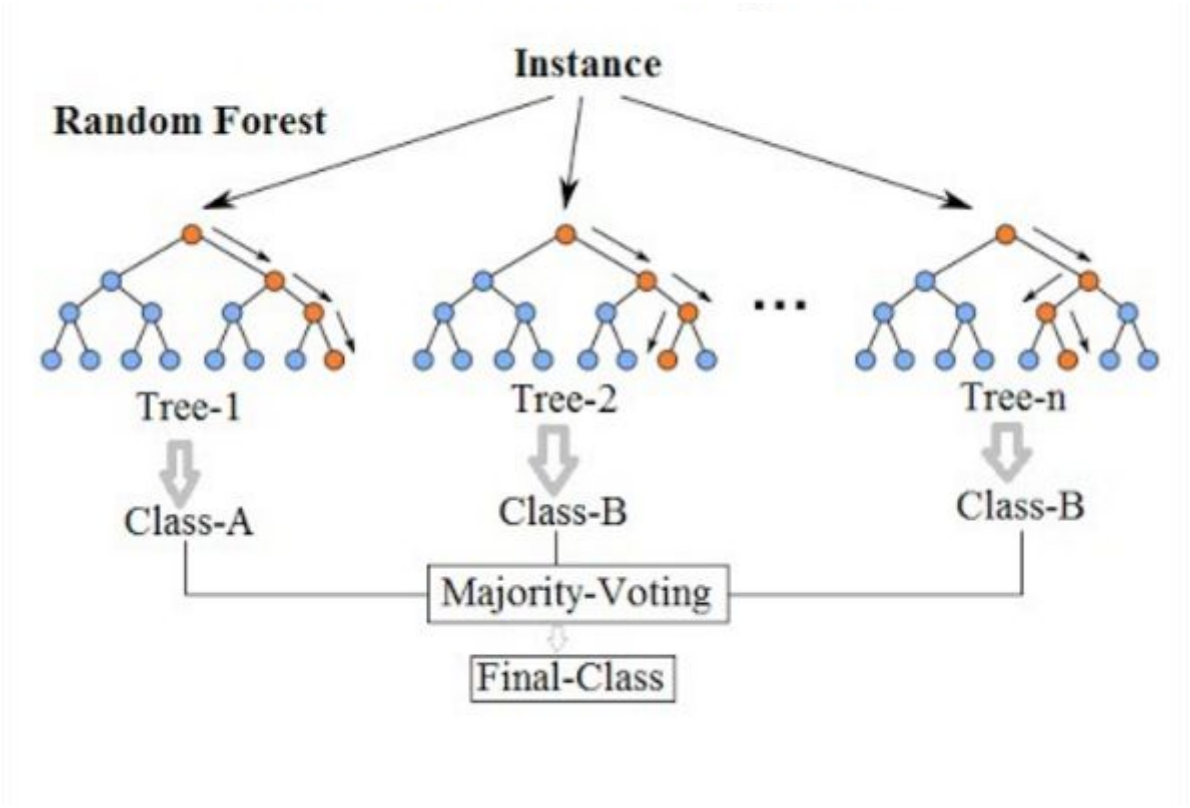

Figure 4 : Random Forest Simplified

K Nearest neighbors

In pattern recognition, the *k*-nearest neighbors algorithm (*k*-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.
- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms.

2.4. Benchmark

The benchmark model we chose was Linear Regression model. Linear regression is a common Statistical Data Analysis technique. It is used to determine the extent to which there is a linear relationship between a dependent variable and one or more independent variables. Prediction results was evaluated based on F1 Score with 'weighted' average. We get the Training Score = 78.03937701315711% after using Linear Regression model.

3. Methodology

3.1. Data Processing

First, all the null attributes are removed. Then the building_Ownership_Use.csv and Building_Structure.csv are merged and stored in the 'merge' variable which then is merged with the training data in reference to the building_id. Then I stored it in merge_file.csv. After

which I replaced all the 5 grade names with their respective grades such as ‘Grade 1’ to ‘1’, etc , then dropped the building_id column and then filled all the null values with the dummy data. I then performed the same procedure for the testing data.

3.2. Implementation

1. In **model 1** a Linear Regression benchmark model is used for measuring the performance. An initial **benchmark** Linear Regression model was defined and then trained on the pre-processed training data and damaged grade. After training the model, the Training Accuracy = 78.03937701315711% .

2. In **model 2** I used Logistic Regression model for measuring the performance. Logistic Regression classifier with a random_state=0, solver='lbfgs', multi_class='multinomial' is used and then I trained the model, the Training Accuracy = 46.82371972945465% .

