Analysing earthquakes in United States to determine the possibly risky to quakes hotels

IBM APPLIED DATA SCIENCE CAPSTONE

Swastik Nath

29 April, 2019

Background:

Every year millions of people around the world, travellers, migrators, die in one of the most catastrophic natural disaster, Earthquakes. With massive and severe destruction of human-built architectures, skyscrapers, hotels and more, earthquakes cast the very curse of nature upon human. Instead of rapid advancement of Machine learning and neural networks there haven't been a prediction system that could meet our expectation and the required accuracy. But by harnessing the earthquake records over the places and analysing street level information we will be able to view which of the skyscrapers; hotels requires attention for upgrading or to equip them with anti-quake devices or earthquake resistant systems. With the help of powerful data science tools, it is possible to investigate this problem further.

Problem

It is the data that can help us in determining the earthquake heavy regions across USA and to determine certain hotels that requires upgrade to anti-quake devices and scrutiny for structural strength. This project aims to predict such hotels in earthquake heavy regions that requires upgrade with their respective vulnerability scale.

Interest:

Different government organizations such as city management department, geological surveyors and stakeholders at different hotels will be interested in getting to know about the status of their buildings against earthquakes and their vulnerability for the sake human lives. This assignment is aimed to determine the earthquake heavy regions in the United States and to find the vulnerable to earthquake hotels across the Unites States. This assignment will be helpful to the government city management departments and or, stakeholders of the hotels to determine if the place where they are constructing or have constructed hotels are safe from earthquakes and if not, it will trigger them to inspect for the structural security. AS, United States and its neighbourhoods and their vicinity are always ready for business and cultural programmes, it's obvious that there are so many hotels around the cities we choose. But we will use a different approach, called Vulnerable Occurrences (frequency of occurrence as a nearby hotel per recoded place) to determine the vulnerabilities in different hotels. In this project we will use different data science tools to determine the vulnerable places and hotels

near them. While applying the tools, we will follow the methodical framework as prescribed by Dr John Rollins of IBM.

Data acquisition and pre-processing:

2.1 Selecting Data Providers:

A. Fetching Earthquakes Records:

As the earthquake data provider, the United States Geological Survey (USGS) provided a very intuitive API and web portal service that supports different output formats and enables the users to search their database with many flexibilities such as specifying the region, timeframe and data limit both through the web client and the API. The dataset achieved was first downloaded and then loaded into a private cloud bucket for easier access. The dataset had missing values, which in case of numerical value, was replaced with their respective column mean and in case of categorical value was replaced with their most frequent occurrence. I obtained the data in a CSV format from here using the API and the web client.

B. Fetching Street Level Business Information:

As the street level business information provider including information about different hotels and skyscrapers, Foursquare API has been selected, due to its reliability and ease-of-use and affordability. For this project a Foursquare Developer Account was created and its credentials have been used throughout the project for requesting data from the Foursquare servers.

Data Cleaning:

Due to historical records, there were some missing data values, which I had to re-interpret and replace with some other values to make them significant. Such as there were missing information about number of stations used to record the earthquake event, angular distance of the recording place from the epicentre of the earthquake. So, these being numerical values, I replaced them with their respective column frequency. I discarded the rows with null magnitude values or specific location values. For categorical values such as 'Status', I converted them into numerical numbers using lambda function and encoded them with One-Hot-Encoding method. I discarded some specific columns due to their less importance such as 'Sources', 'ids' , 'id', 'code, 'URL', 'tz' for easier data management and analysis. I checked for incorrect datatypes per columns and converted the 'time' and 'updated' object to their perfect datatype of datetime64. Let's now look at, what every column of the dataset means:

Serial Number:	Column Header:	Reference by the column
01.	time	Time of the earthquake event
02.	latitude	Latitude of the recorded event
03.	longitude	Longitude of the recorded event
04.	depth	The depth where the earthquake begins to rupture

