Applied Machine Learning

Earthquake Magnitude Prediction

Name: Ajinkya Agnihotri

NetId: aaa210016

The recent spate of earthquakes has served as an eye-opener for us, which is why I decided to develop a data science project aimed at predicting earthquake magnitudes to help governments, emergency responders, and insurance companies minimize the impact on lives and infrastructure.

The earthquakes with higher magnitudes release significantly more energy than those with lower magnitudes. For instance, a magnitude 8 earthquake can release up to 1,000 times more energy than a magnitude 6 earthquake. Therefore, it is crucial to identify which regions are more susceptible to earthquakes of specific magnitudes. By doing so, we can improve our ability to predict the likelihood and potential impact of future earthquakes, allowing for better emergency response planning and risk mitigation strategies.

The dataset that I have chosen to work with contains information on earthquakes that have occurred between 1965 and 2016, with a magnitude above 5.5 on the Richter scale.

```
In [6]:  1 or_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23412 entries, 0 to 23411
Data columns (total 24 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Date                        23412 non-null  object
 1   Time                        23412 non-null  object
 2   Latitude                    23412 non-null  float64
 3   Longitude                   23412 non-null  float64
 4   Type                        23412 non-null  object
 5   Depth                       23412 non-null  float64
 6   Depth Error                 4461 non-null   float64
 7   Depth Seismic Stations      7097 non-null   float64
 8   Magnitude                   23412 non-null  float64
 9   Magnitude Type              23409 non-null  object
 10  Magnitude Error             327 non-null    float64
 11  Magnitude Seismic Stations  2564 non-null   float64
 12  Azimuthal Gap               7299 non-null   float64
 13  Horizontal Distance         1604 non-null   float64
 14  Horizontal Error            1156 non-null   float64
 15  Root Mean Square            17352 non-null  float64
 16  ID                          23412 non-null  object
 17  Source                      23412 non-null  object
 18  Location Source             23412 non-null  object
 19  Magnitude Source            23412 non-null  object
 20  Status                      23412 non-null  object
 21  date                        23412 non-null  object
 22  month                       23412 non-null  int64
 23  year                        23412 non-null  int64
dtypes: float64(12), int64(2), object(10)
memory usage: 4.3+ MB
```

These are the list of columns in the dataset:

['Date', 'Time', 'Latitude', 'Longitude', 'Type', 'Depth', 'Depth Error', 'Depth Seismic Stations', 'Magnitude', 'Magnitude Type', 'Magnitude Error', 'Magnitude Seismic Stations', 'Azimuthal Gap', 'Horizontal Distance', 'Horizontal Error', 'Root Mean Square', 'ID', 'Source', 'Location Source', 'Magnitude Source', 'Status']

- Target variable:

Magnitude: The magnitude of the earthquake on the Richter scale.

- Explanatory variables:

Date: The date and time (UTC) when the earthquake occurred.

Time: The time (UTC) of day when the earthquake occurred.

Latitude: The latitude coordinate of the earthquake's epicenter.

Longitude: The longitude coordinate of the earthquake's epicenter.

Type: Type of event that caused it like earthquake, rocket brust, nuclear explosion, etc.

Depth: The depth (in kilometers) of the earthquake's focus below the earth's surface.

Depth Error: The estimated error (in kilometers) in the depth measurement.

Depth Seismic Stations: The number of seismic stations that contributed to the depth calculations.

Magnitude Type: The type of magnitudes like MW, MWC, MB etc.

Magnitude Error: The estimated error in the magnitude measurement.

Magnitude Seismic Stations: No. of seismic stations that contributed to the magnitude calculation.

Azimuthal Gap: Azimuthal gap (in degrees) between the closest stations recording the earthquake.

Horizontal Distance: Horizontal distance (in degrees) from epicenter to the nearest seismic station.

Horizontal Error: The estimated error (in kilometers) in the horizontal distance measurement.

Root Mean Square: The root mean square (RMS) travel time residual for the earthquake, which measures the quality of the seismic data.

ID: A unique identifier assigned to each earthquake event.

Source: The organization responsible for providing the earthquake data.

Location Source: The organization responsible for providing the location data.

Magnitude Source: The organization responsible for providing the magnitude data.

Status: Indicates whether the earthquake event has been reviewed or is preliminary (automatic).

To get more information out of the data I extracted ‘month’ and ‘year’ from the date.

