HACK HUB HACKATHON-2019 -Review 3

TEAM- SUDO NINJAS

ANALYSIS OF BUILDING DAMAGE BY EARTHQUAKE USING MACHINE LEARNING (AI)

Problem Statement

Extent and damage of earthquake 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. In this challenge we provide you with the before and after details of nearly one million buildings after an earthquake. The damage to a building is categorized in five grades. Each grade depicts the extent of damage done to a building post an earthquake. Given building details, your task is to build a model that can predict the extent of damage that has been done to a building after an earthquake.

Introduction/abstract

Earthquake is one of the natural disasters that happens in the nature. One of the main prey for such disasters are buildings. Buildings get affected in more bad manner leading them to get worsened in their characteristics which in turn makes the people reside inside them unsafe. Hence to reduce this stress faced by the people, we aim to determine the prominent factors causing damage to the buildings after an earthquake by, using the dataset which has ample features compared between buildings before and after an earthquake.The dataset also provides us with a measure of the damage caused to various types of buildings in the form of grade of damage metric which has 5 classes representing the amount of damage caused to the building. The dataset is being seen with our

perspective by thesThe Poor construction technique, where slab walls and floors are not tied together correctly, for example, makes buildings far more vulnerable to earthquake damage; buildings where the bricks have been held in place with the correct mortar tend to survive much better. However, there is more to the issue of building design than simply the difference between careful and shoddy construction. Based on all these factors. The dataset given is analysed accordingly. The target variable of the dataset is chosen to be 'Damage_grade' which is the multi labelled variable. Classification algorithms are used here to classify.

Specifications with architecture

Specifications:

Given that there were 3 Datasets for training purpose.

1. Train.csv
2. Building_Ownership_Use.csv
3. Building_Structure.csv

Attribute Count:

There were 14 attributes in Train.csv including the output variable. In which the **risk factors of the Buildings** were used.

There were 17 attributes in the Building Owner_Ship.csv in which the **secondary uses of the building** is mentioned if the part of house is used for any other purpose.

There were 29 attributes in the Building Structure.csv in which the **dimensions of the building** are mentioned.

When closely observed, we shall sense that there is a relationship between the Attributes that describe the building characteristics, secondary uses and the risks related to the building are interrelated. Hence we **merge the data in them with respect to the Building Id's** given and form a new dataset called 'mydf.csv'. Figure 1.1 shows the relationship between each and every variables in the form of the heat map. Here abs(correlation(df)) is being shown.

