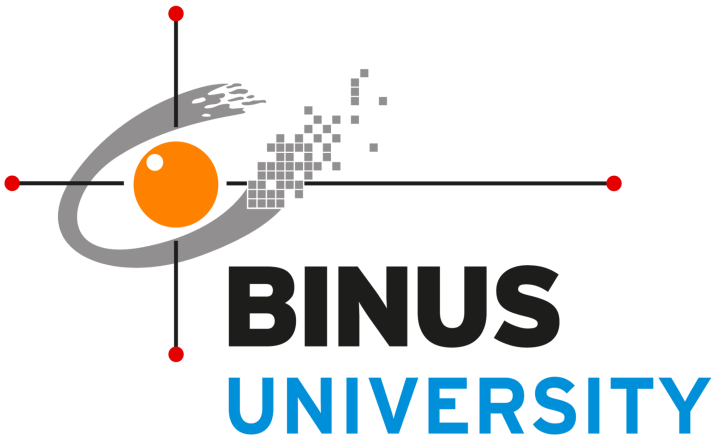
THE EFFECT OF MAGNITUDE AND DEPTH ON EARTHQUAKES

Fundamentals of Data Science



ARRANGED BY:

CHELLSHE LOVE SIMROCHELLE (2502043040)

RAISSA AZARIA (2502005805)

PRESENTED TO:

Mr. IDA BAGUS KERTHYAYANA MANUABA, S.T., Ph.D.

Ms. NUNUNG NURULQOMARIYAH, S.Kom., M.T.I., Ph.D.

Faculty of Computing and Media

BINUS INTERNATIONAL UNIVERSITY

2022

Problem Analysis

Earthquakes are felt on the Earth's surface as shockwaves or intense vibrations. Seismic waves are typically caused by ruptures along geological fault lines in the Earth's crust, resulting in the sudden release of energy. They can also be caused by volcanic activity or human-caused events such as industrial or military explosions.

Earthquakes have the potential to devastate our built environments as well as the critical systems on which we rely for our survival and livelihood. They also have the potential to cause landslides and tsunamis (large ocean waves that can flood and destroy coastal areas), both of which can have catastrophic consequences for people and communities. Earthquakes can have far-reaching social and economic consequences, and recovery can take years. That is why earthquakes are one of the most unpredictable and life-changing natural disasters on the planet. The chance of you actually feeling one below your feet is relatively low, but they can occur on every continent worldwide. The U.S. Geological Survey (USGS) estimates that around 500,000 earthquakes occur yearly. Many of those happen deep in Earth’s crust and without the use of seismographs, they would go undetected. Seismologists estimate that only around 20% of the world’s earthquakes are felt by humans and about 100 each year cause damage. The United States is ranked first for the country with the most natural disasters in 2021 and earthquakes are one of them. No one, not even the best scientists at the USGS, can predict an earthquake. Experts say it is because the mechanisms that trigger shaking happen so far under the ground in slow motion. But advances in recent years have given people a better timeframe of when to expect an earthquake.

One of the ways to measure earthquakes is using the magnitude. Magnitude generally implies a quantity or distance. We can relate the magnitude of the movement to the size and movement speed of the object. The magnitude of a thing or an amount is its size (Borman, 2021). There are a lot of things to measure the earthquake.

This project aims to determine whether the magnitude or depth will affect earthquakes in any category since the magnitude and depth of an earthquake take an important part in a natural disaster.

Hypothesis

Null hypothesis: The magnitude and depth did not affect the earthquake

Alternative hypothesis: The magnitude and depth did affect the earthquake

About Dataset

This set of data is from the USGS (U.S. Geological Survey). The USGS offers reliable scientific data to characterize and understand the Earth, reduce property damage and human casualties from natural catastrophes, manage water, biological, and energy resources, and improve and safeguard our quality of life. The dataset contains details of all earthquakes that have happened in the last 30 days and is updated every 15 mins on the USGS website. The original owner of this dataset has uploaded this dataset with updated settings to weekly levels. In this dataset, they have several columns that will help us to work with the dataset such as time, latitude, longitude, depth, mag, magType, nst, gap, dmin and rms with the explanation below:

-Latitude: measures the distance north or south of the equator. Latitude lines start at the equator (0 degrees latitude) and run east and west, parallel to the equator. Lines of latitude are measured in degrees north or south of the equator to 90 degrees at the North or South poles

-Longitude: Longitude measures the distance east or west of the prime meridian. Lines of longitude, also called meridians, are imaginary lines that divide the Earth. They run north to south from pole to pole, but they measure the distance east or west. Longitude is measured in degrees, minutes, and seconds.

