**REPORT**

1. **EARTHQUAKE IN NEPAL DATASET (BONUS PROJECT)**

**II. SIGNIFICANT EARTHQUAKE DATASET**

**III.WINE DATASET**

**IV. CONCLUSION**

I have choosed 3 data sets in which 2 contains huge data about Earthquake and 3rd data set which contains less amount than the other two data set. I have Normalized the data set and done the preprocessing, visualization and clustering on all three data sets.

The datasets of earthquake contained data of all the attributes like date , time , integer, objects, float etc.

|  |  |
| --- | --- |
| **DATA SETS** | **SIZE** |
| EARTHQUAKE NEPAL | 260601 rows, 39 columns |
| SIGNIFICANT EARTHQUAKE | 23412 rows × 21 columns |
| WINDSET | 178 rows × 13 columns |

I.Modeling Nepal Earthquake damage

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 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.

Thata data set has been downloaded from <https://www.drivendata.org/competitions/57/nepal-earthquake/>

<https://www.kaggle.com/mullerismail/richters-predictor-modeling-earthquake-damage/activity>

1. **READING DATA SET AND DATA PREPROCESSING**

The data set has following attributes

* ***geo\_level\_1\_id, geo\_level\_2\_id, geo\_level\_3\_id :*** 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 :*** number of floors in the building before the earthquake.
* ***age (type: int):*** age of the building in years.
* ***area\_percentage :*** normalized area of the building footprint.
* ***height\_percentage :*** normalized height of the building footprint.
* ***land\_surface\_condition :*** surface condition of the land where the building was built. Possible values: n, o, t.
* ***foundation\_type :*** type of foundation used while building. Possible values: h, i, r, u, w.
* ***roof\_type :*** type of roof used while building. Possible values: n, q, x.
* ***ground\_floor\_type :*** type of the ground floor. Possible values: f, m, v, x, z.
* ***other\_floor\_type :*** type of constructions used in higher than the ground floors (except of roof). Possible values: j, q, s, x.
* ***position :*** position of the building. Possible values: j, o, s, t.
* ***plan\_configuration :*** building plan configuration. Possible values: a, c, d, f, m, n, o, q, s, u.
* ***has\_superstructure\_adobe\_mud :*** flag variable that indicates if the superstructure was made of Adobe/Mud.
* ***has\_superstructure\_mud\_mortar\_stone):*** flag variable that indicates if the superstructure was made of Mud Mortar - Stone.
* ***has\_superstructure\_stone\_flag :*** 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 :*** flag variable that indicates if the superstructure was made of Mud Mortar - Brick.
* ***has\_superstructure\_cement\_mortar\_brick :*** flag variable that indicates if the superstructure was made of Cement Mortar - Brick.
* ***has\_superstructure\_timber***  indicates if the superstructure was made of Timber.
* ***has\_superstructure\_bamboo***  indicates if the superstructure was made of Bamboo.
* ***has\_superstructure\_rc\_non\_engineered*** indicates if the superstructure was made of non-engineered reinforced concrete.
* ***has\_superstructure\_rc\_engineer*** if the superstructure was made of engineered reinforced concrete.
* ***has\_superstructure\_other*** if the superstructure was made of any other material.
* ***legal\_ownership\_status*** legal ownership status of the land where building was built. Possible values: a, r, v, w.
* ***count\_families :*** number of families that live in the building.
* ***has\_secondary\_use***flag variable that indicates if the building was used for any secondary purpose.
* ***has\_secondary\_use\_agriculture***  flag variable that indicates if the building was used for agricultural purposes.
* ***has\_secondary\_use\_hotel :*** flag variable that indicates if the building was used as a hotel.
* ***has\_secondary\_use\_rental :*** flag variable that indicates if the building was used for rental purposes.
* ***has\_secondary\_use\_institution:*** flag variable that indicates if the building was used as a location of any institution.
* ***has\_secondary\_use\_school :*** flag variable that indicates if the building was used as a school.
* ***has\_secondary\_use\_industry ):*** flag variable that indicates if the building was used for industrial purposes.
* ***has\_secondary\_use\_health\_post :*** flag variable that indicates if the building was used as a health post.
* ***has\_secondary\_use\_gov\_office):*** flag variable that indicates if the building was used fas a government office.
* ***has\_secondary\_use\_use\_polic:*** flag variable that indicates if the building was used as a police station

After configuring the dataset In jupyter notebook and finding the dtypes we found that

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 260601 entries, 0 to 260600

Data columns (total 39 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 building\_id 260601 non-null int64

1 geo\_level\_1\_id 260601 non-null int64

2 geo\_level\_2\_id 260601 non-null int64

3 geo\_level\_3\_id 260601 non-null int64

4 count\_floors\_pre\_eq 260601 non-null int64

5 age 260601 non-null int64

6 area\_percentage 260601 non-null int64

7 height\_percentage 260601 non-null int64

8 land\_surface\_condition 260601 non-null object

9 foundation\_type 260601 non-null object

10 roof\_type 260601 non-null object

11 ground\_floor\_type 260601 non-null object

12 other\_floor\_type 260601 non-null object

13 position 260601 non-null object

14 plan\_configuration 260601 non-null object

15 has\_superstructure\_adobe\_mud 260601 non-null int64

16 has\_superstructure\_mud\_mortar\_stone 260601 non-null int64

17 has\_superstructure\_stone\_flag 260601 non-null int64

18 has\_superstructure\_cement\_mortar\_stone 260601 non-null int64

19 has\_superstructure\_mud\_mortar\_brick 260601 non-null int64

20 has\_superstructure\_cement\_mortar\_brick 260601 non-null int64

21 has\_superstructure\_timber 260601 non-null int64

22 has\_superstructure\_bamboo 260601 non-null int64

23 has\_superstructure\_rc\_non\_engineered 260601 non-null int64

24 has\_superstructure\_rc\_engineered 260601 non-null int64

25 has\_superstructure\_other 260601 non-null int64

26 legal\_ownership\_status 260601 non-null object

27 count\_families 260601 non-null int64

28 has\_secondary\_use 260601 non-null int64

29 has\_secondary\_use\_agriculture 260601 non-null int64

30 has\_secondary\_use\_hotel 260601 non-null int64

31 has\_secondary\_use\_rental 260601 non-null int64

32 has\_secondary\_use\_institution 260601 non-null int64

33 has\_secondary\_use\_school 260601 non-null int64

34 has\_secondary\_use\_industry 260601 non-null int64

35 has\_secondary\_use\_health\_post 260601 non-null int64

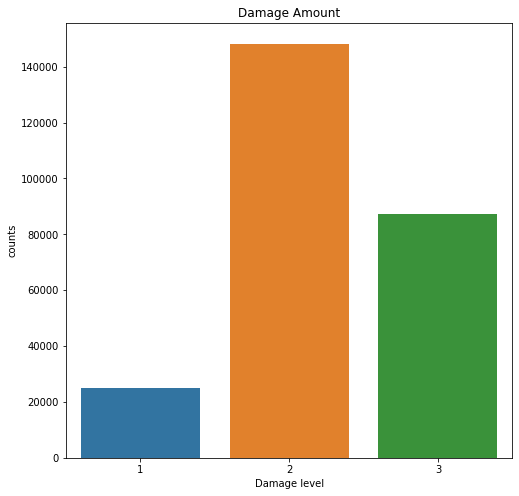
36 has\_secondary\_use\_gov\_office 260601 non-null int64

37 has\_secondary\_use\_use\_police 260601 non-null int64

38 has\_secondary\_use\_other

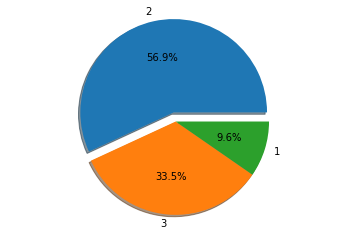
We omitted the building id since it’s just the number indicating the buildings. WE the converted all the object files into the integer file. There were no missing values in this data set