05.	mag	Magnitude of the event recorded.
06.	magType	The method or algorithm used to calculate the preferred magnitude for the event
07.	nst	The total number of seismic stations used to determine earthquake location
08.	gap	The largest azimuthal gap between azimuthally adjacent stations (in degrees)
09.	rms	The root-mean-square (RMS) travel time residual, in sec, using all weights
10.	id	A unique identifier for the event.
11.	updated	Time when the event was most recently updated.
12.	place	Textual description of named geographic region near to the event.
13.	type	Type of the incident recorded.
14.	horizontalError	Uncertainty of reported location of the event in kilometres
15.	depthError	Uncertainty of reported depth of the event in kilometres
16.	magError	Uncertainty of reported magnitude of the event
17.	magNst	The total number of seismic stations used to calculate the magnitude for this earthquake
18.	status	Status of the recorded event.
19.	dmin	Horizontal distance from the epicentre to the nearest station (in degrees)
20.	locationSource	The network that originally authored the reported location of this event
21.	magSource	The network that originally authored the reported magnitude of the event.

After the row and column fixation, I looked into the data for outliers and cleaned the outliers by dropping them. The nature and natural data are full of irregularities, due to their complex and inapprehensible algorithms. So, minimal deviations have been ignored through the rest of this project. There are the entries which were reviewed by human at the time of their registration into the system, so those data are quite accurate than the data automatically generated by the machine. As we proceed, we do not drop the machine generated data for sufficient samples.

Methodology and Analysis Framework.

In this assignment we sort out the places where we have found higher frequency of earthquakes in the United States. We will at first visualise all the significant earthquakes around the world and then we will narrow down to the United States only. Next, we will further narrow down our results to the places with higher frequency of earthquakes in the US by passing in the geocoded places of recorded earthquake places.

After we finish our fetching stage, we, further move on to descriptive statistics of the data frame. Before diving deeper, we replace the null values and drop the insignificant columns. We proceed with all steps to obtain a cleaned dataset with values all significant.

In the very next step, we proceed to exploratory data analysis where we explore different relationships between different between columns and explore the factors that have an impact upon the magnitude by plotting them differently and visualising them both in 2 and 3 dimensional plots, which will deliberately help us in better understanding the relation curve.

The very next step of us, is to fetch street level details to obtain information about different businesses, especially hotels near the risky to earthquake places by requesting through an API. Next, we generate a data frame from the response of the API to better represent the resulting dataset, narrowed down to hotels only, we plot them to a map and visualise circular boundaries around them.

We introduce a new terminology here, Vulnerable occurrence, which is in simple words the frequency of occurrence as a nearby place of a single business place (here hotels) per place of recorded earthquake event. We count Vulnerable Occurrence of per business places by simply counting them.

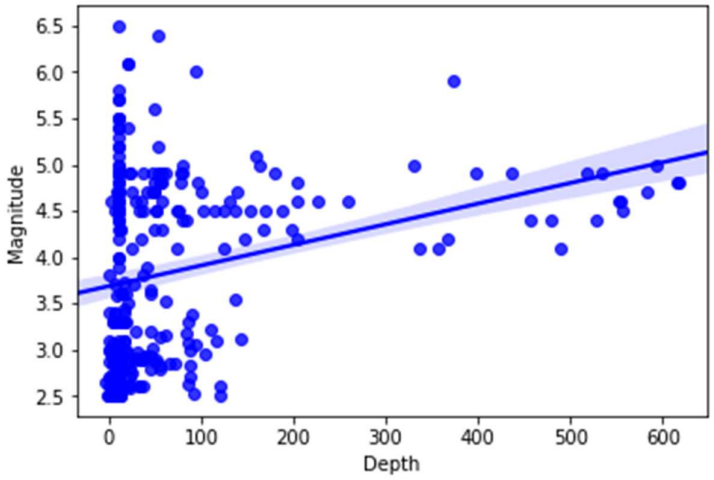
We, proceed further and cluster the hotels using K-Means Clustering, a supervised learning process where, the algorithm clusters the unlabelled dataset into pre-specified number of clusters and assigns them a label. We plot them to our map for visualisation and evaluation.