- Depths: the depths of earthquakes gives us important information about the Earth's structure and the tectonic setting where the earthquakes are occurring.

- Magnitude: Magnitude is the most common measure of an earthquake's size. It is a measure of the size of the earthquake source and is the same number no matter where you are or what the shaking feels like.

- Magnitude type: method or formula used to determine the event's optimal magnitude.

- Nst: Number of seismic stations which reported P- and S-arrival times for this earthquake. This number may be larger than Nph if arrival times are rejected because the distance to a seismic station exceeds the maximum allowable distance or because the arrival-time observation is inconsistent with the solution.

- Gap: Region along an active fault where stress is accumulating because no earthquakes have occurred there recently. Seismic gaps are often flanked by areas that have experienced earthquakes in the near past. Scientists often consider these regions to be high-risk areas for earthquakes in the near future.

- Dmin: Horizontal distance from the epicenter to the nearest station (in km). In general, the smaller this number, the more reliable the calculated depth of the earthquake.

- rms: For seismic integration, RMS is the most commonly used post-stack amplitude attribute, it computes the square root of the sum of squared amplitude values divided by the number of samples within the specified window.

-net: Data contributor's ID. Indicates which network is thought to be the most reliable source of information about this occurrence.

- id: id contains a unique value to name each earthquake so they will have different data from the other.

- Updated: this column shows the date and time when did they update the data
- Place: tell us about the location detail of the earthquake. This also includes the wind direction when the disaster happens.
- Type: in this dataset, earthquakes become the number one accident that happens and then followed by quarry blasts, explosions, ice quakes and other events

- Horizontal error: The length of the biggest projection of the three primary mistakes on a horizontal plane is the horizontal location error, expressed in kilometers (km). The main axes of the error ellipsoid, known as the principal errors, are mutually perpendicular. For the best locations, those in the midst of closely spaced seismograph networks, the horizontal and vertical uncertainties in the location of an event range from around 100 m horizontally and 300 m vertically to 10s of kilometers for global events in many parts of the world.

- Depth error: Uncertainty of reported depth of the event in kilometers. The depth error, in km, is defined as the largest projection of the three principal errors on a vertical line.

- Mag error: Uncertainty regarding the event's claimed magnitude. The magnitude's estimated standard error. The uncertainty is particular to the magnitude type being reported and does not account for biases and changes in magnitude between scales.

- magNst: the total number of seismic stations utilized to determine this earthquake's magnitude.

- Status: Status of the product. There is only one reserved status DELETE, which indicates the product has been deleted. Any other value indicates a product update and may vary depending on product type.

- Location Source: The network that originally authored the reported location of this event.

- MagSource: Network that originally authored the reported magnitude for this event.

Methodology/Output

I. Identifying Topic

Identifying topics for research is the first and most important thing before starting to do research. There are several things to keep in mind before identifying a topic. We need to have a clear vision of the problem we want to solve using our research. Finding the appropriate and suitable data for research is also needed. Since most of the time, people get information easily through the internet, we need to check whether the data is accurate.

II. Research

Doing research after identifying topics is the second step in making our research paper. Research here means that we need to find whether there are any similar works that have been published or not. This step was done to prevent plagiarism, even though there would be quite similarities between our paper and the other one, we prevent plagiarism which is why research is needed before continuing to the next step. Besides that, we also need to do further research on the data used. We need to make sure that the data matches the research we are working on now.

III. Data Preparing

First we installed basemap, the matplotlib basemap toolkit is a library for plotting 2D data on maps in Python as we are going to be using it to be plotting the geographical visualization. It is similar in functionality to the matlab mapping toolbox, the IDL mapping facilities, GrADS, or the Generic Mapping Tools. PyNGL and CDAT are other libraries that provide similar capabilities in Python.

```
[224] !pip install basemap

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: basemap in /usr/local/lib/python3.8/dist-packages (1.3.6)
Requirement already satisfied: pyshp<2.4,>=1.2 in /usr/local/lib/python3.8/dist-packages (from basemap) (2.3.1)
Requirement already satisfied: numpy<1.24,>=1.22 in /usr/local/lib/python3.8/dist-packages (from basemap) (1.23.5)
Requirement already satisfied: basemap-data<1.4,>=1.3.2 in /usr/local/lib/python3.8/dist-packages (from basemap) (1.3.2)
Requirement already satisfied: pyproj<3.5.0,>=1.9.3 in /usr/local/lib/python3.8/dist-packages (from basemap) (3.4.1)
Requirement already satisfied: matplotlib<3.7,>=1.5 in /usr/local/lib/python3.8/dist-packages (from basemap) (3.2.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=1.5->basemap) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=1.5->basemap) (1.4.4)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=1.5->basemap) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=1.5->basemap) (3.0.9
Requirement already satisfied: certifi in /usr/local/lib/python3.8/dist-packages (from pyproj<3.5.0,>=1.9.3->basemap) (2022.12.7)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.1->matplotlib<3.7,>=1.5->basemap) (1.15.0)
```