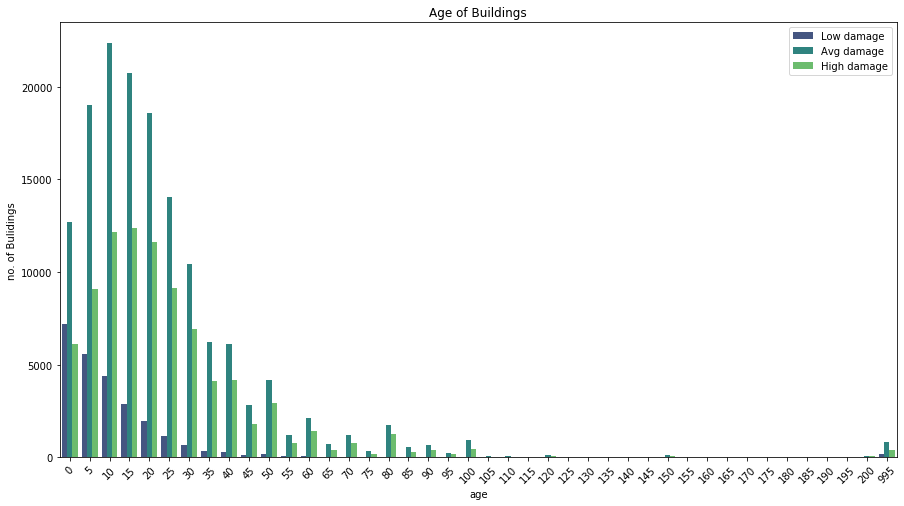
1. VISUALIZING THE DATA SET



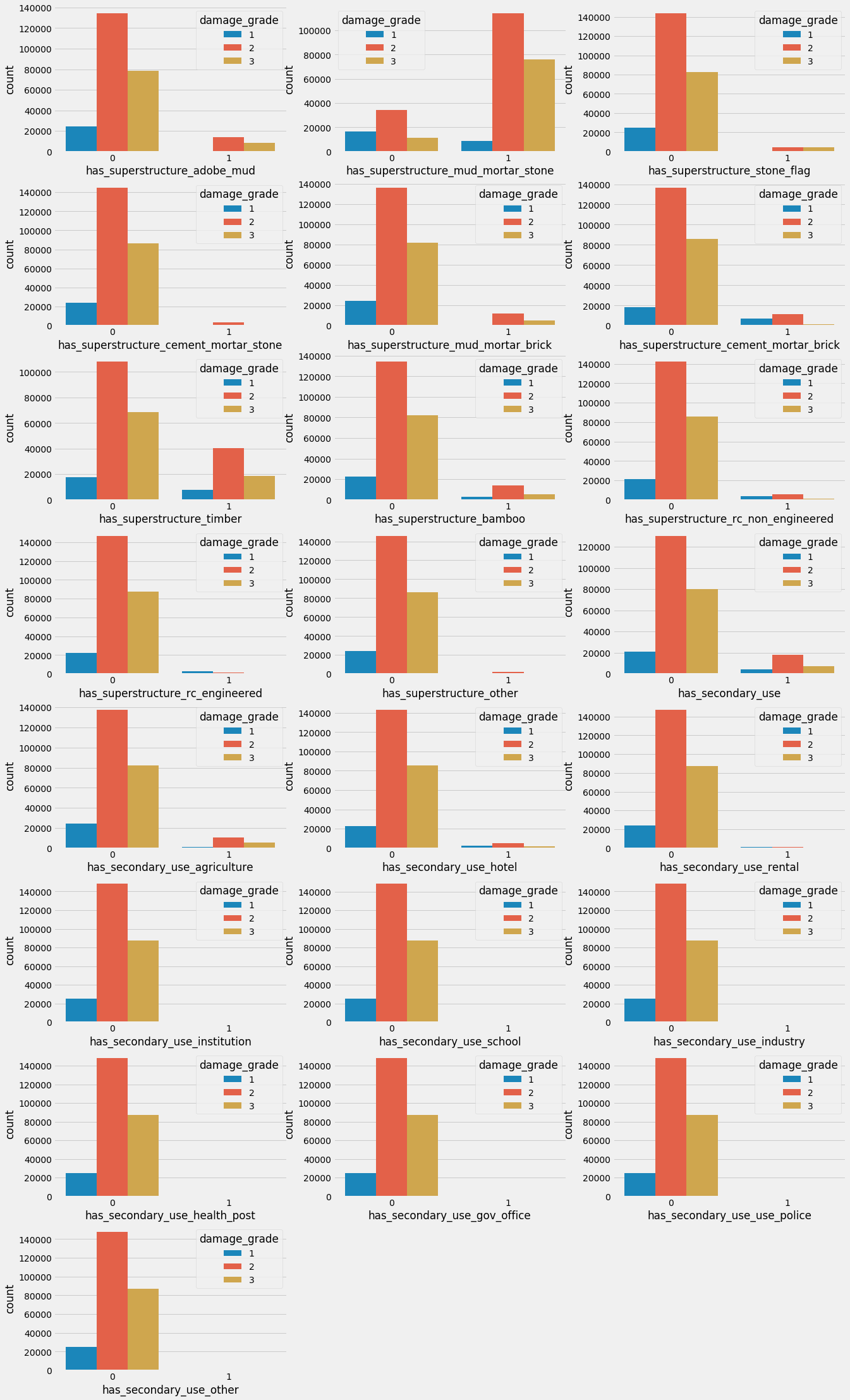
This is amount of buildings categorized by the damage level 1, little damage, 2, Sufficient damage and 3 as a heavy damage

And below is the percentage of the damage grade

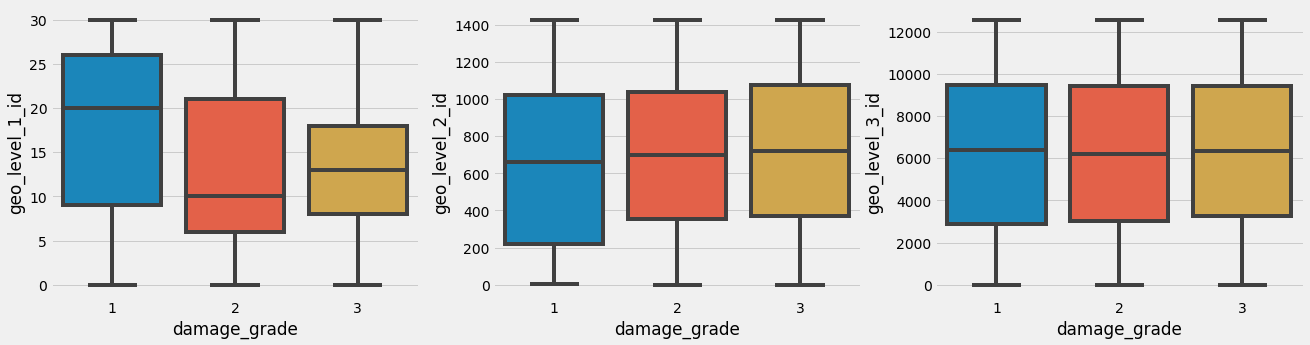




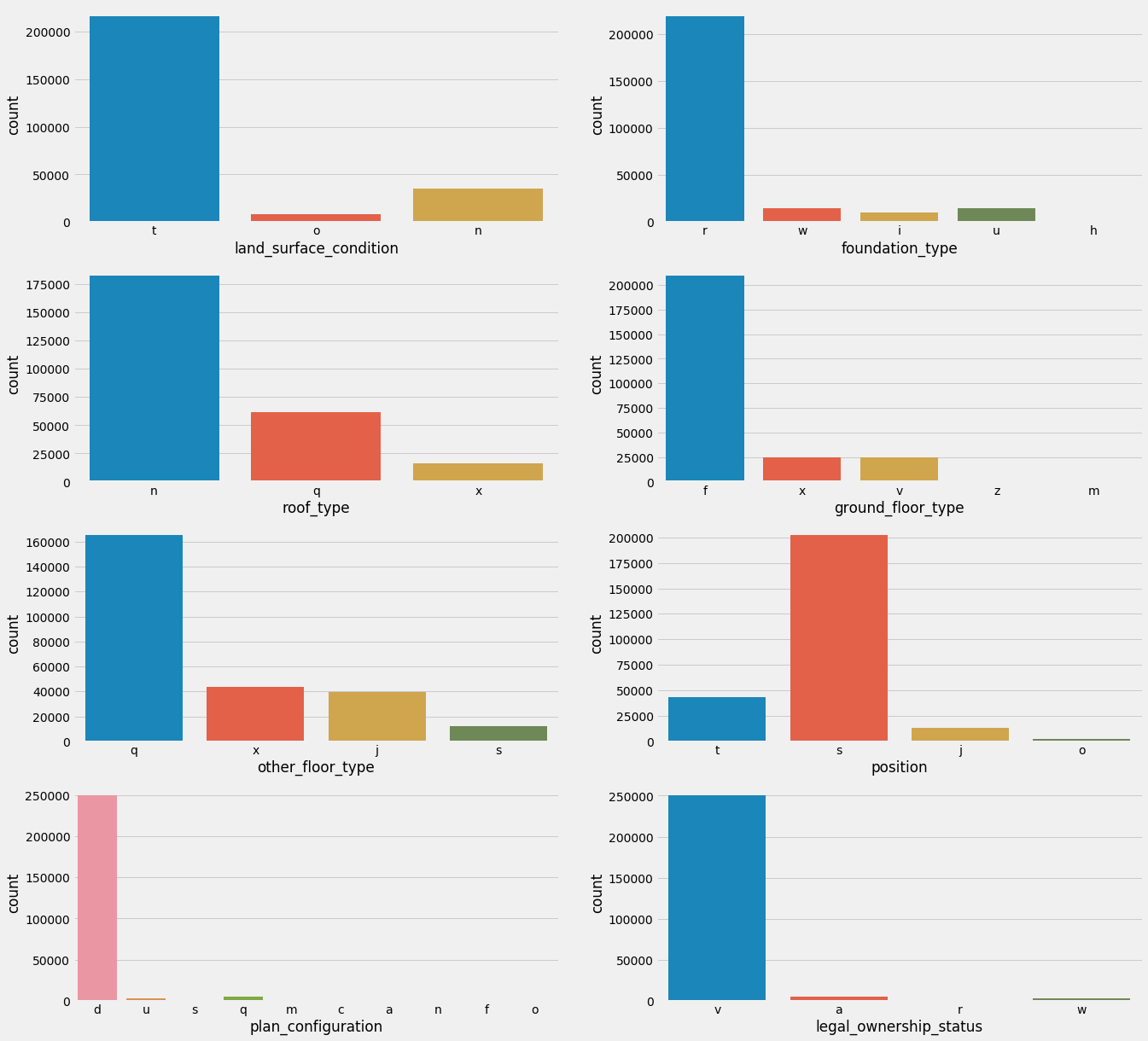
The above graph indicates the age of the building in the Nepal, where most of the data has been from the age group 0-50 age and there are also very few building which are more than 200 year old( old temples and monuments)