Visualisation:

1. Scatter Plot between Depth and Magnitude

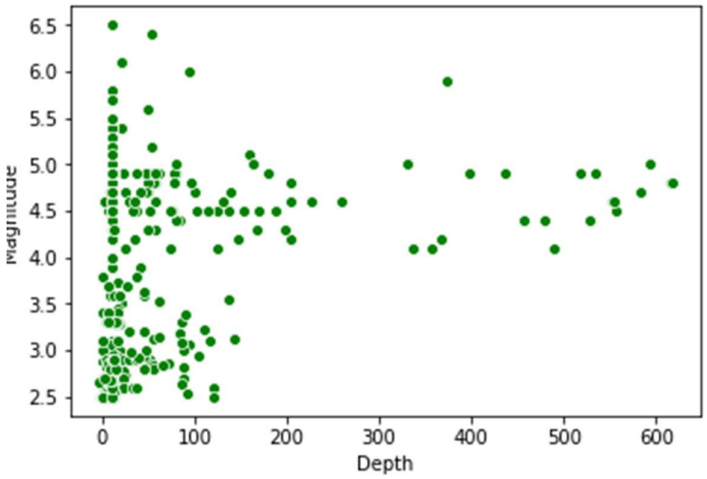
The Scatter plot between depth and magnitude does not yield any kind of relationship among the two perfectly. But we can analyse a linear data between the two. As the regression line shown in the illustration. However, the regression line is showing deviation in both sides of it.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd47c9b0>
```



2. Regression plot between Depth and Magnitude:

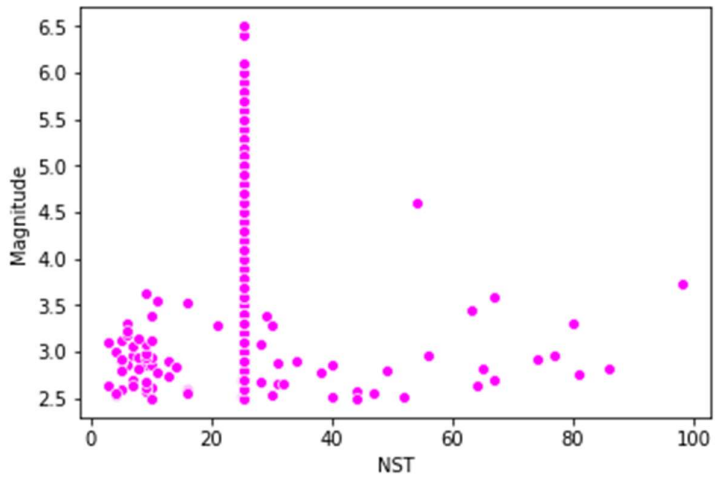
The regression plot between the depth and the magnitude values shows a linear relationship between the two, instead of showing deviations on the both sides.



3. Scatter Plot between Number of stations required to measure the event versus magnitude

As we can see, there is certainly no relationship between these two variables, however we can see a linear plot at a certain point along the NST axis. But that is of course of no use.

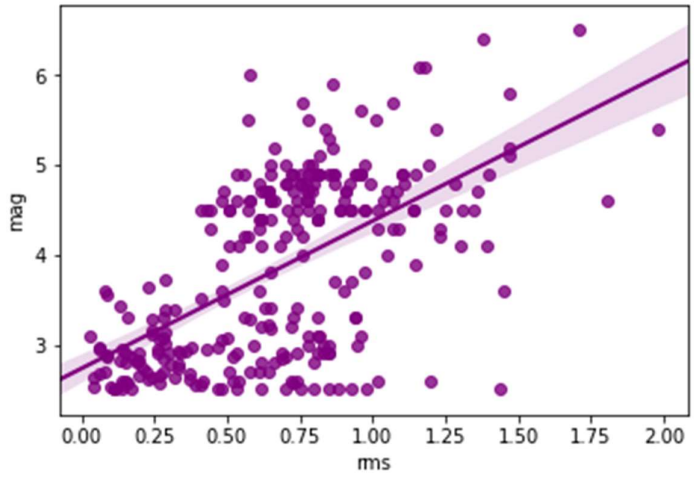
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd416860>
```



4. Regression plot between the Root Mean square speed of the waves and the magnitude.

The plot shows a linear regression among the two with certain deviations. But we can still ignore the irregularities due to the nature. Due to the not so significant amount of deviation along the regression line proves the linearity of the relationship.