HEAT MAP OF THE CORRELATION BETWEEN THE VARIABLES

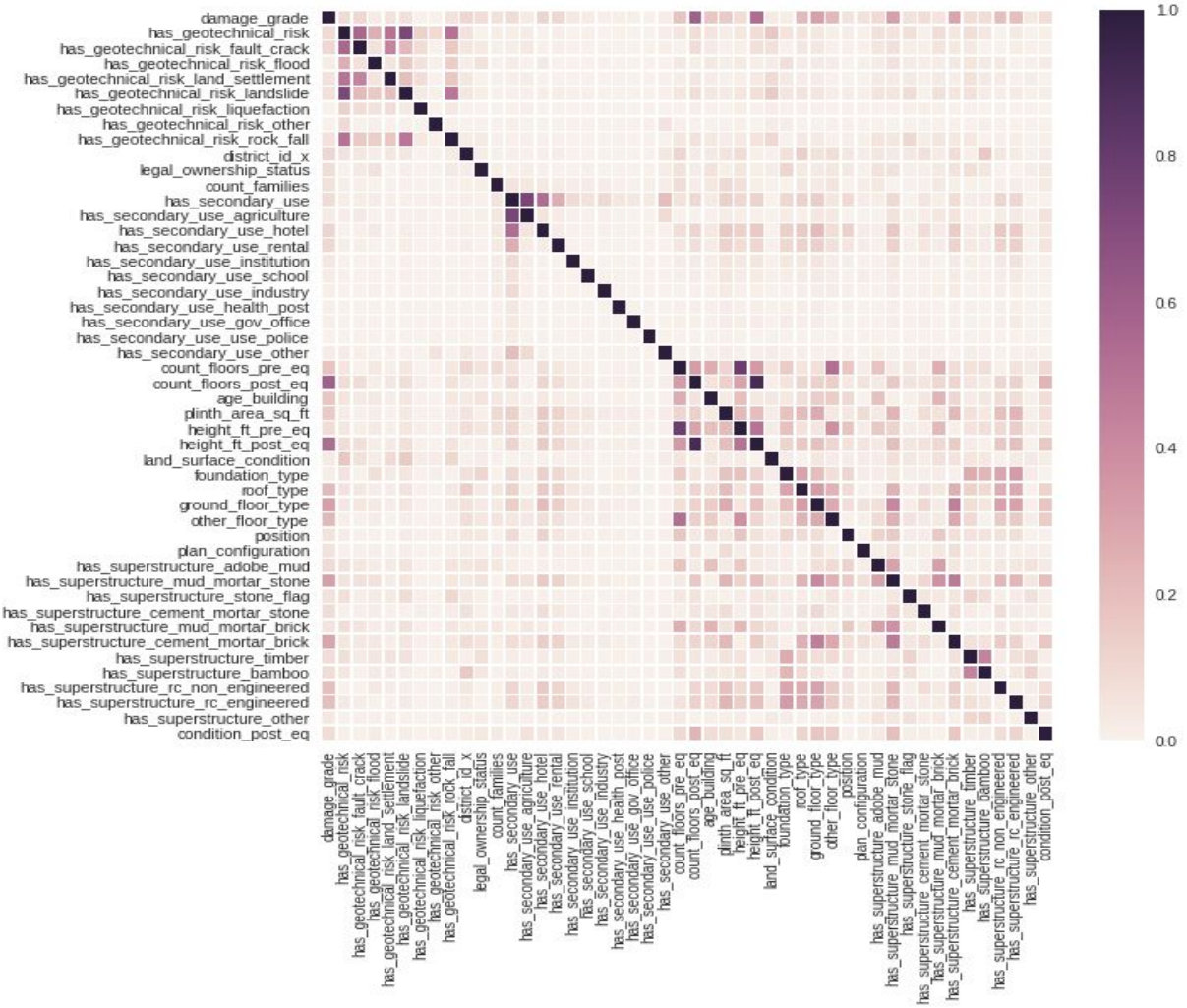

Fig 1.1

Now, when we analyse about the output variable i.e. “Damage Grades”, the distribution or the headcount of the “damage grade“ is as follows:

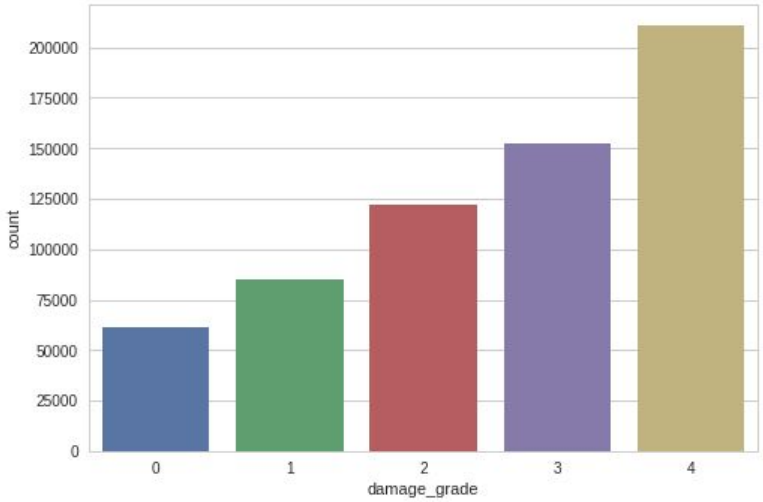

Fig 1.2

There were Continuous valued attributes in the mydf.csv dataset. (Combined). They are: ("count_floors_pre_eq", "count_floors_post_eq", "age_building", "plinth_area_sq_ft", "height_ft_pre_eq", "height_ft_post_eq", "count_families"). The plot of them with respect to the output are shown in Fig 1.3