3. In **model 3** I used Decision Tree Classifier model for measuring the performance. Decision Tree classifier model was defined and then trained on the pre-processed training data and damaged grade. After training the model, the Training Accuracy = 99.80166550325202% .

4. In **model 4** I used Decision Tree Regressor for measuring the performance. Decision Tree regressor model was defined and then trained on the pre-processed training data and damaged grade. After training the model, the Training Accuracy = 99.91836297287398% .

5. In **model 5** I used K-Neighbors Classifier for measuring the performance. K-Neighbors model was defined with number of neighbors = 3 and then trained on the pre-processed training data and damaged grade. After training the model, the Training Accuracy = 78.19713467592966% .

6. In **model 6** I used Random Forest Classifier for measuring the performance. Random Forest model was defined with number of estimators = 20 and then trained on the

pre-processed training data and damaged grade. After training the model, the Training Accuracy = 99.56201791500267% .

3.3. Refinement

1. Model 2 - This model shows a great downfall of accuracy when compared with the benchmark model. I used Logistic Regression classifier with a random_state=0, solver='lbfgs', multi_class='multinomial' and get the Training Accuracy = 46.82371972945465% .

2. Model 3 - This model shows a significant improvement in accuracy when compared with the benchmark model and the model 2 as this model was able to achieve a Training Accuracy = 99.80166550325202% .

3. Model 4 - This model shows a very slight increase in accuracy when compared with the model 3 but significant improvement when compared with benchmark model and model 2 as this model achieve a Training Accuracy = 99.91836297287398%.

4. Model 5 - This model shows a very slight increase in accuracy when compared with benchmark model and achieve a Training Accuracy = 78.19713467592966%.

5. Model 6 - This model shows a similarity with very slight increase in accuracy when compared with the model 3 and model 4 but significant improvement when compared with benchmark model, model 5 and model 2 as this model achieve a Training Accuracy= 99.56201791500267%.

4. Results

4.1. Model Evaluation and Validation

The model 4 outperforms the benchmarks and other models used in this project. It takes nearly half of the time taken by the other models and uses Decision Tree Regressor. The

advantage of Decision Tree Regressor is its ability to assign specific values to problem, decisions, and outcomes of each decision. This reduces ambiguity in decision-making.

4.2. Justification

The final model performs better than the benchmark and other model as well. Where benchmark model gets a accuracy of 78.03937701315711%, the final model made a significant improvement of 21.87% (approx) and get a accuracy of 99.91836297287398%

Accuracy of different Models used in the project:-

Model	Accuracy
Benchmark model	78.03937701315711%
Logistic Regression	46.82371972945465%
Decision Tree Classifier	99.80166550325202%
Decision Tree Regressor	99.91836297287398%
K-Neighbors Classifier	78.19713467592966%
Random Forest	99.56201791500267%

Table 2: Accuracy of each model

Conclusion

Free-Form Visualization

First, all the null attributes are removed. Then the building_Ownership_Use.csv and Building_Structure.csv are merged and stored in the 'merge' variable which then is merged with the training data in reference to the building_id. Then I stored it in merge_file.csv. After which I replaced all the 5 grade names with their respective grades such as 'Grade 1' to '1', etc , then dropped the building_id column and then filled all the null values with the dummy data. I then performed the same procedure for the testing data.