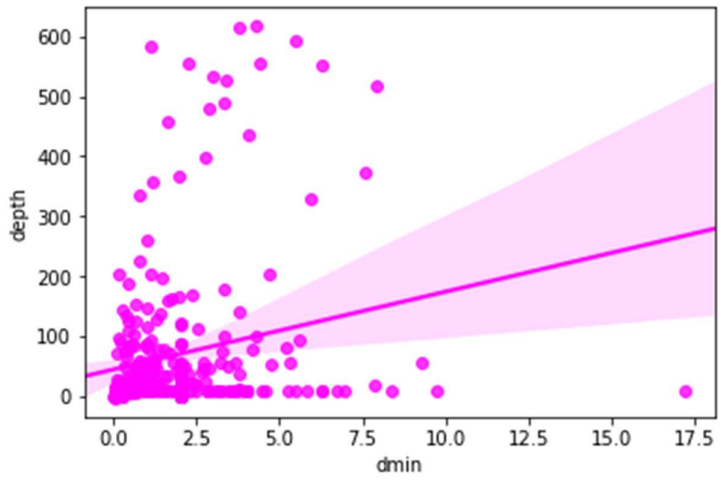
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd40a550>
```



5. Regression plot between the horizontal distance of the recorded place with the depth of the rupture of the waves.

The regression plot shows a large amount of deviation meaning that the relationship among the two is not linear.

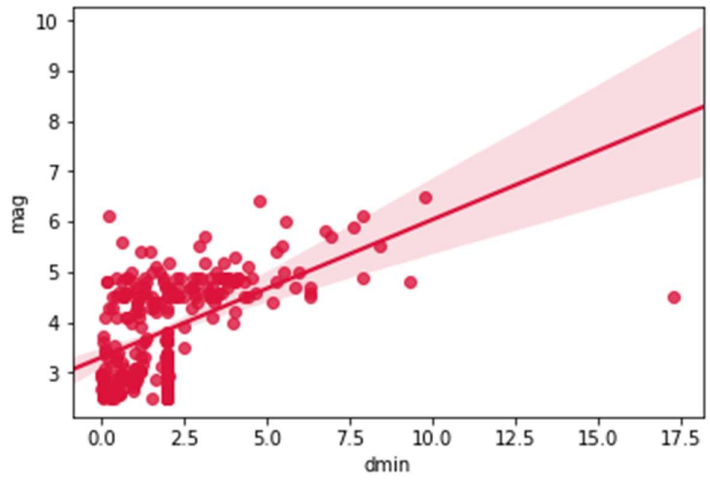
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd34e1d0>
```



6. Regression plot between the horizontal distance and the magnitude:

The regression plot between the horizontal distance from the epicentre and the magnitude shows a linear regression plot but with higher deviations, basically implying non-linearity between the two variables.

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd331240>
```

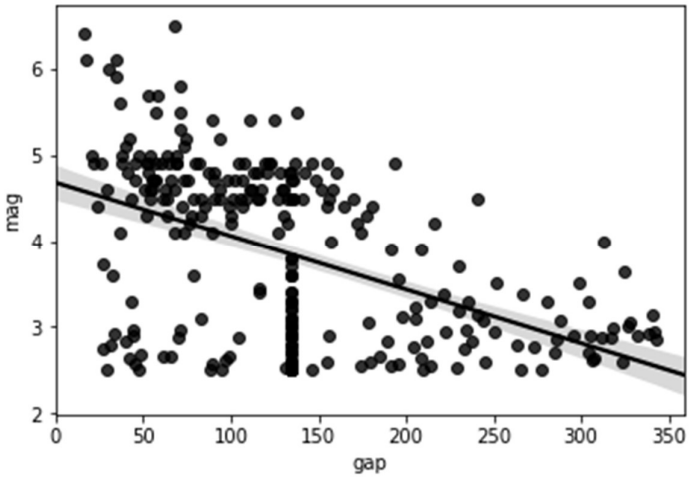


7. Regression plot between azimuthal gap versus magnitude:

The regression plot shows a negative linear regression with a minimal deviation simply stating the relationship between the two a negative linear regression. Despite the deviation in the plot, the linearity is noticeable in the plot.

On the other hand, the negative regression signifies the positive linear regression relationship between magnitude and the azimuthal gap.

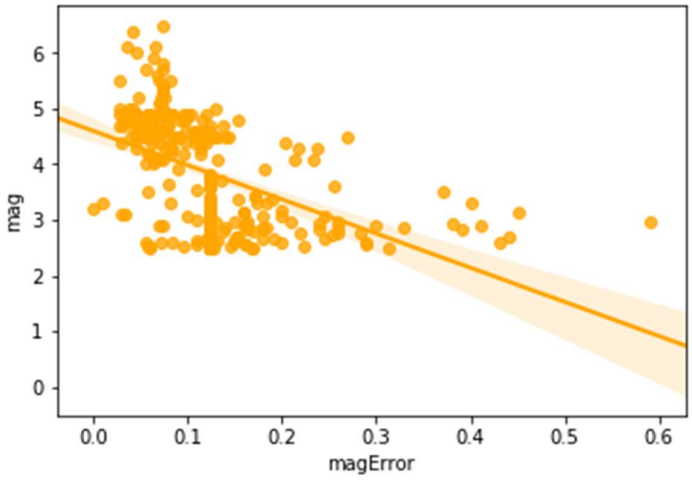
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd27f588>
```