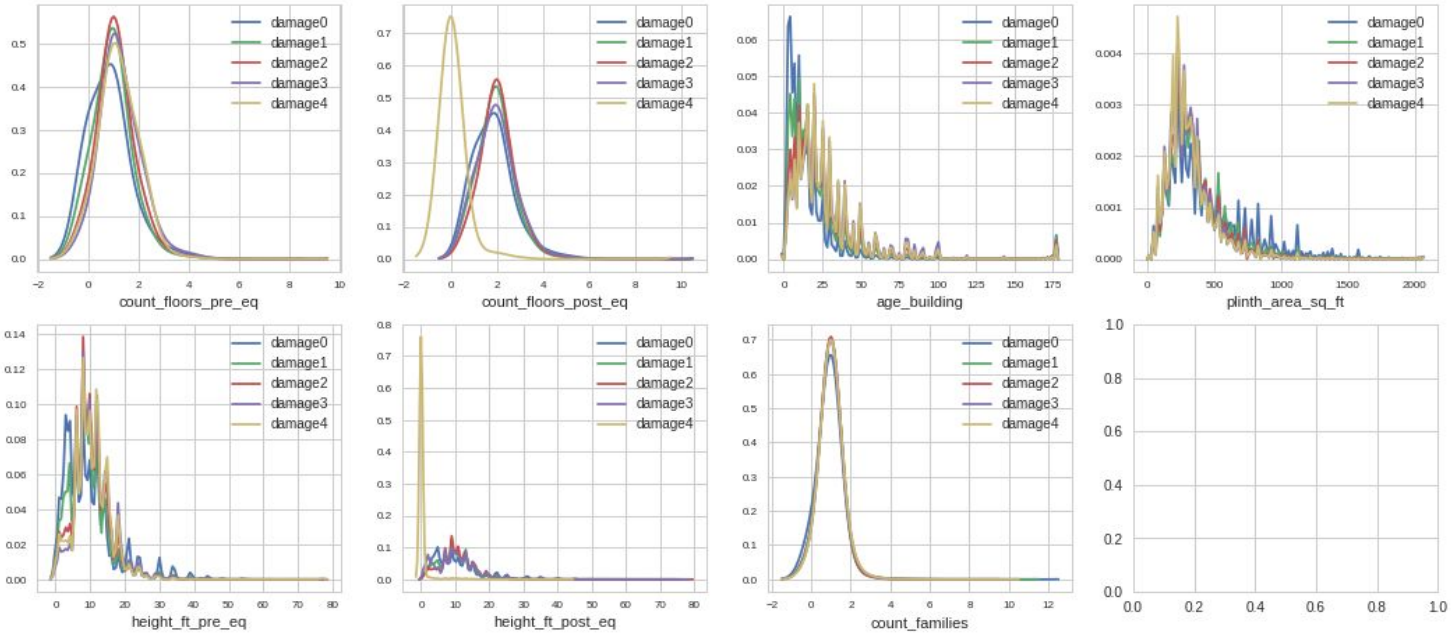

Fig 1.3

The correlation between these variables with respect to the damage_grade attribute are given in the figure 1.4. We can see that much of the variables are positively correlated with respect to the final output. Hence we shall perform some feature engineering after this analysis using the figure 1.1 and 1.4.

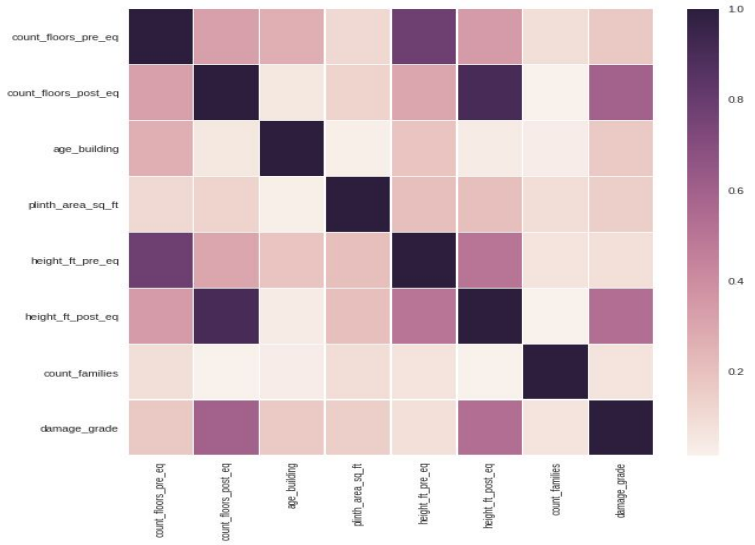

Fig 1.4

Methodology

Big data is the huge pool of collection of data. This required information from this pool can be acquired from few well defined processes. One of such processes is the Process of Knowledge Discovery in Databases (KDD). KDD is the process of discovering useful knowledge from a collection of data. KDD follows the sequence of steps namely aggregation, Preprocessing, Data Modeling and Data Analysis. For the Data Modeling, one requires the Machine learning algorithms from which the proper hypothesis can be framed.

METHOD-1:

The proposed model uses the techniques of Feature Engineering, Principle Component Analysis to preprocess the data and the technology of Stacking involving the Ensemble approaches of Random Forests, Extra Trees with Boosting as well as the Neural Network

approaches. The results of the proposed stacking model for the forest cover classification is done with its base models. The analysis of the results of the proposed Stacking model with it's base model are also done.