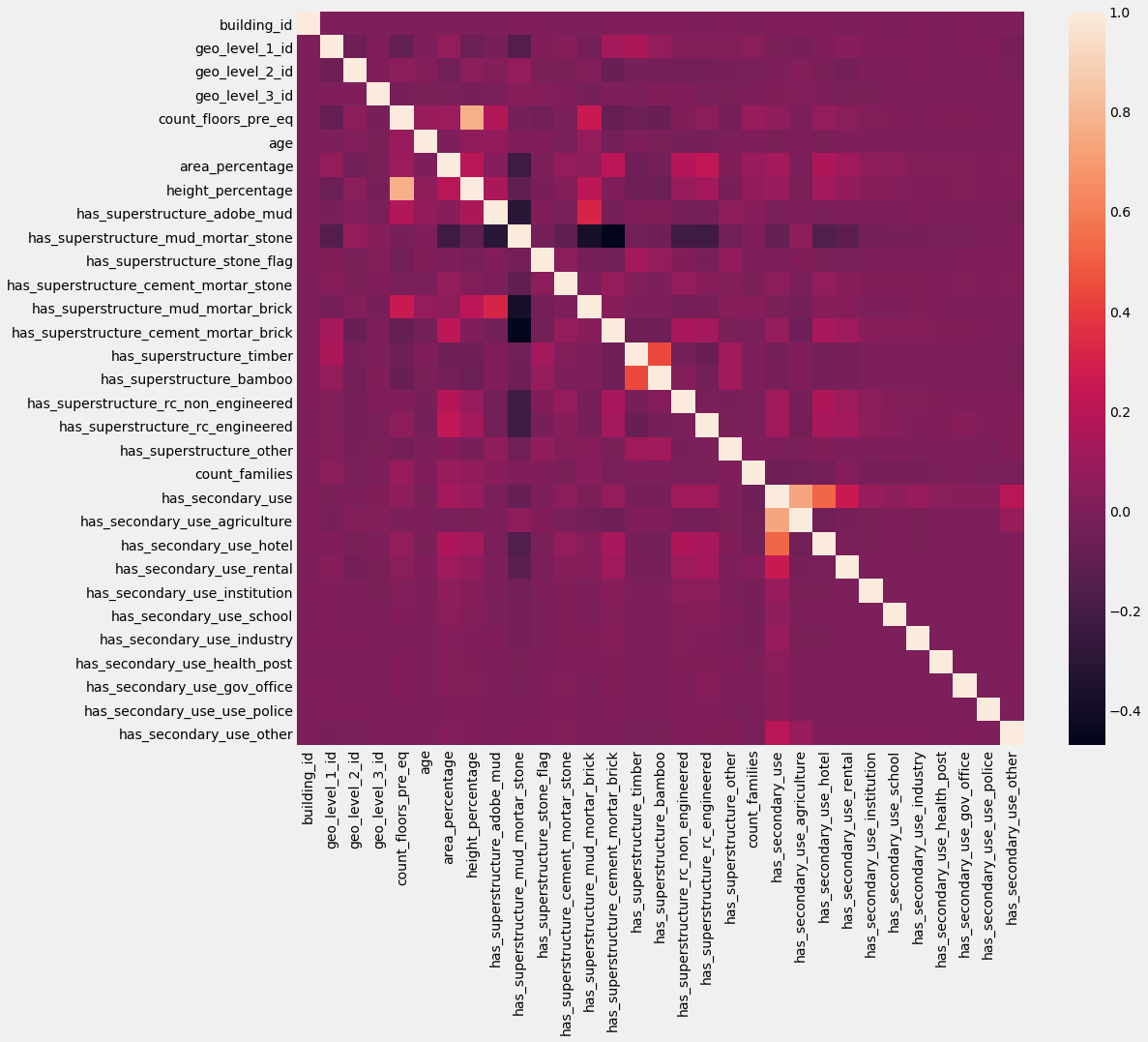
The level of damage grade according to the structure of the house.



The above graph indicates the box plot of the damage grade with respect to the geo level id 1 , 2 and 3



The above graph indicates damage grade according to the surface conditions,

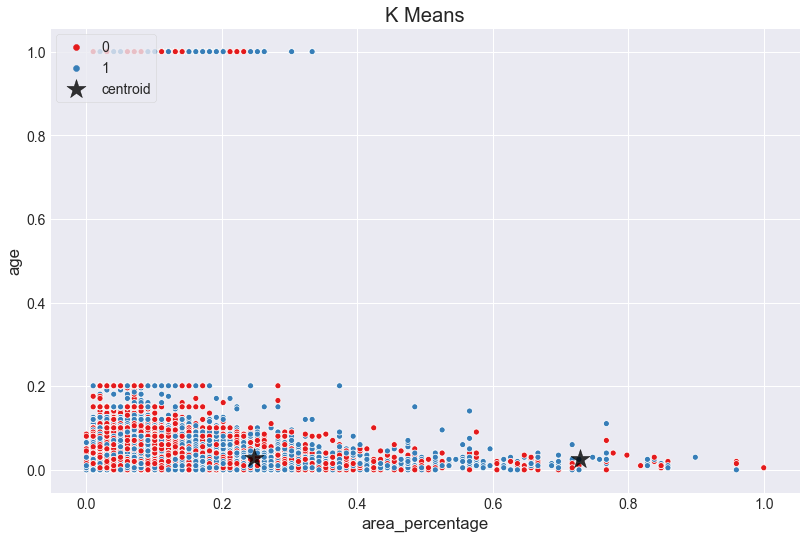


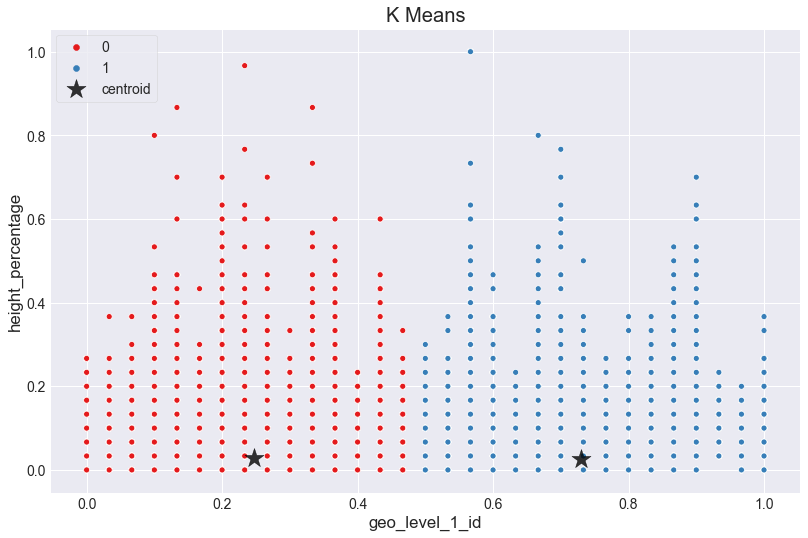
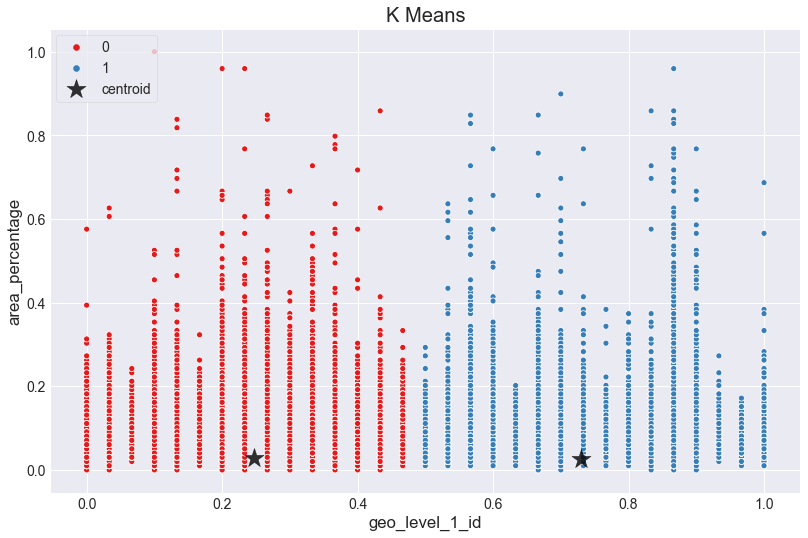
HEAT MAP OF ALL CONDITIONS

3.NORMALIZING THE DATA AND PERFORM CLUSTERING

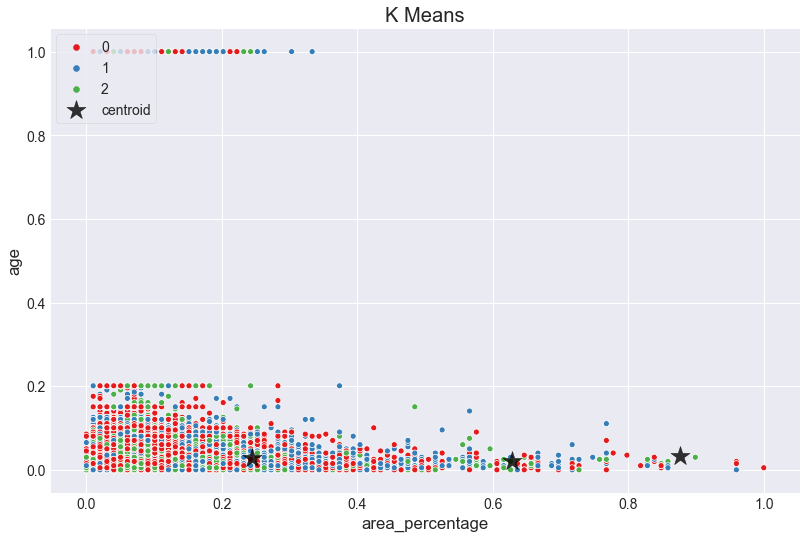
There are two methods of clustering that i implemented in the project , one is the normal traditional method and another with sklearn library. After performing the clustering thorugh the mean of K mean sklearn library we obtain the following graphs