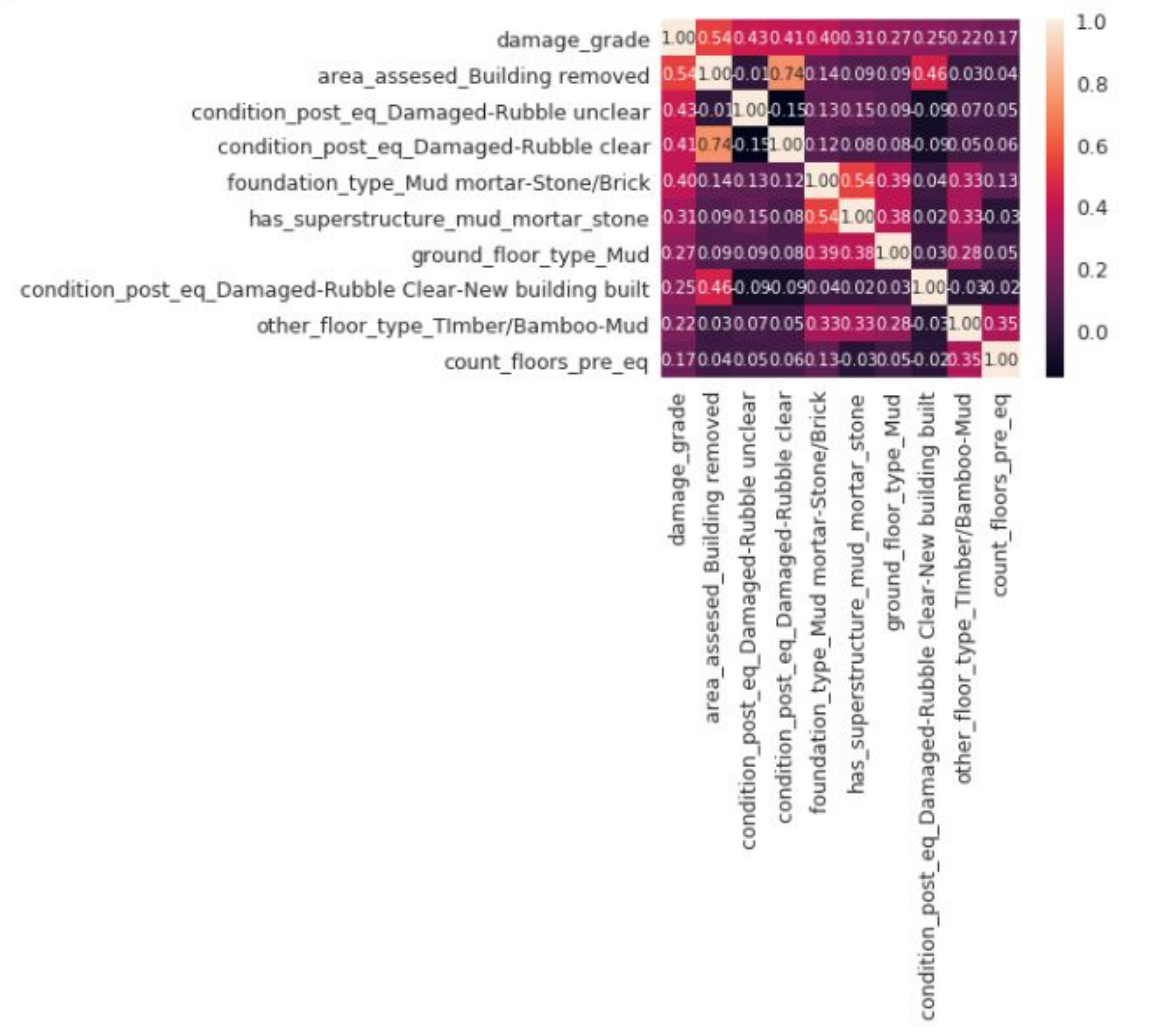

Figure 5: Final heatmap visualization of training data

Reflection

In this project, I created a model for predicting the damage to a building. First I build a benchmark model of Linear Regression. Then I used other models to train such as Logistic Regression, Decision Trees , Random forest and K-nearest-neighbors. The machine learning models which achieved better results than the benchmark model were Decision Tree Classifier, Decision Tree Regressor and Random Forest among which Decision Tree Regressor shows the outstanding results followed by Decision tree classifier. In Logistic Regression the accuracy was decreased compared to the benchmark model and in K-nearest-neighbors, it is similar to the benchmark model. The Decision tree based models perform better in the field of classification and therefore the results of the project are as expected. However, in the beginning, I thought that the Random Forest model will be capable

to learn even from unnormalized noisy data, and this was not true. I spent a lot of time with different models and parameters tuning, however without any good results. I will look forward doing some more projects to learn more about them.

Improvements

As the accuracy is nearly 99.9% so there is not much need of improvement, but by using more models to train the data, more better accuracy could be expected.

References

- https://en.wikipedia.org/wiki/Earthquake_prediction
- http://www.iitk.ac.in/nicee/wcee/article/14_S10-032.PDF
- https://en.wikipedia.org/wiki/F1_score
- https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- https://en.wikipedia.org/wiki/Logistic_regression
- https://en.wikipedia.org/wiki/Random_forest
- https://en.wikipedia.org/wiki/Decision_tree

Domain background

- https://www.hackerearth.com/problem/machine-learning/predict-the-energy-used-612632a9-3f496e7f/description/

Datasets

- https://he-s3.s3.amazonaws.com/media/hackathon/machine-learning-challenge-6-1/predict-the-energy-used-612632a9-3f496e7f/a490e594-6-Dataset.zip