In [4]:
```
def convert_date(Date):
    return pd.to_datetime(Date).strftime('%m/%d/%Y')

or_df['date'] = or_df['Date'].apply(convert_date)
```

In [5]:
```
or_df['month'] = pd.to_datetime(or_df['date']).dt.month
or_df['year'] = pd.to_datetime(or_df['date']).dt.year
```

Then I used an API ‘**rg.search**’ to find the country names based on the Latitude and Longitude data. Not all the Latitude and Longitude values are registered in the python libraries, so this API searches the nearest known values and returns the corresponding country name. I have run this part which took around 10Hrs to provide the output for around 23000 rows. Directly using that output in the next cell.

In [297]:
```
#Run time 10 Hrs so directly using the o/p in the next cell

# for i in range(len(df)):
#     coordinates = (df.Latitude[i],df.Longitude[i])
#     a= rg.search(coordinates)
#     df['Impacted Country'] = a[0]['cc']
#     print(i, a[0]['cc'])
```

In [298]:
```
IC_df = df.copy()
countries = ['MP', 'ID', 'TO', 'GS', 'PH', 'VU', 'IN', 'VU', 'GS', 'FJ', 'ID', 'ID', 'RU', 'TO', 'TJ', 'AU', 'RU', 'RU', 'RU
IC_df.loc[:, 'Impacted Country'] = countries
IC_df.head()
```

Out[298]:

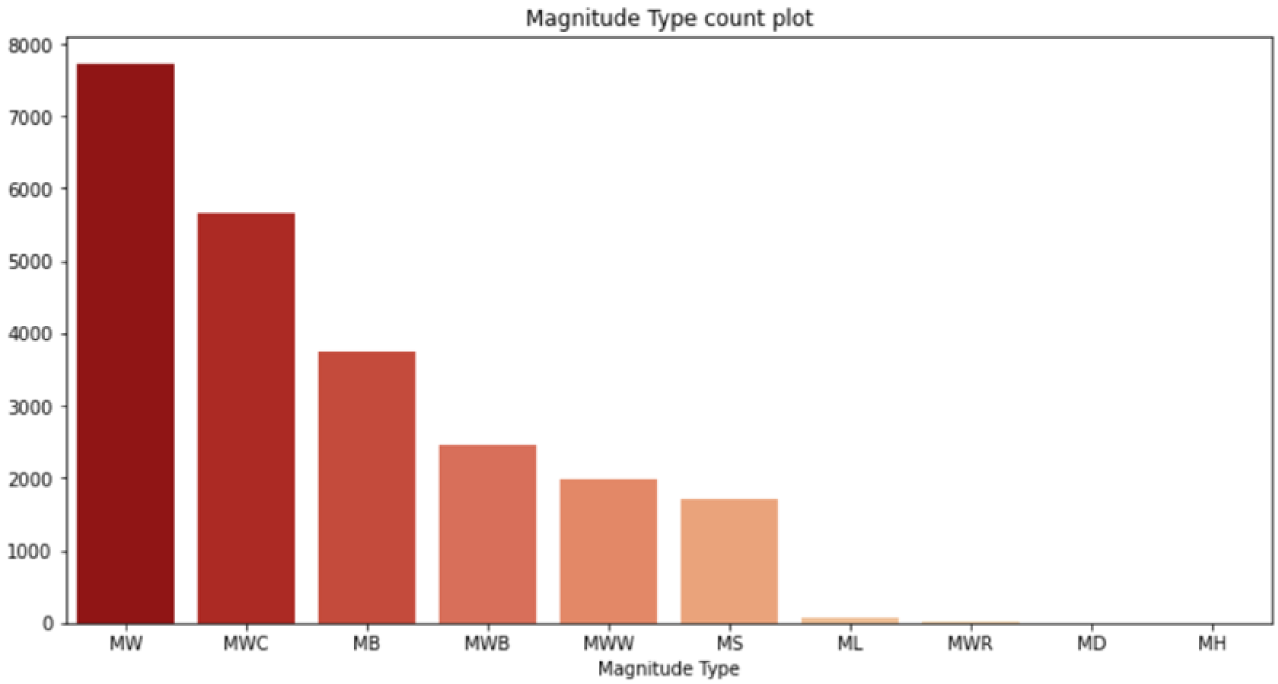
	month	year	Latitude	Longitude	Depth	Magnitude	Impacted Country
0	1	1965	19.246	145.616	131.6	6.0	MP
1	1	1965	1.863	127.352	80.0	5.8	ID
2	1	1965	-20.579	-173.972	20.0	6.2	TO
3	1	1965	-59.076	-23.557	15.0	5.8	GS
4	1	1965	11.938	126.427	15.0	5.8	PH