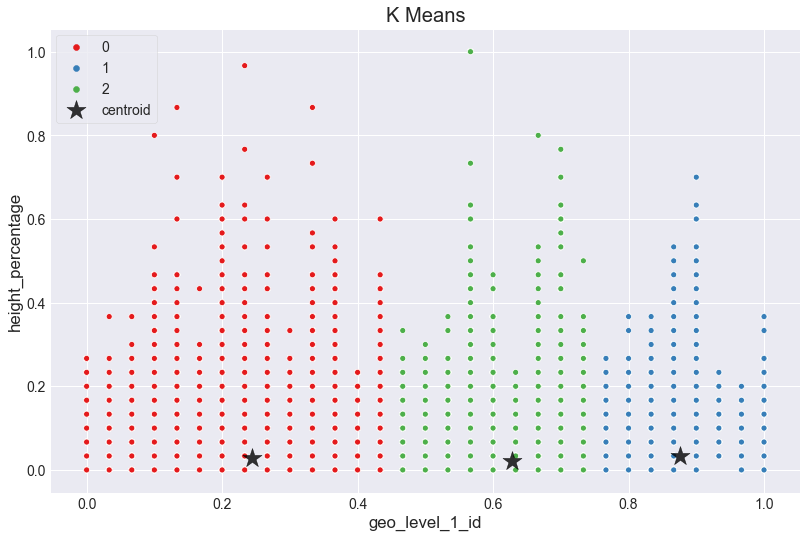
Choosing number of cluster as 2

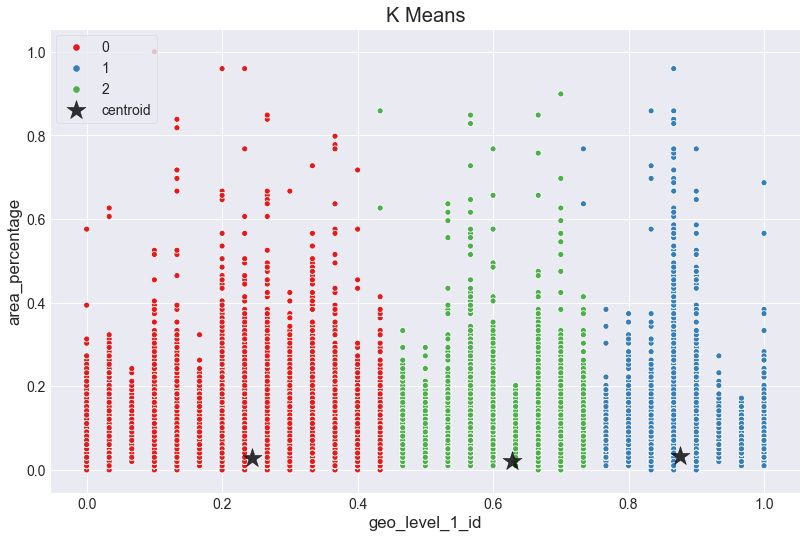




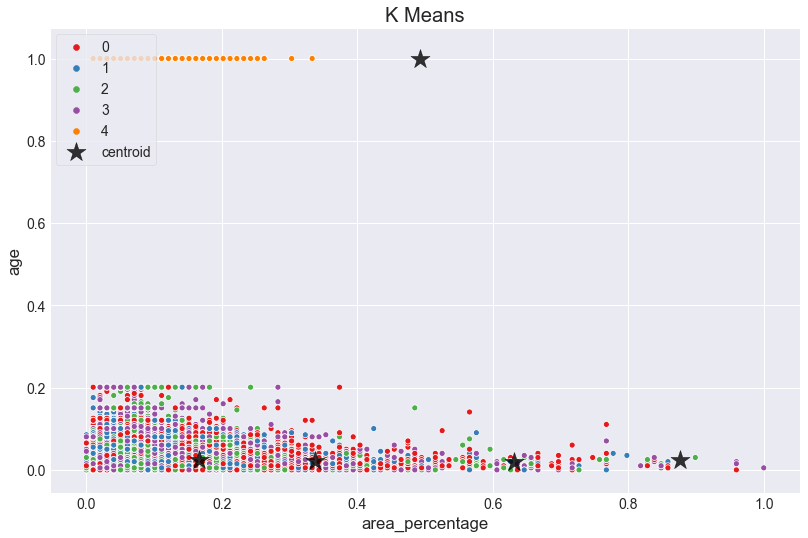
Using number of cluster as 3

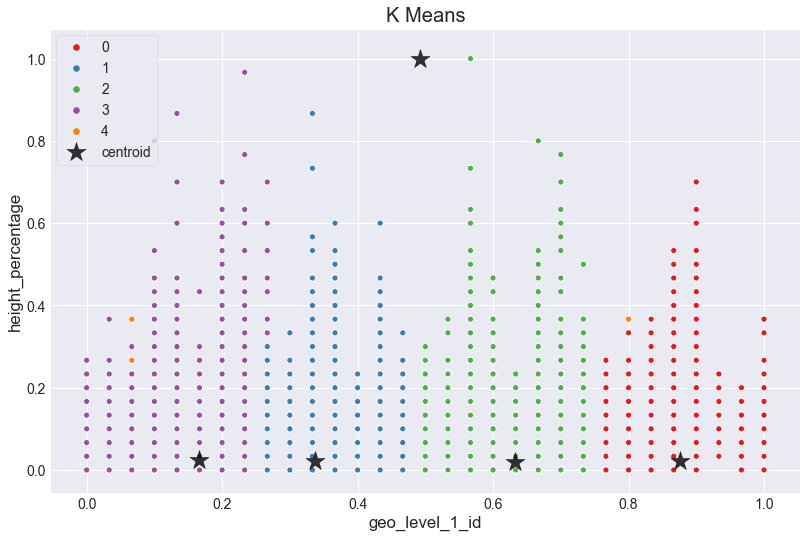


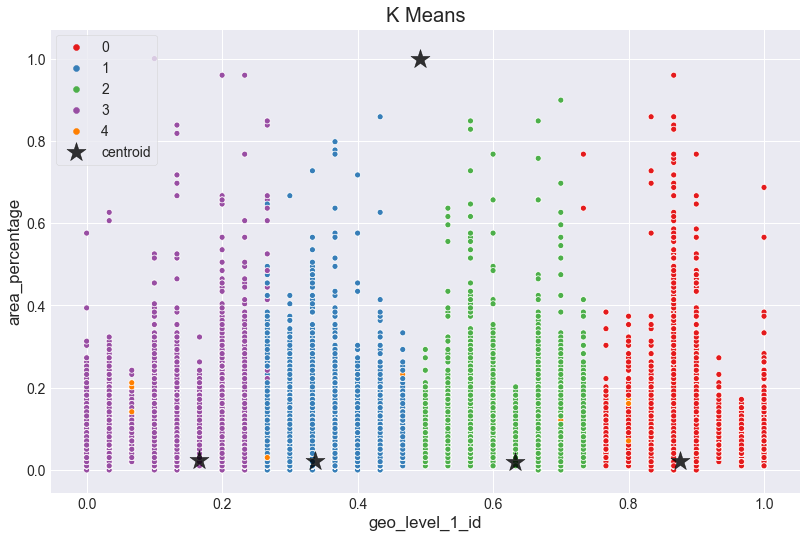




Using number of clusters as 5







**Earthquakes, 1965-2016**

**II. Significant Earthquakes from 1965 to 2016**

Date time and location of all the earthquakes with magnitude of 5.5 or higherThis dataset includes a record of the date, time, location, depth, magnitude, and source of every earthquake with a reported magnitude 5.5 or higher since 1965.

The data set was downloaded from

<https://www.kaggle.com/usgs/earthquake-database>

Jupyter notebook was used to run the program. Different necessaries libraries were installed to read and show the output. Each line were run to see the output.

Plotly was also installed in the anaconda file in this project

Pip install plotly

conda install -c plotly plotly

1. **READING THE DATA SET**

This is a large containing attribute like date, time, latitude, longitude and many different attributes. Most of the attributes here are not necessary we take Latitude, longitude, depth and magnitude into consideration while performing the clustering.

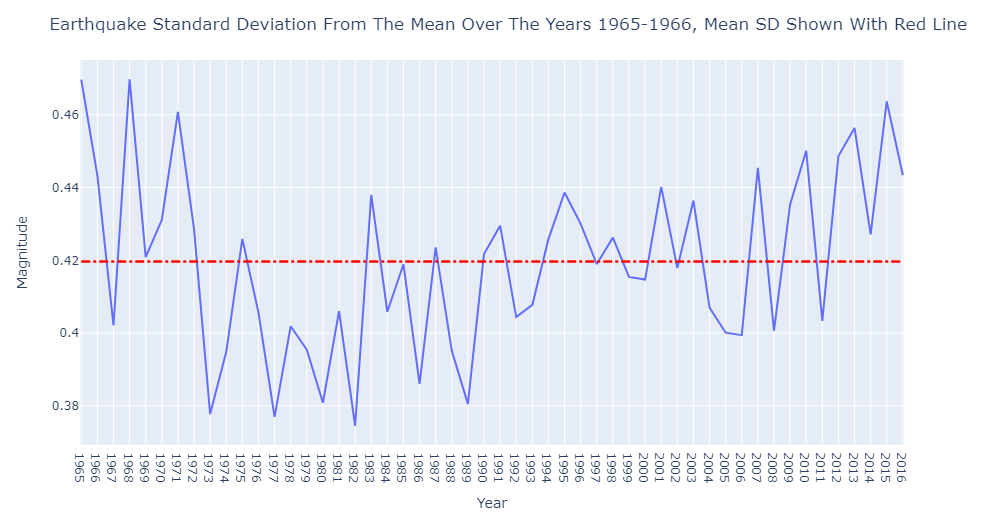
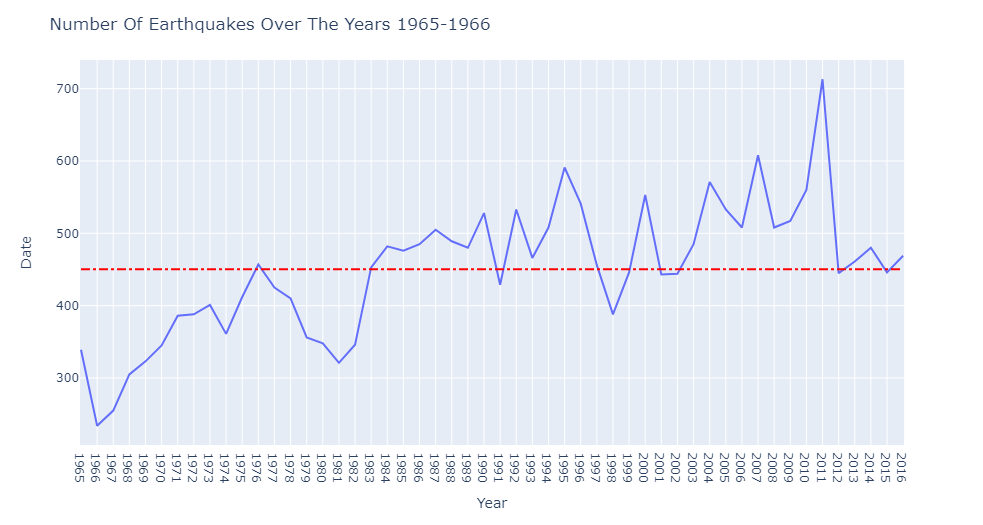
**2.DATA PREPROCESSING**

### There were missing data in the data set which was processed in the jupyter notebook. we then use the mode of the magnitude type feature to replace the small number of missing values, as for the other features which can be done by regression and nearest neighbor approach.