8. Regression Plot between error in measuring magnitude

This plot shows a negative linear regression with a minimal deviation, referring that, if the error in measuring the magnitude increases, the magnitude decreases. This is true and just basically verifying the data in the column.

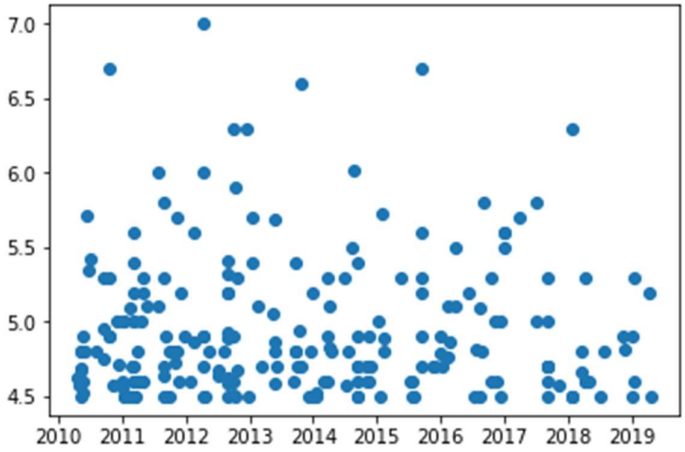
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34bd209898>
```



Plotting Magnitudes in a timeframe:

We use the scripting layer of Matplotlib to generate a timeframe of the magnitudes recorded in the dataset. We change the datetime (64) object into int object by using the matplotlib's dates.date2num function. Next, we use the scripting layer and pass in the required arguments.

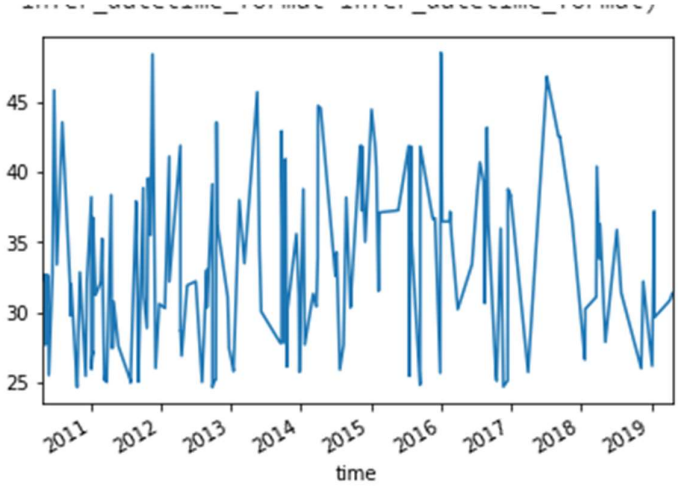
```
[<matplotlib.lines.Line2D at 0x7f34bce6e550>]
```



Generating the series plot for a better time-frame visualisation.

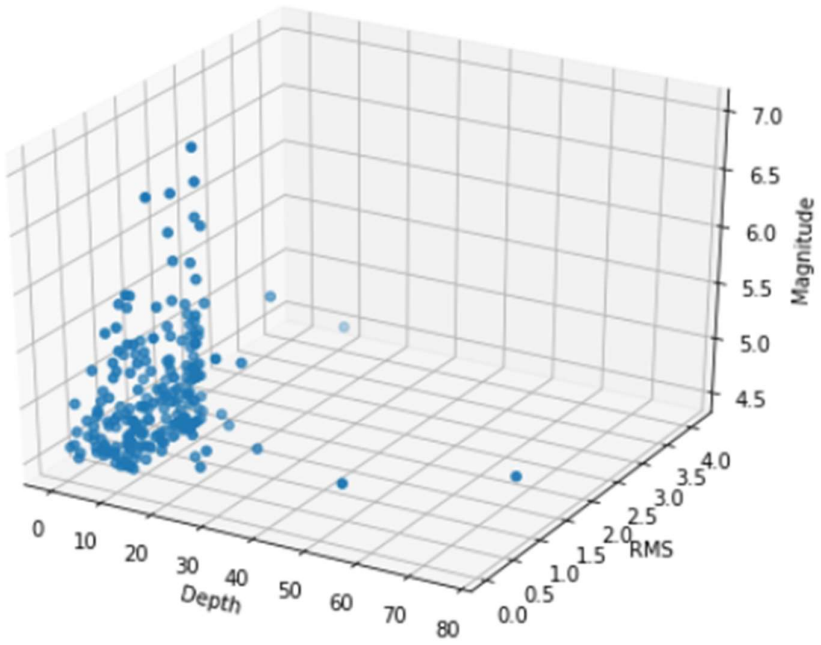
We directly pass in the CSV file containing the dataset to pandas form a Series object and matplotlib is intelligent enough to show the continuous flow of magnitude in a lower scale with respect to time automatically converted to an int object.

In the following plot, we get a continuous plot of the magnitudes with respect to time.



3-Dimensional Visualisation of Depth vs. Root Mean Square Speed vs. Magnitude:

Here we plot Depth in the X-Axis, Root Mean Square Speed in the Y-Axis and Magnitude of the recorded event in the Z-Axis to obtain a 3-dimensional plot for better understanding the relationship between the variables.



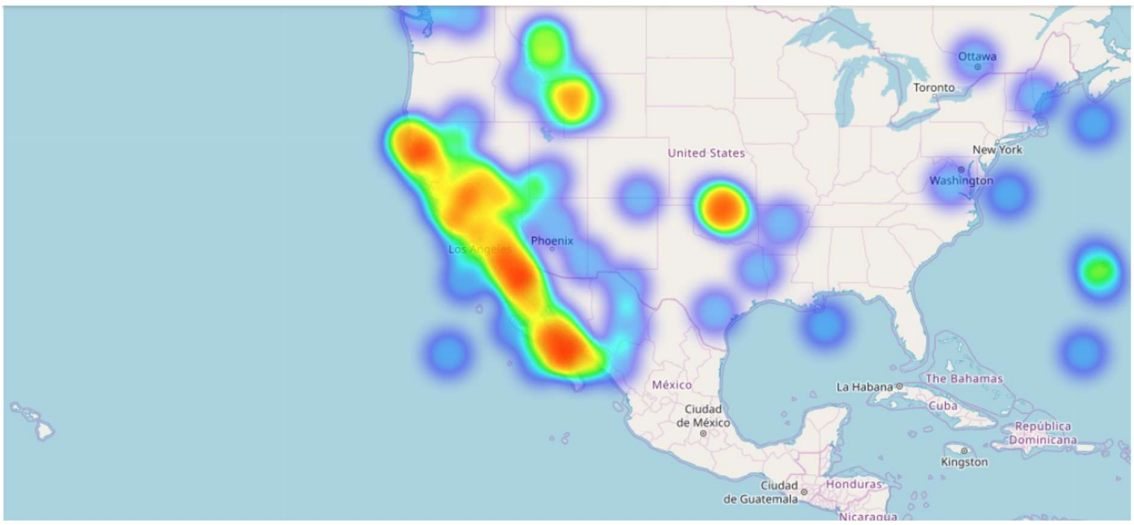
Visualisation of Earthquakes across USA:

Let's visualise the earthquakes from the dataset in a map to get a real-world look at the places mostly affected. To de-clutter the map, I have used FastMarkerCluster to cluster the markers and unfold them with respect to the zooming