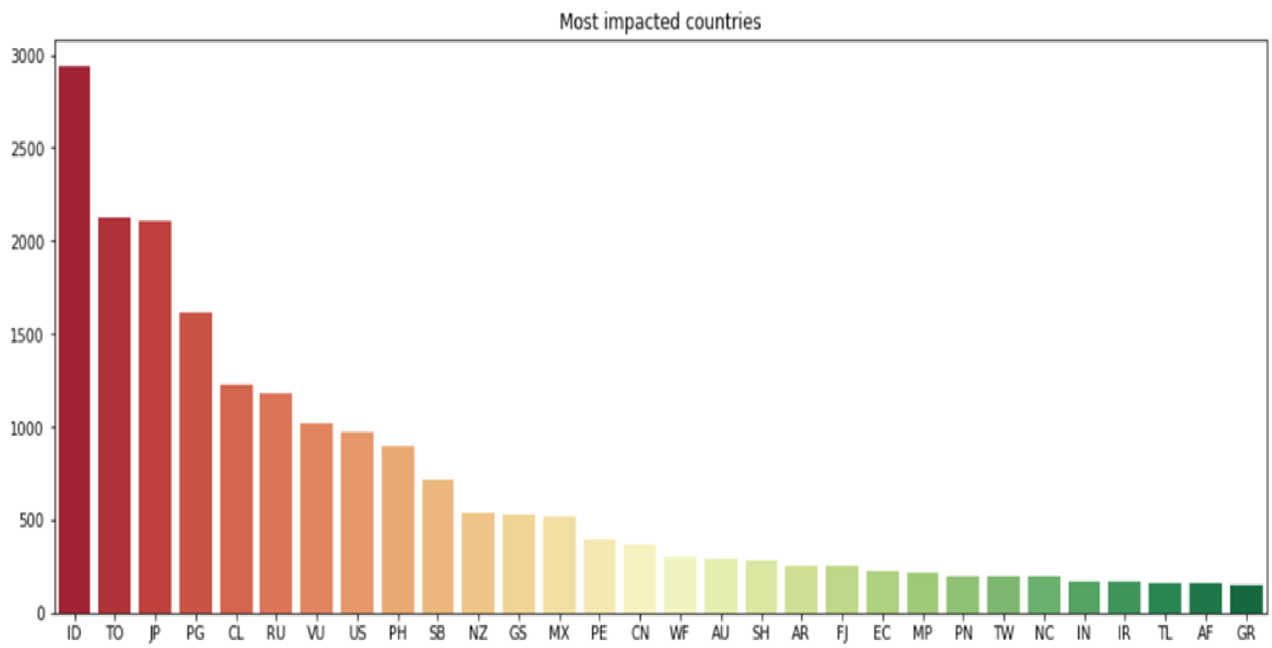
After extracting the required information, I started the data cleaning process. I tried figuring out why the data is missing and if it was random or not via internet but couldn’t find anything. Filling those values in some of the ways doesn’t seem right so decided to drop the null values. Here is the snapshot of the data after cleaning:

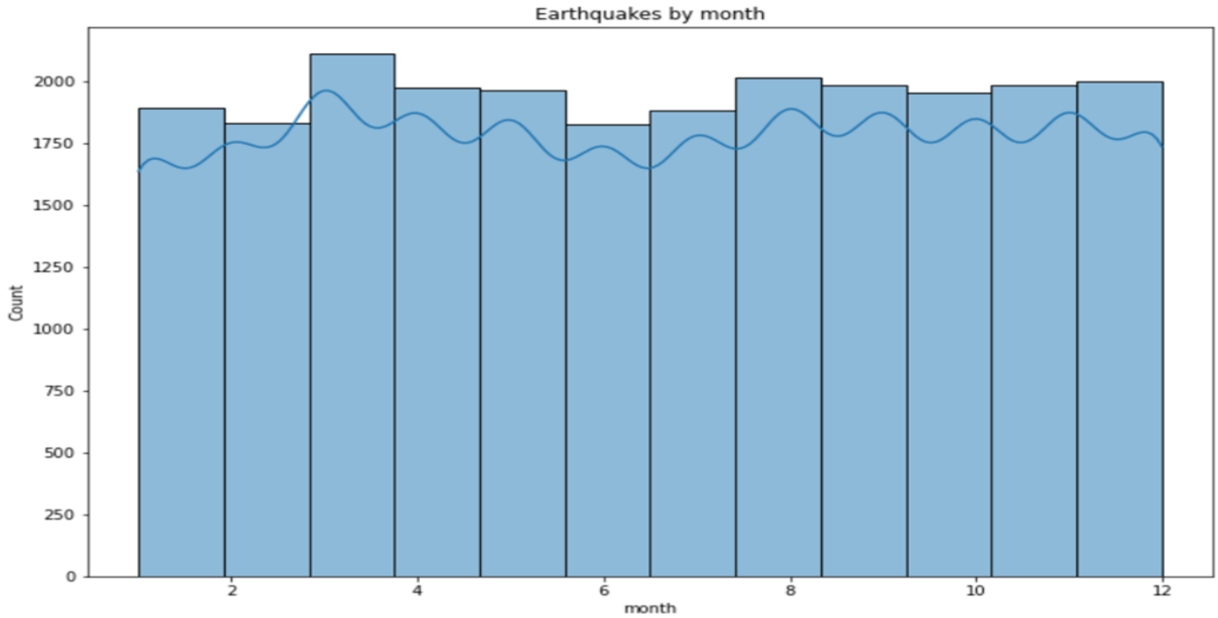
In [34]:
```
IC_df.info()
```

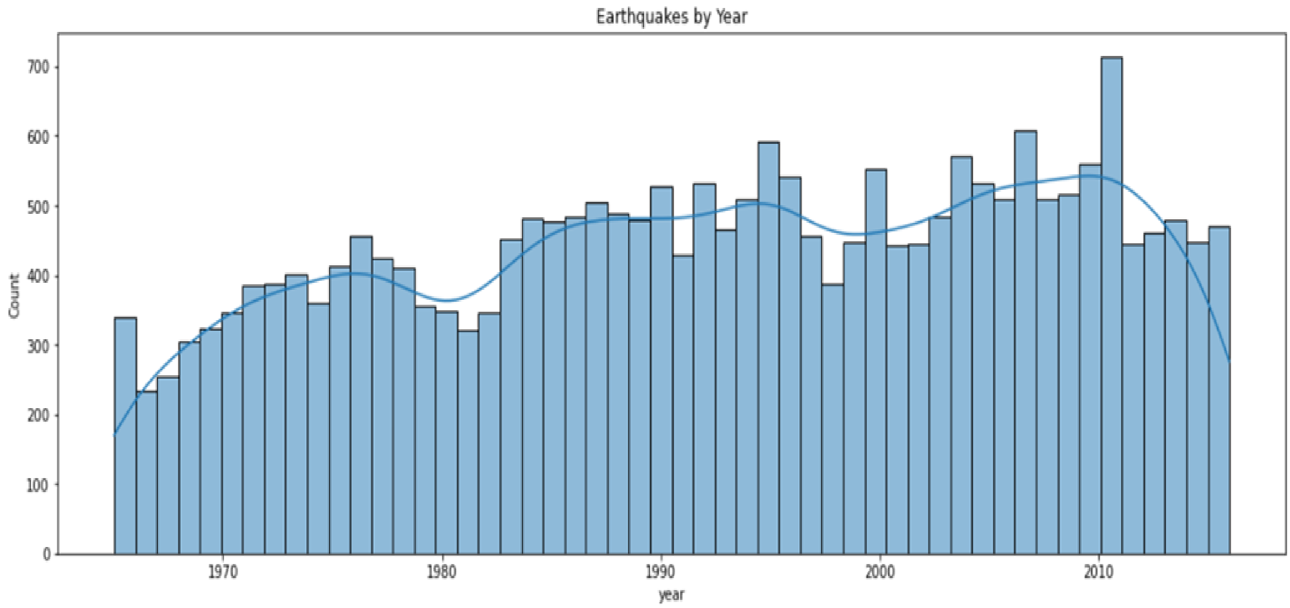
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23407 entries, 0 to 23411
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   month             23407 non-null  int64
 1   year              23407 non-null  int64
 2   Latitude          23407 non-null  float64
 3   Longitude         23407 non-null  float64
 4   Depth             23407 non-null  float64
 5   Magnitude         23407 non-null  float64
 6   Impacted Country  23407 non-null  object
 7   Magnitude Type    23407 non-null  object
dtypes: float64(4), int64(2), object(2)
memory usage: 1.6+ MB
```

After that I cleaned the data by dropping NA and duplicates, moved to the next part of the process which is Exploratory Data Analysis (EDA). Here are some of the plots which gives a lot of information about the data.