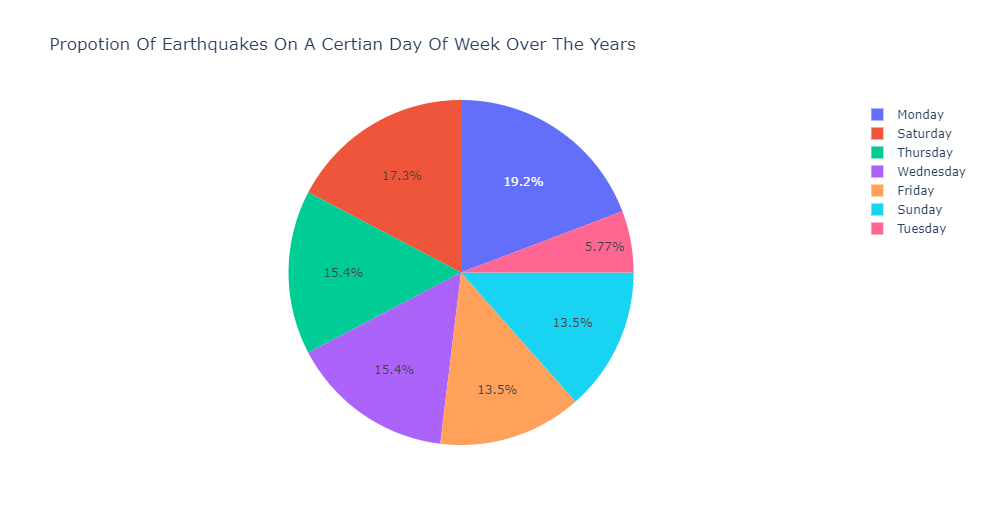
### We then perform the skewness of the data to see the positive skewness and negative skewness of the data. It helps us to see the direction and relative magnitude of a distribution's deviation from the normal distribution.

### 3.VISUALIZTION OF DATA

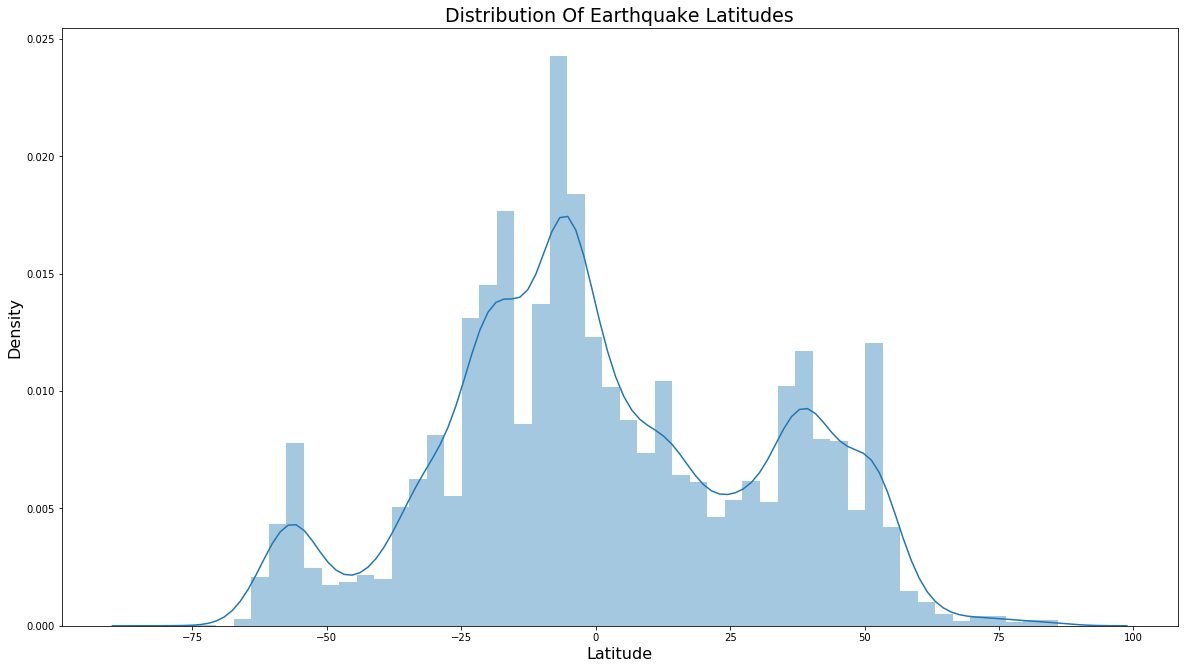
### Distibution Of Yearly Magnitude/Deapth Over Our Time Line For Each Partition

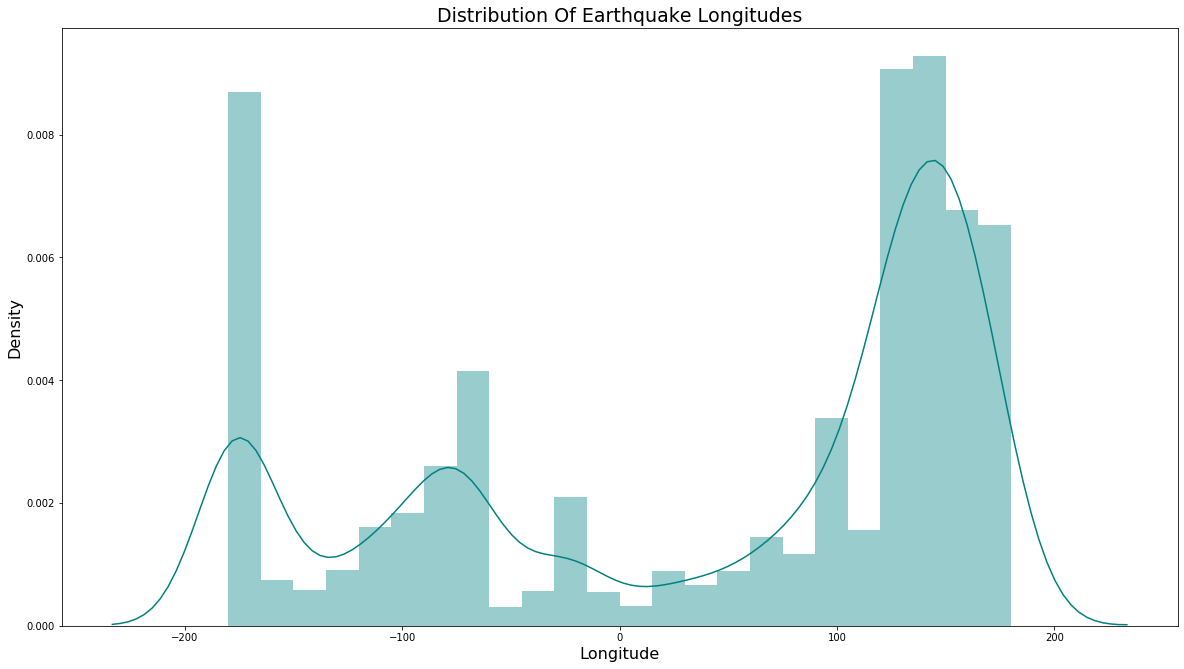


Plotting number of earth quake over the years . we see the level of magnitude have increased and decreased over the years and it shows growing trend from yeas of lowest magnitude record in 1982 and highest in 1965.



The pie chard shows the amount of earthquake among the days of the week which indicates equal amount distributed in the weeks.





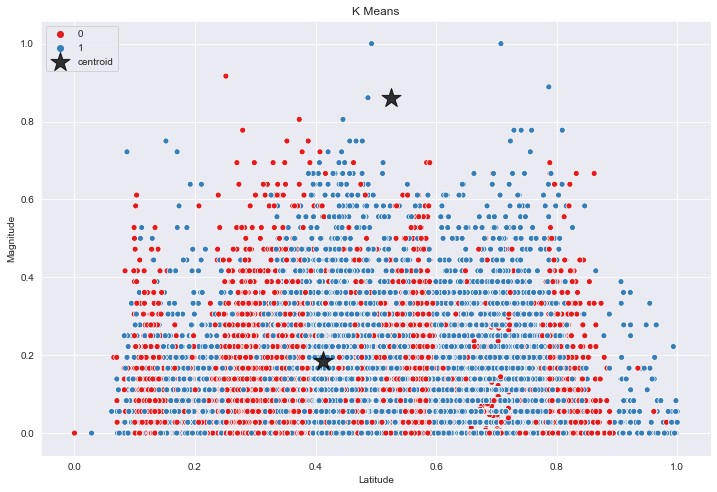
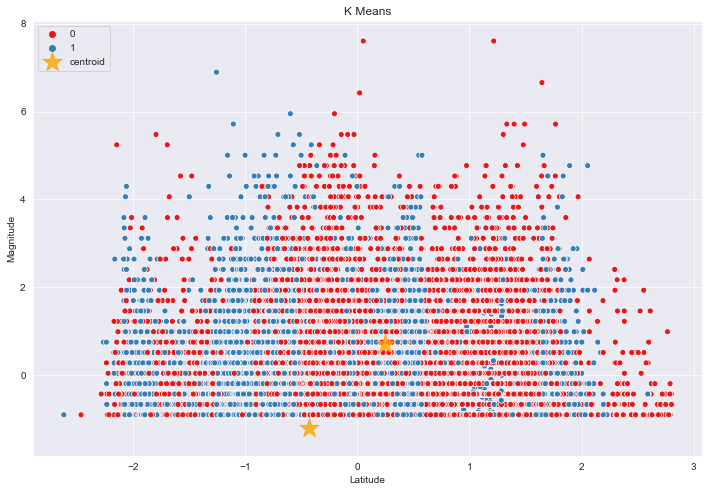
Distribution of earthquake in latitude and longitude.

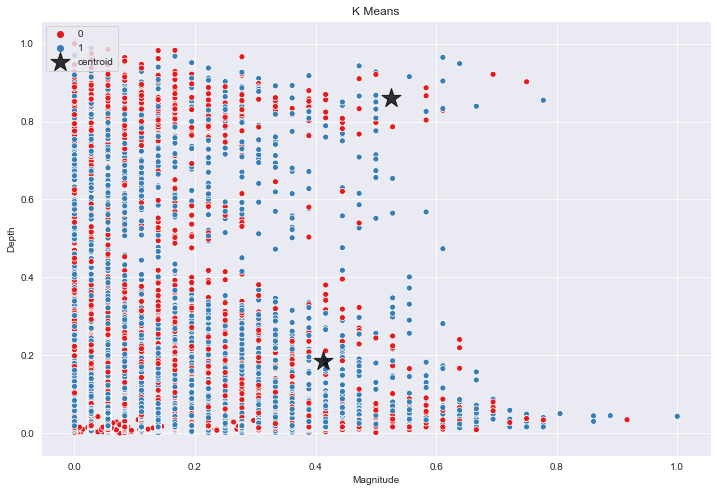
**4. Clustering techniques used**

K mean clustering and DB Scan clustering algorithm were used by the mean of sklearn libraries in jupyter notebook.