Creating a Heatmap of the Earthquake Affected areas:

We use the Heatmap function imported from the folium.plugins package to generate the heatmap.

We create a heat map to understand the frequency of the affected areas across USA. We see that, the map is heated mainly in the pacific coastline of USA.



Let's Get the places with highest earthquakes count for the period of 2010-2019 in USA.

Let's print the top earthquake heavy places across USA and create a data frame representing the places and the earthquake count in that specific place.

The following horizontal bar plot shows the frequency of earthquake counts with respective to their places.

In the next, we will use the following places and their latitude and longitudes to find out the hotels that requires attention for implementing earthquake safety products.

Here we use the previously created helper function to fetch the JSON response from the USGS servers and create a data frame to display the results.

We also, clean the dataset by dropping null values and resetting the index values.

Connecting to Foursquare for obtaining the Hotel Details:

We will use Foursquare API to search for the hotels nearby the recoded earthquake locations. We will pass in the recorded latitude and longitude of the places to the Foursquare places API to fetch the results in a JSON format. We will then format the response into a pandas data frame for easy management. Let's create a custom helper function to obtain the JSON response from the Foursquare API and create a pandas data frame with the selected responses.

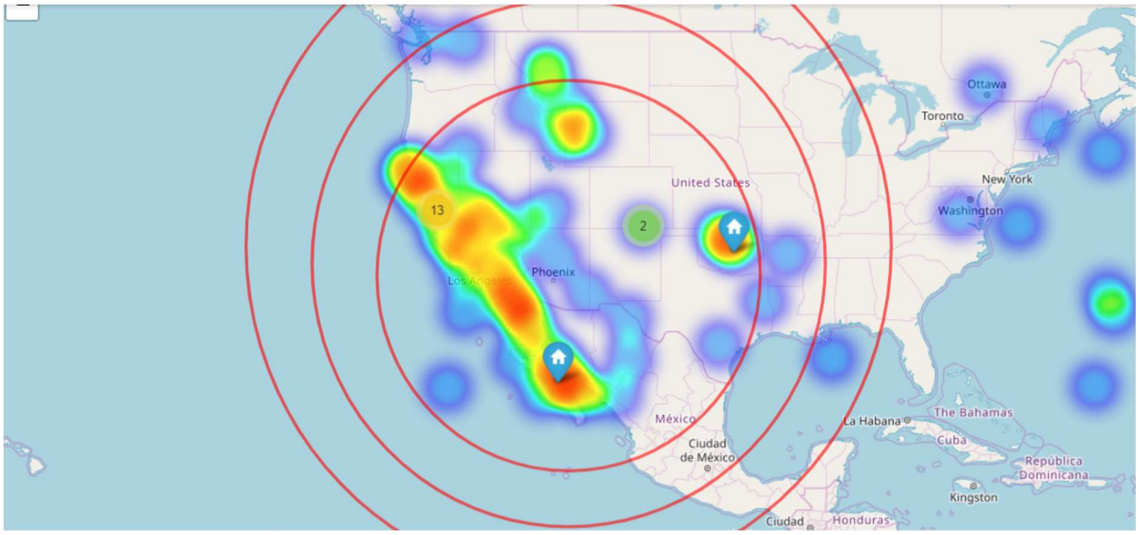
What is Vulnerable occurrence?