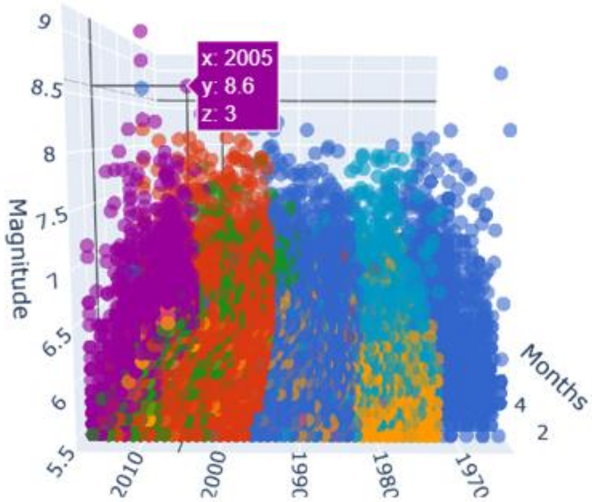



Earthquakes by month
Count
month
0
250
500
750
1000
1250
1500
1750
2000
2
4
6
8
10
12

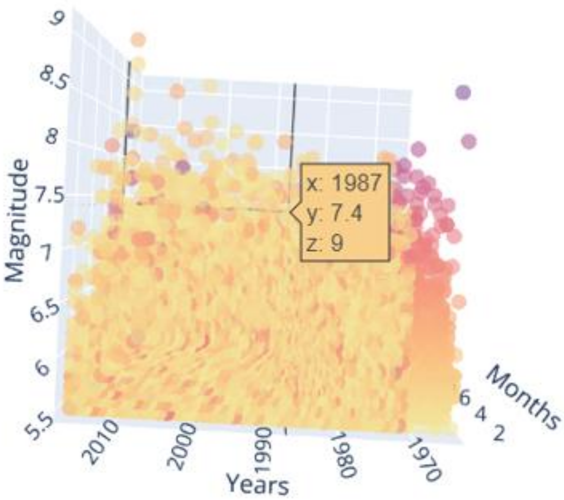

Earthquakes by Year
Count
year
0
100
200
300
400
500
600
700
1970
1980
1990
2000
2010

I have also plotted 3D interactive visualizations using plotly to explore the data in a better way. This shows us data pattern among three variables in a visual way.

Years Vs Magnitude Vs Months with different Magnitude Types

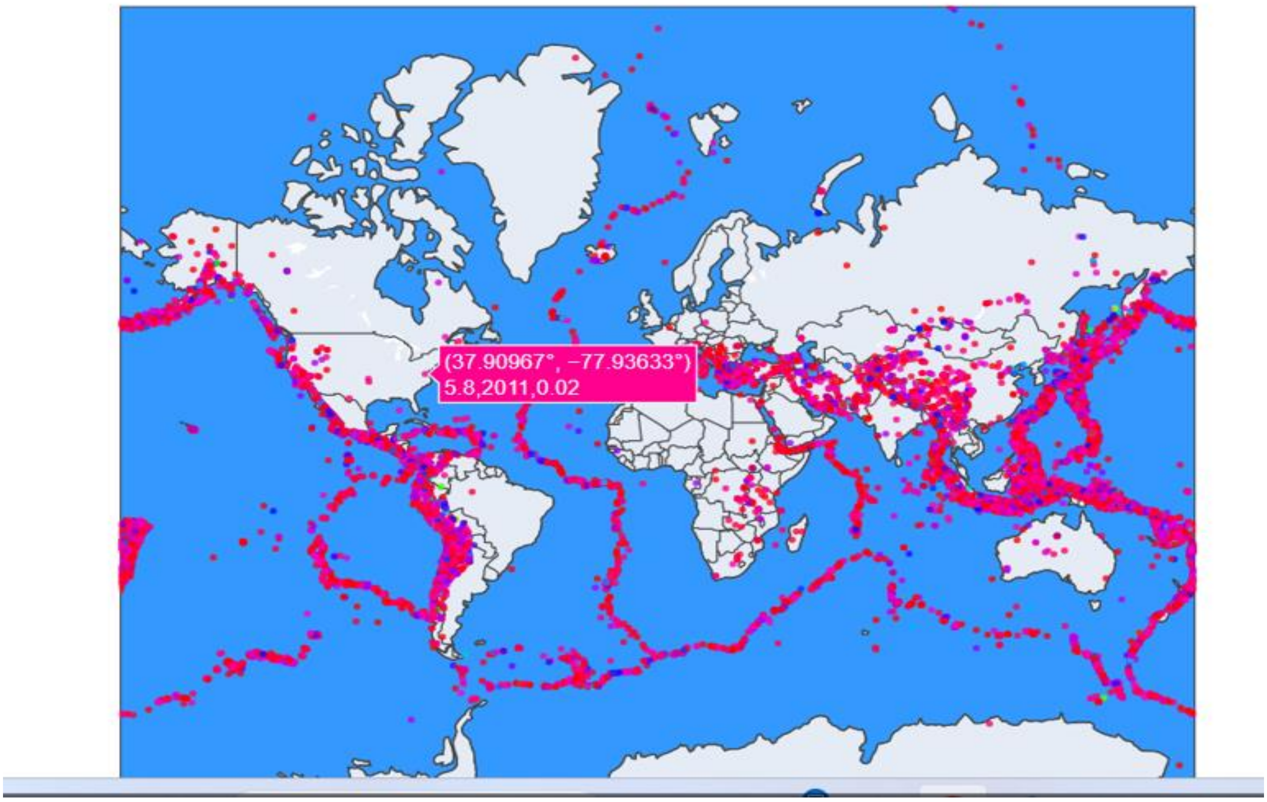


Years Vs Magnitude Vs Month

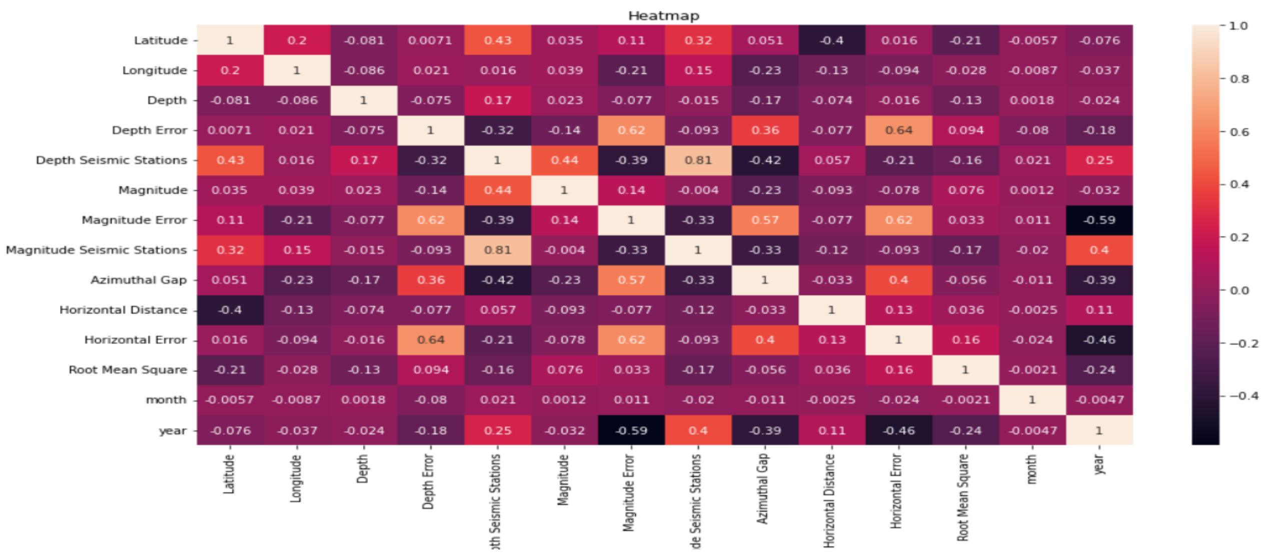


Here are all the Earthquakes plotted on the world map with different parameters highlighted. This gives us a clear picture of the pattern followed by such events.