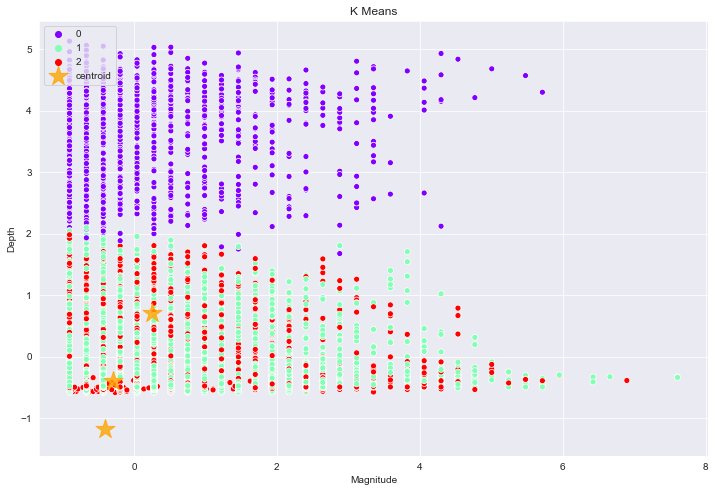
4 attributes, 'Latitude','Longitude','Depth','Magnitude' were chosen which was normalized .

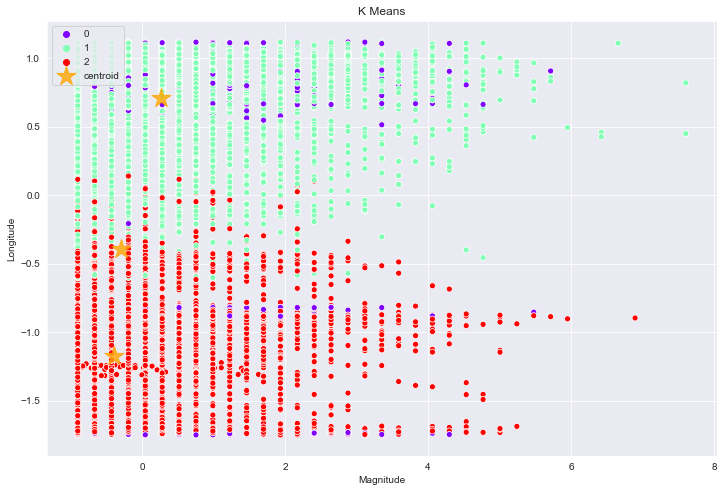
Using k as 2

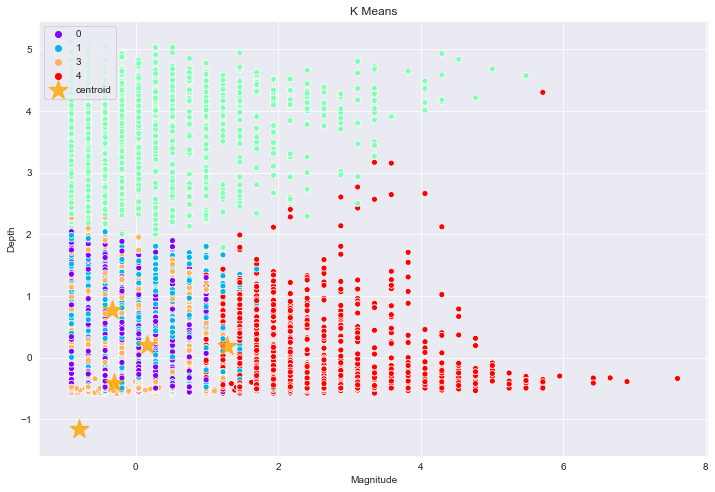
 



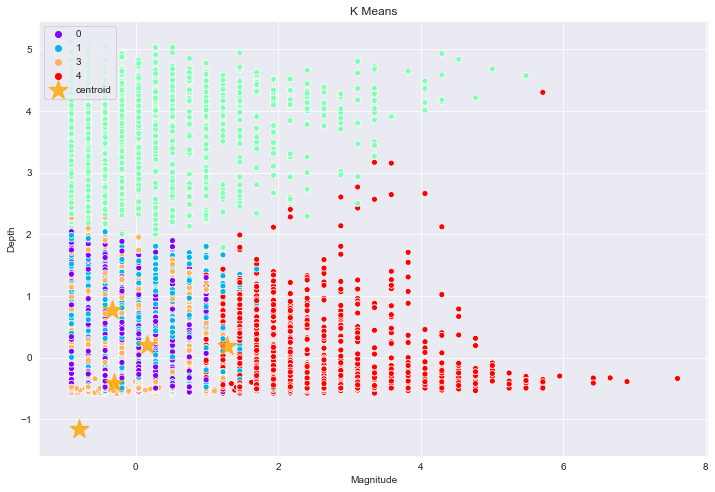
Using K as 3

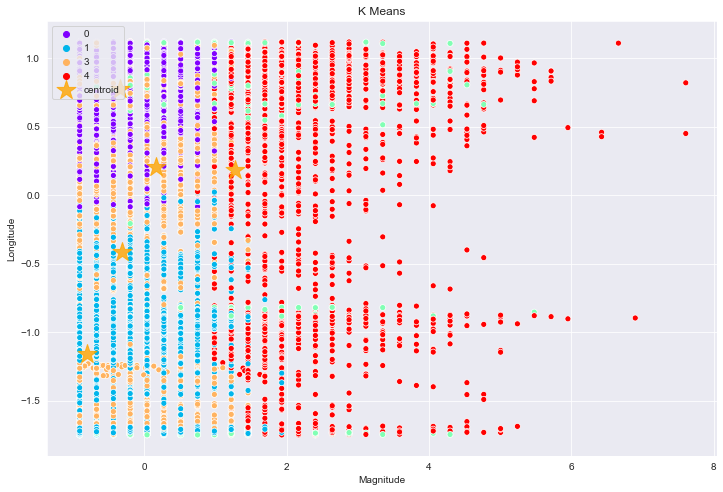






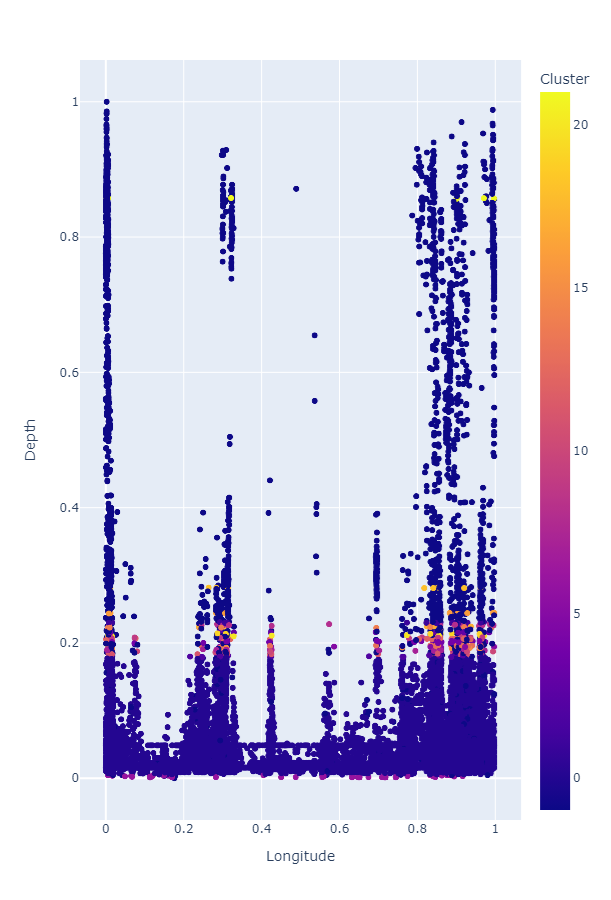
Using K as 5

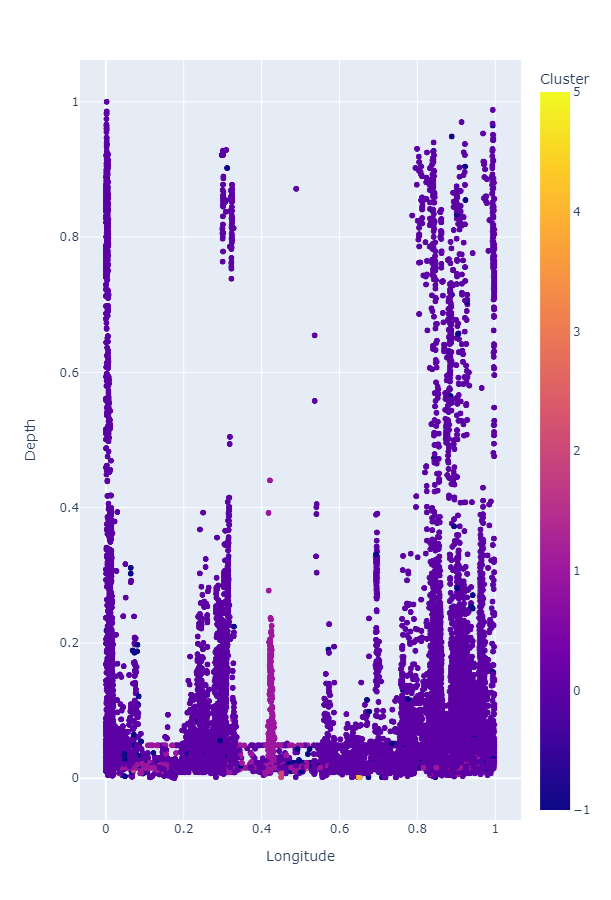


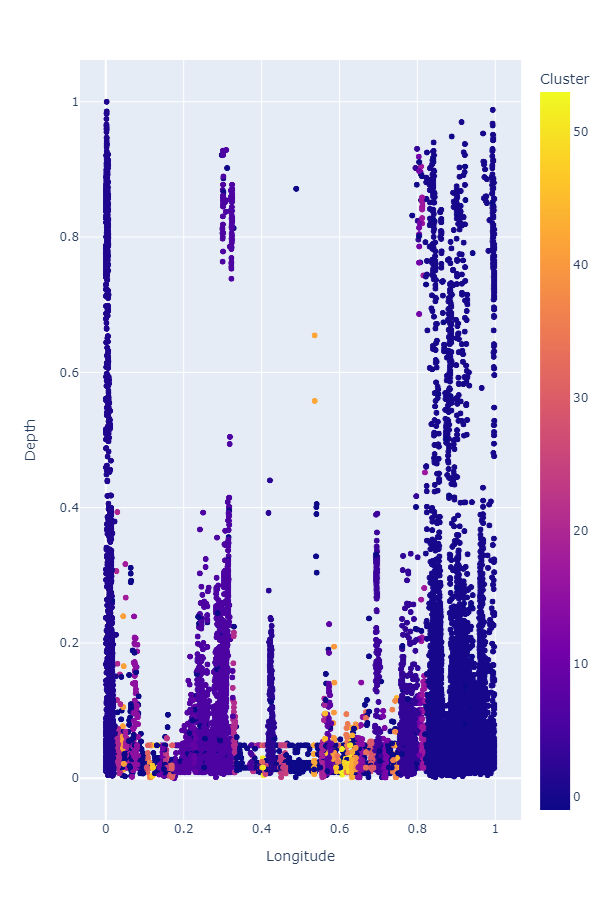


DBSCAN ALGORITHM

Using EPS as 0.5 and Min sample as 15







1. **WINE DATA SET**

Jupyter notebook was installed and necessary libraries like panda, numpy etc were also installed. Each cell was run to see the result