Import the necessary libraries required for building the model and data analysis of the earthquakes.

```
[225] # Basic library
      import pandas as pd # Data analysis
      import numpy as np # Array handler
      import matplotlib.pyplot as plt #Visualization
      import seaborn as sns #Visualization
      import re
      import missingno as msno
      import warnings
      import random as rd

      # clustering
      from sklearn.cluster import KMeans # Clustering machine learning
      from sklearn.datasets import make_blobs
      from sklearn.metrics import silhouette_score
      from sklearn.datasets import make_blobs

      # preprocessing
      from sklearn.preprocessing import MinMaxScaler # Features scaler
      from mpl_toolkits.basemap import Basemap

      # upload csv file
      from google.colab import files
```

Read the data from the csv file and the columns necessary for the model and the column that needs to be predicted.

```
[227] # data loading
      df = pd.read_csv("all_month.csv")

      warnings.filterwarnings("ignore")

      x = df.iloc[:, [0, 1, 2, 3]].values
```

```
[226] # reading dataset
      !gdown --id 1rjTf2IPpX4S8uYpyQJvhSStIM_RT4WRB

/usr/local/lib/python3.8/dist-packages/gdown/cli.py:127: FutureWarning: Option `--id` was deprecated in version 4.3.1 and will be removed in 5.0. You don't n
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1rjTf2IPpX4S8uYpyQJvhSStIM_RT4WRB
To: /content/all_month.csv
100% 2.08M/2.08M [00:00<00:00, 168MB/s]
```

```
[228] # checking the first 5 rows of the data
      df.head()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalError	depthError	magErro
0	2022-12-24T02:23:14.097Z	56.963300	-155.453700	25.900000	2.70	ml	NaN	NaN	NaN	0.38	...	2022-12-24T02:25:36.127Z	78 km W of Akhiok, Alaska	earthquake	NaN	0.700	Na
1	2022-12-24T02:22:21.110Z	19.219999	-155.429993	33.400002	1.94	md	28.0	144.0	NaN	0.14	...	2022-12-24T02:25:35.660Z	5 km ENE of Pāhala, Hawaii	earthquake	0.72	0.840	1.98
2	2022-12-24T01:50:43.200Z	19.248167	-155.395340	31.740000	2.15	ml	41.0	134.0	NaN	0.14	...	2022-12-24T01:56:14.230Z	10 km ENE of Pāhala, Hawaii	earthquake	0.75	0.740	0.21
3	2022-12-24T01:47:09.698Z	-5.123400	153.304600	40.437000	4.70	mb	23.0	128.0	1.466	0.73	...	2022-12-24T02:23:27.040Z	New Ireland region, Papua New Guinea	earthquake	12.19	7.840	0.12
4	2022-12-24T01:39:51.776Z	44.090900	148.181500	57.073000	4.60	mb	45.0	169.0	4.025	0.59	...	2022-12-24T01:57:44.040Z	Kuril Islands	earthquake	9.06	5.403	0.06

5 rows × 22 columns

```
# checking the last 5 rows of the data
df.tail()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalError	depthError	m
10941	2022-11-24T03:00:29.990Z	37.146667	-121.542333	3.71	0.79	md	17.0	50.0	0.069260	0.1400	...	2022-11-24T12:41:15.684Z	9km NE of San Martin, CA	earthquake	0.29	0.96	
10942	2022-11-24T02:53:01.020Z	18.308833	-67.168667	14.96	2.49	md	3.0	328.0	0.306000	0.1300	...	2022-11-24T04:00:27.327Z	2 km NW of Las Marias, Puerto Rico	earthquake	1.44	12.41	
10943	2022-11-24T02:42:08.540Z	33.758000	-116.917000	10.62	0.67	ml	21.0	129.0	0.070960	0.0900	...	2022-11-28T20:36:10.575Z	2km WNW of Valle Vista, CA	earthquake	0.23	0.35	
10944	2022-11-24T02:35:21.590Z	38.799333	-122.751167	1.51	1.38	md	44.0	47.0	0.006801	0.0500	...	2022-11-26T10:46:12.336Z	2km N of The Geysers, CA	earthquake	0.16	0.27	
10945	2022-11-24T02:33:35.197Z	39.595700	-119.086100	0.00	1.80	ml	24.0	115.3	0.258000	0.2032	...	2022-11-24T16:04:41.194Z	14 km E of Fernley, Nevada	earthquake	NaN	0.00	