Earthquakes on worldmap with Lat, Long, Magnitude, Year, Depth



Plotted a heatmap to get the correlation between the variables. These variables show hardly any correlation between the variables and with the target variable. This is expected as well because of how random and unpredictable these seismic events are.



With these and some more plots I figured the important variables and started building a prediction model. The data is split into 70-30% to train and test the predictions.

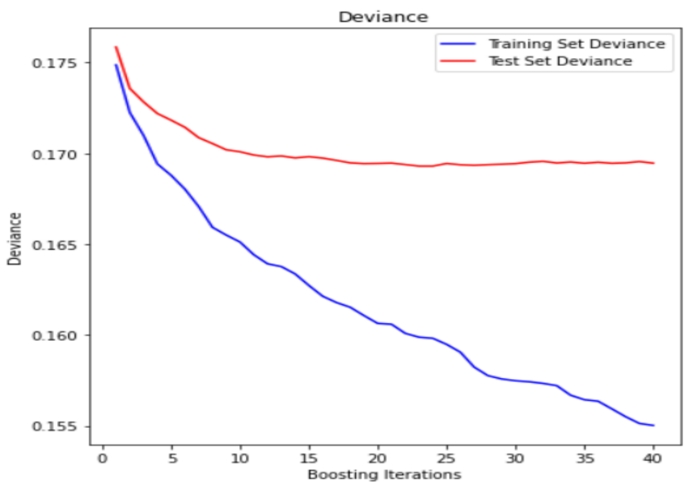
- **Gradient Boosting**

The first model I used is Gradient Boosting. Also Implemented GridSearchCV to find the best params. The output can be seen below. The model performed fairly well.