The data set was downloaded from <https://archive.ics.uci.edu/ml/datasets/wine>

The attributes are:

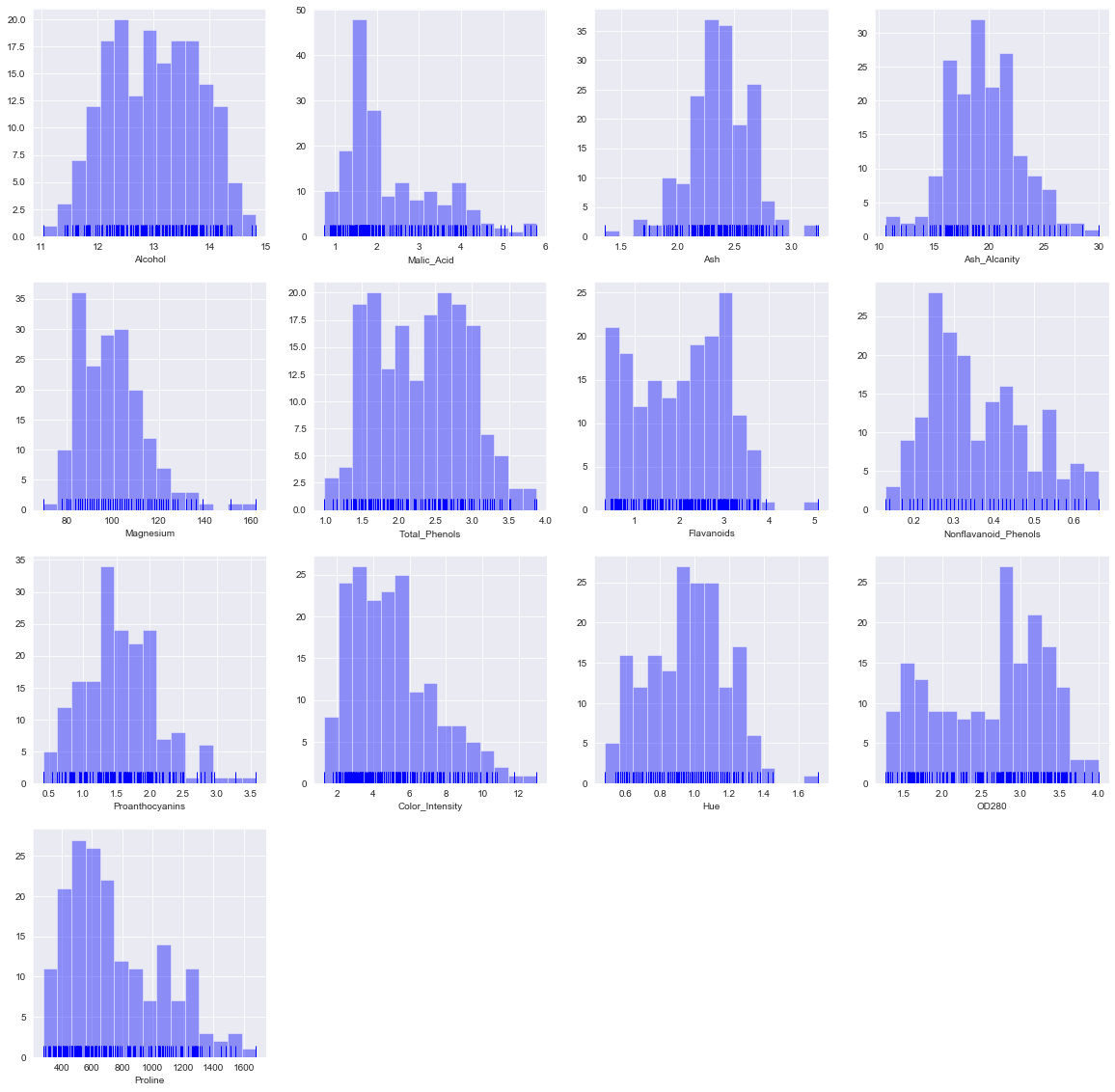
1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

1.DATA READING AND PREPROCESSING

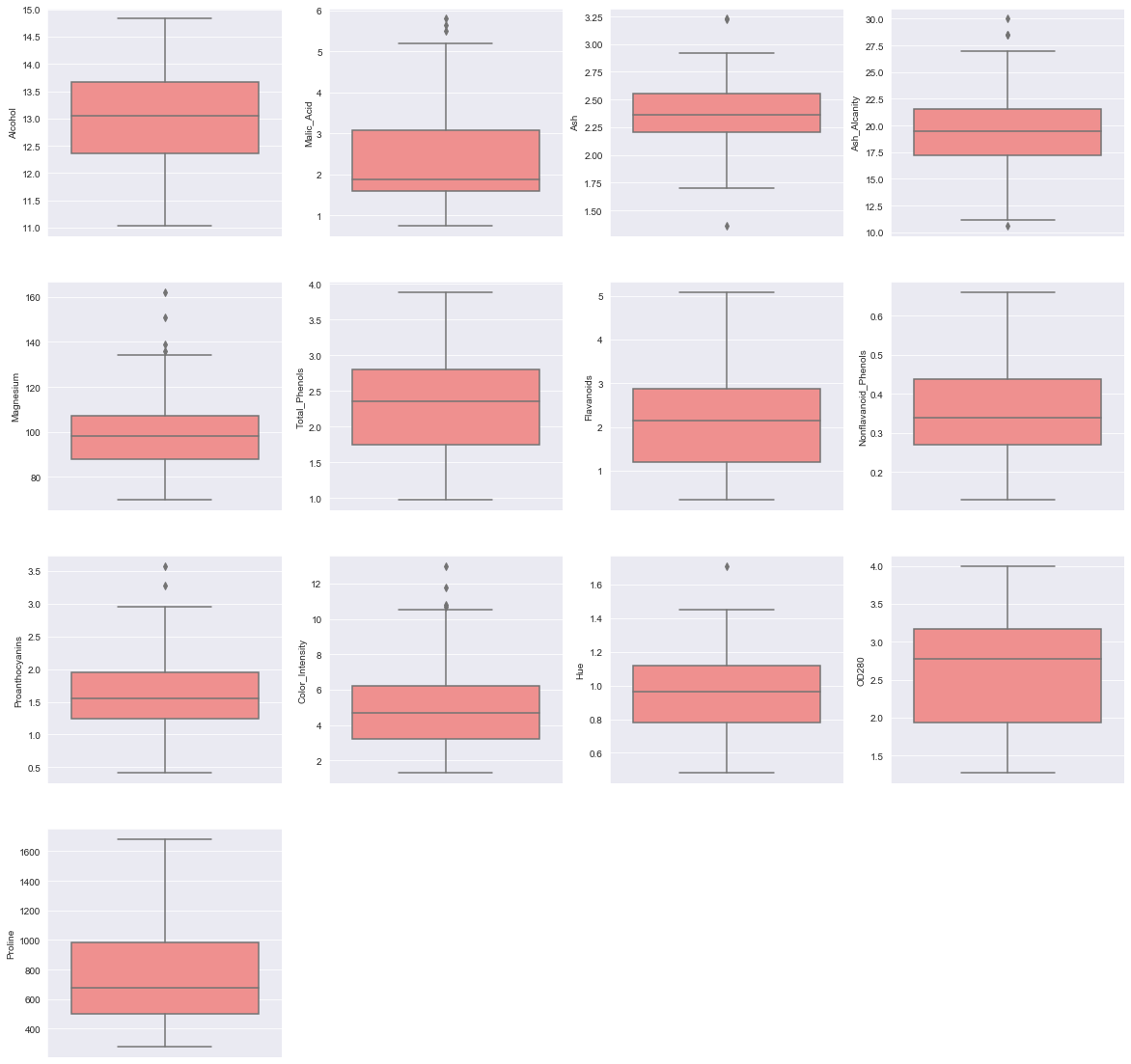
Removing the information about the types of wine for unsupervised learning. There were no object data type so there was no need to convert the attribute to integer. There were also no missing data in the data set.