Before that, PCA (Principle Component Analysis) is applied over the feature engineered dataset. Principle Component analysis technique reduces the dimensionality without disturbing the overall features of the dataset. The goal of PCA is to reduce dimensions into an occasional dimensional area that most of the essential information contained in an exceedingly high dimensional area. The given data comprises of many more added features during feature engineering too. Hence the redundant features shall be removed for the maximum performance of the data model. Fig 2.1 shows a piece of code where the number of rows were reduced to 54 after PCA application.

```
all_data=x
all_data_t=x.T
all_data_t
covar=np.cov(all_data_t)
covar
eigen_val, eigen_vectors=np.linalg.eig(covar)
eigen_val_vector_pair=[]

for i in range(len(eigen_val)):
    eig_vec=eigen_vectors[:, i]
    eigen_val_vector_pair.append((eigen_val[i], eig_vec))
eigen_val_vector_pair.sort(reverse=True)
```

```
counter_eig=0
for i in (eigen_val):
    if(i>0):
        counter_eig+=1
counter_eig
```

```
54
```

Fig 2.1

The 3 major datasets are combined together and then are being trained using different models. Instead of analysing using various models, to increase the accuracy, all the used models' output are taken as the input to another meta classifier and are stacked to give the proper optimised output.

The used sub models are:

Random Forests

Extra Trees (With Applied Ada Booster)

MLP (Using 'tanh' activator)

Meta classifier used: Gradient Boosting Classifier.

This method is used to tackle the Imbalance nature of the dataset. Hence The Random Sampling with replacement takes place when used the bootstrapping. Henceforth the model gave the initial accuracy of 71%.

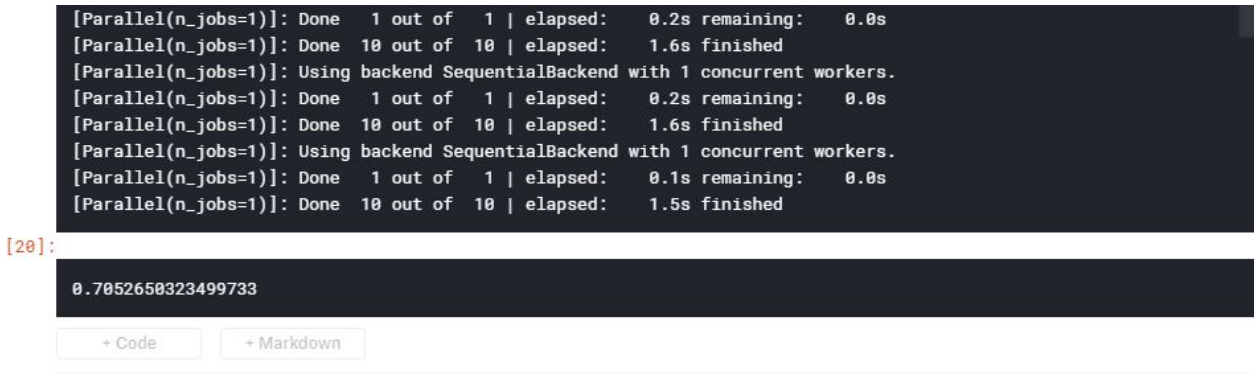

Fig 2.2

Method-2:

The feature Engineering is the main technique to add the weightage to the data. It always makes up the pre-processing part of the dataset complete. Hence we're using the Feature Engineering and the feature dependency as the major factor to predict outcomes. We first find the chi square testing with the categorical values with the rest of the variables and based on the test results we form the features by feature engineering. And then split the train data to train and validation data. Later train it with the Random forests classifier.

Here we got the accuracy upto 75%.

System description

Our system had two models.

One used the Bootstrap aggregation by means of Stacking the following things using meta classifier of Gradient Descent Boosting.

Random Forests

Extra Trees (With Applied Ada Booster)

MLP (Using ‘tanh’ activator)

This technology is used to tackle the imbalanced nature of the dataset.

Second Method is about the Feature Engineering. We’ll be using the Feature Engineering and the feature dependency as the major factor to predict outcomes. We first find the chi square testing with the categorical values with the rest of the variables and based on the test results we form the features by feature engineering. And then split the train data to train and validation data. Later train it with the Random forests classifier.

Results and conclusions/inferences

Hence forth We eloquently analyse by running the models on the data set to provide viable outputs upto 75% accuracy.