```
GB Model working with test size=0.20
---------------------------------------------
     learning_rate  max_depth  min_samples_leaf  min_samples_split  \
0              0.3          4                 4                  4
1              0.3          4                 4                  4
2              0.3          4                 4                  4
3              0.3          4                 4                  6
4              0.3          4                 4                  6
..             ...        ...               ...                ...
319            0.7         20                10                  6
320            0.7         20                10                  6
321            0.7         20                10                 10
322            0.7         20                10                 10
323            0.7         20                10                 10

     n_estimators  Accuracy
0              40  0.046190
1              70  0.040062
2              90  0.034950
3              40  0.046114
4              70  0.039957
..            ...       ...
319            70 -0.439904
320            90 -0.461458
321            40 -0.372637
322            70 -0.430281
323            90 -0.450859

[324 rows x 6 columns]

     [324 rows x 6 columns]
     ---------------------------------------------
     Best parameter setting with test size=0.20:  {'learning_rate': 0.3, 'max_depth': 4, 'min_samples_leaf': 10, 'min_samples_spli
     t': 10, 'n_estimators': 40}
     Best accuracy score with test size=0.20:  0.052045766851685174
     ------------------------------------------------------------------------------------------

Out[15]: ' GBClf_33=GridSearchCV(GB_model,param_grid=gb_grid,cv=10)\nGBClf_33.fit(X_train_33,y_train_33)\nprint("GB Model working with t
     est size=0.33")\nprint("---------------------------------------------")\nparams_combine_GB_33=pd.concat([pd.DataFrame(GBClf_33.cv
     _results_["params"]),pd.DataFrame(GBClf_33.cv_results_["mean_test_score"], columns=["Accuracy"])],axis=1)\nprint(params_combine
     _GB_33)\nprint("---------------------------------------------")\nprint("Best parameter setting with test size=0.33: ",GBClf_33.be
     st_params_)\nprint("Best accuracy score with test size=0.33: ",GBClf_33.best_score_)\nprint("-----------------------------------
     -----------------------------------------------------") '
```

The Root Mean Squared Error (RMSE) : 0.4117

Model Accuracy: 0.056



- **Neural Networks**

Then I moved to Neural Networks. I tried ANN with two different ways. Used 'relu' activation with 'adam' optimizer in the model to minimize the 'mse'. In the First one I fit the model without scaling the data. This gave me the following output.