2. DATA VISUALIZATION





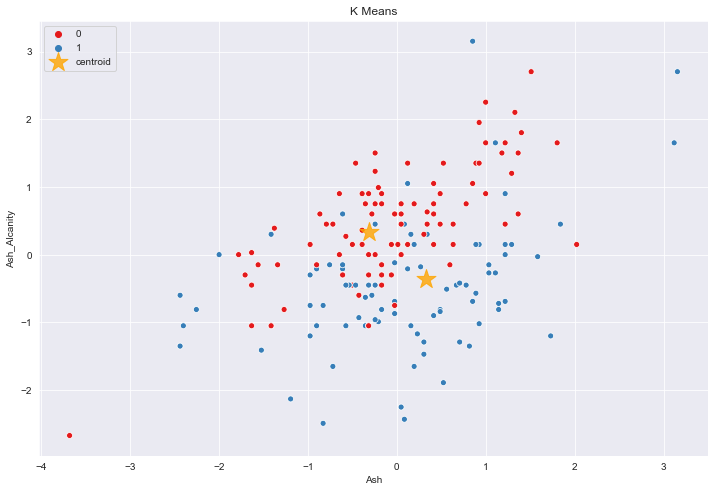
Linegraph of different attributes in the data set

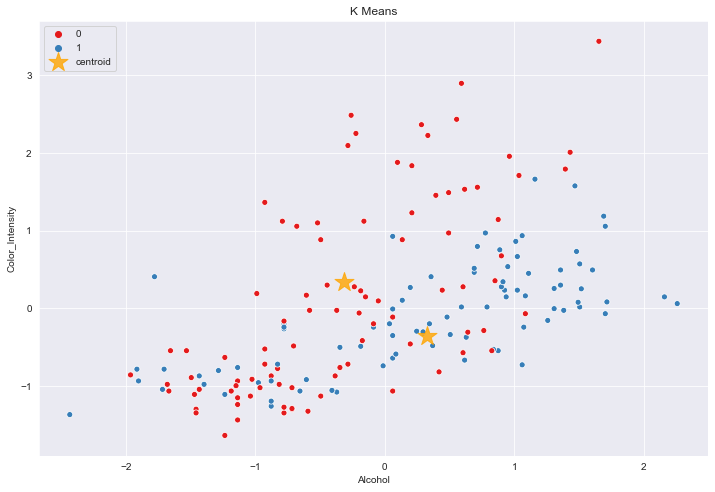


Box plot of different attributes.

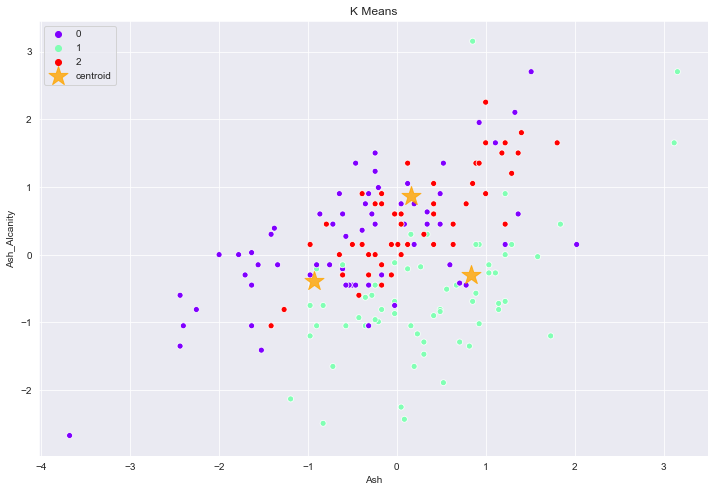
K MEAN CLUSTERING

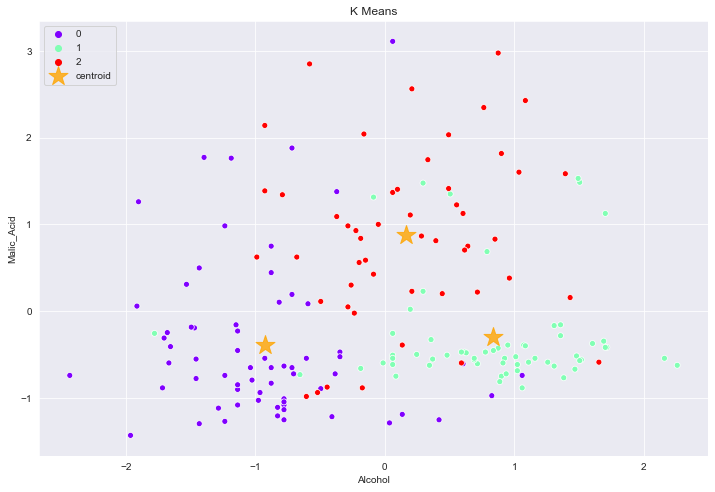
K AS 2



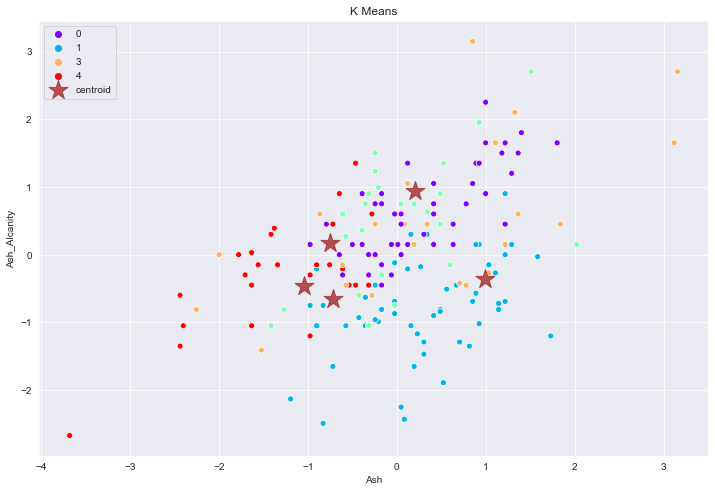


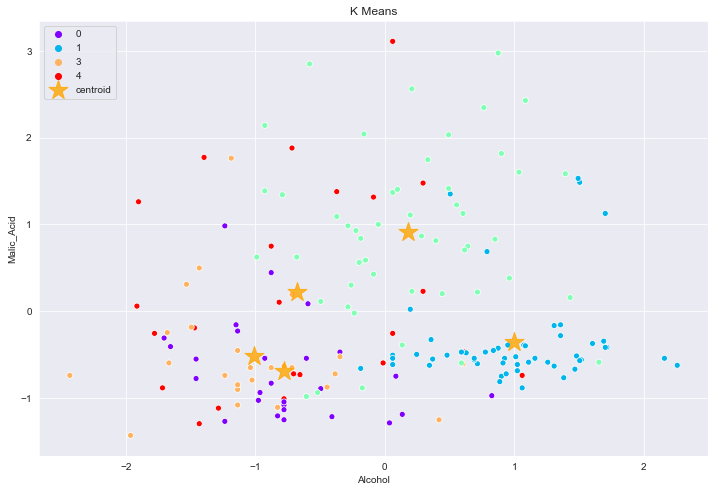
K as 3



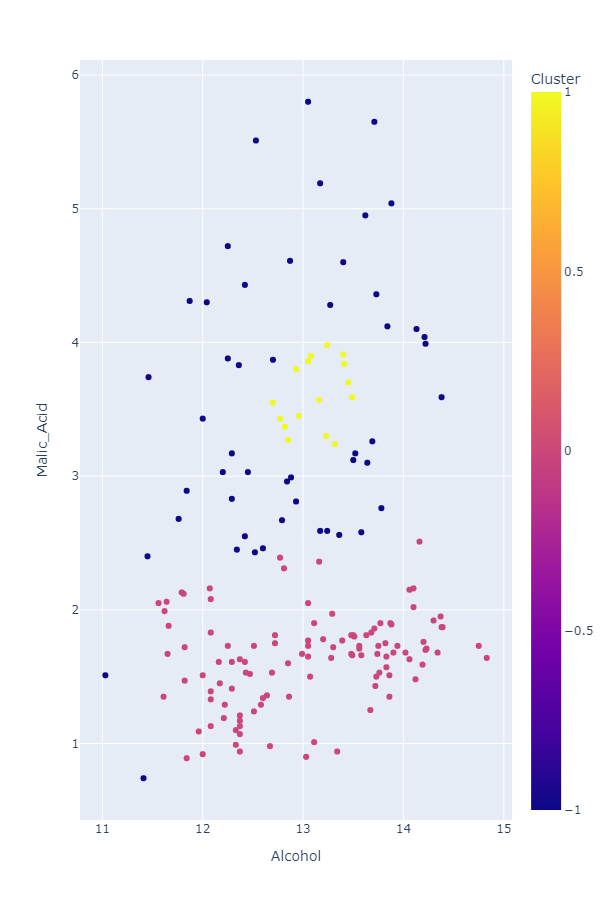


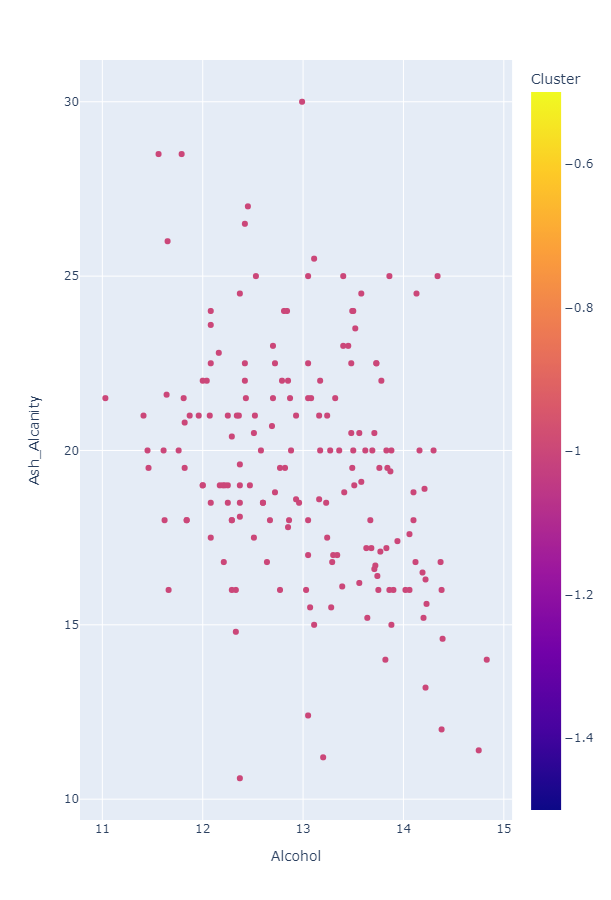
K as 5





DB SCAN WITH EPS 0.5





1. **CONCLUSION**

After comparing the data sets, the clustering graph of Wine data shape looks clearer and better compared to the earthquake data set**.** It is because of the lowers data shape and size. Also, the datasets of wine contained better attributes that would be compared easily, the datasets of earthquake were complex and the results that they showed were also very complex compared to the dataset of wine because of the huge amount of data. Furthermore, the DBSCAN graph of the earthquake in Nepal dataset got stuck and showed the memory error because of its complex nature.

On the other hand, The Visualization of earthquake were better compared with that of smaller data set due to the large amount of data.