Here we measure which hotels are nearby to the recorded places. Vulnerable Occurrence is the number of occurrences of each recorded place as a nearby place per hotel. Simply stating, it's the measurement, how frequently each hotel appears as a nearby hotel per recoded place based on its latitude and longitude, within a specified radius of interest.

The higher the number of vulnerable occurrences, the higher is the need of inspection in the specified hotel for risk factors to earthquakes.

Adding the potentially risky hotels into the Heatmap

We plot each of the potentially risky candidates to our heatmap to better visualise their location and the effectiveness of our analysis. We have used FastMarkerCluster to visualise the vulnerable hotels in the map.



Cluster the potentially risky candidates

We use the KMeans Clustering to cluster the hotels we received as a response from the Foursquare API. As clustering is an unsupervised learning, let's drop the labels from the dataset and pass in the unlabelled dataset as a parameter into the KMeans clustering algorithm. We will cluster the hotels into five different clusters and plot them into a map.

Conclusion

The assignment was intended for determining the potentially risky places to earthquakes and finding the potentially vulnerable hotels around the places of most frequent earthquakes. With different data science tools ranging from data validation to visualisation helped in creating the assignment.

The clusters will help in segmenting the hotels with similar attributes in the map for the sake of better understanding the potential risk factors.

The final decision about the scrutiny in the hotels for earthquake safety and upgrade is completely upon the government city management team, the hotel stakeholders and surveyors.