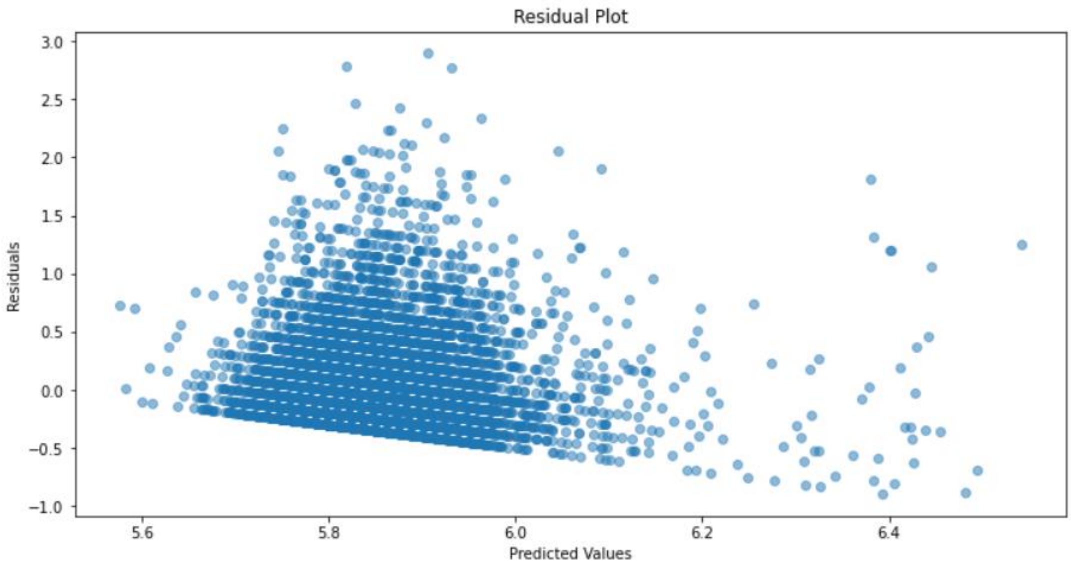
The Root Mean Squared Error (RMSE) : 0.4396

Model Accuracy: 0.1932

In the second one, I scaled the data using **StandardScalar**, changed some params to optimize the 'mse'. This produced the following results:

The Root Mean Squared Error (RMSE) : 0.4185

Model Accuracy: 0.1752



- **Random Forest**

Then I figured, the heatmap shows the two most important variable predicting the Magnitude are 'Depth Seismic Stations' and 'Azimuthal Gap'. I have dropped these columns due to a lot of missing values. So, I included these two variables and a lot of more variables with creating dummy variables. After cleaning this data, the total number of rows reduced to 5773 from 23000 as shown below:

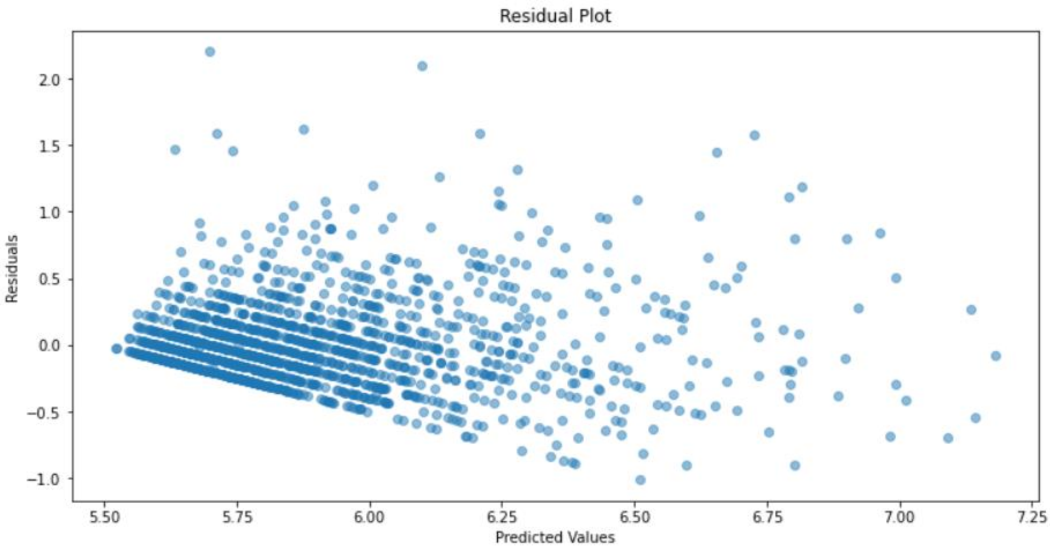
	Magnitude	Latitude	Longitude	Depth	Depth Seismic Stations	Azimuthal Gap	month	year	Type_Earthquake	Type_Explosion	...	Magnitu
count	5773.000000	5773.000000	5773.000000	5773.000000	5773.000000	5773.000000	5773.000000	5773.000000	5773.000000	5773.0	...	
mean	5.876769	1.474721	39.680042	61.696550	286.173047	45.636726	6.346267	2007.496969	0.997748	0.0	...	
std	0.425155	29.922187	124.158689	119.617467	165.854909	33.390791	3.465972	4.360830	0.047404	0.0	...	
min	5.500000	-77.080000	-179.996000	-1.100000	0.000000	0.000000	1.000000	1966.000000	0.000000	0.0	...	
25%	5.600000	-18.701000	-76.489000	10.000000	155.000000	25.500000	3.000000	2005.000000	1.000000	0.0	...	
50%	5.700000	-2.768000	97.957000	22.000000	268.000000	37.000000	6.000000	2008.000000	1.000000	0.0	...	
75%	6.000000	27.324000	143.083000	40.800000	396.000000	54.900000	9.000000	2011.000000	1.000000	0.0	...	
max	9.100000	85.263000	179.998000	688.000000	934.000000	360.000000	12.000000	2016.000000	1.000000	0.0	...	

8 rows × 109 columns

I used Random Forest Regressor on this new data with **GridSearchCV** to find best param and got much better results than before.

The Root Mean Squared Error (RMSE) : 0.32

Model Accuracy: 0.42



- **Conclusion**

Based on the comparison of R-Square and RMSE values, we can conclude that Random Forest produces the most accurate predictions, with an accuracy of **42%** and a Root Mean Square Error of **0.32**. This means that, given specific Latitude and Longitude coordinates, as well as other relevant variables, we can use this algorithm to predict the potential magnitude range of an earthquake occurring in a particular Month and Year. For example, if the model predicts a magnitude of 6.5 with a RMSE of 0.3, we can estimate that the potential range of magnitude could be between **6.2 and 6.8**. This information is critical in preparing for the potential impact of this natural disaster.