5 rows × 22 columns

```
[230] # show all column in dataset
      df.columns
```

```
Index(['time', 'latitude', 'longitude', 'depth', 'mag', 'magType', 'nst',
       'gap', 'dmin', 'rms', 'net', 'id', 'updated', 'place', 'type',
       'horizontalError', 'depthError', 'magError', 'magNst', 'status',
       'locationSource', 'magSource'],
      dtype='object')
```

The df.info() method returns data about the DataFrame. The data includes the number of columns, column labels, column data types, memory usage, range index, and cell count for each column (non-null values). And the df[1:10] method prints out the first 10 rows of the DataFrame similar to df.head and df.tail.

```
[231] # prints information about dataset
      df.info()
      df[0:10]

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 10946 entries, 0 to 10945
      Data columns (total 22 columns):
       #   Column           Non-Null Count  Dtype
      ---  ------           --------------  -----
       0   time             10946 non-null  object
       1   latitude         10946 non-null  float64
       2   longitude        10946 non-null  float64
       3   depth            10946 non-null  float64
       4   mag              10943 non-null  float64
       5   magType          10943 non-null  object
       6   nst              8032 non-null   float64
       7   gap              8032 non-null   float64
       8   dmin             5912 non-null   float64
       9   rms              10946 non-null  float64
       10  net              10946 non-null  object
       11  id               10946 non-null  object
       12  updated          10946 non-null  object
       13  place            10946 non-null  object
       14  type             10946 non-null  object
       15  horizontalError  7157 non-null   float64
       16  depthError       10945 non-null  float64
       17  magError         7893 non-null   float64
       18  magNst           8019 non-null   float64
       19  status           10946 non-null  object
       20  locationSource   10946 non-null  object
       21  magSource        10946 non-null  object
      dtypes: float64(12), object(10)
      memory usage: 1.8+ MB
```

[231]

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalError	depthError	magEr
0	2022-12-24T02:23:14.097Z	56.963300	-155.453700	25.900000	2.70	ml	NaN	NaN	NaN	0.38	...	2022-12-24T02:25:36.127Z	78 km W of Akhiok, Alaska	earthquake	NaN	0.700	N
1	2022-12-24T02:22:21.110Z	19.219999	-155.429993	33.400002	1.94	md	28.0	144.0	NaN	0.14	...	2022-12-24T02:25:35.660Z	5 km ENE of Pāhala, Hawaii	earthquake	0.72	0.840	1.
2	2022-12-24T01:50:43.200Z	19.248167	-155.395340	31.740000	2.15	ml	41.0	134.0	NaN	0.14	...	2022-12-24T01:56:14.230Z	10 km ENE of Pāhala, Hawaii	earthquake	0.75	0.740	0.
3	2022-12-24T01:47:09.698Z	-5.123400	153.304600	40.437000	4.70	mb	23.0	128.0	1.46600	0.73	...	2022-12-24T02:23:27.040Z	New Ireland region, Papua New Guinea	earthquake	12.19	7.840	0.
4	2022-12-24T01:39:51.776Z	44.090900	148.181500	57.073000	4.60	mb	45.0	169.0	4.02500	0.59	...	2022-12-24T01:57:44.040Z	Kuril Islands	earthquake	9.06	5.403	0.
5	2022-12-24T01:35:21.590Z	19.179832	-155.471161	33.470001	2.26	md	37.0	89.0	NaN	0.11	...	2022-12-24T01:38:28.740Z	2 km SSE of Pāhala, Hawaii	earthquake	0.66	0.930	0.
6	2022-12-24T01:33:35.390Z	36.982666	-121.634499	4.110000	1.36	md	8.0	94.0	0.05614	0.06	...	2022-12-24T01:59:12.019Z	6km WSW of Gilroy, CA	earthquake	0.40	1.030	0.
7	2022-12-24T01:33:05.490Z	36.982834	-121.634003	4.830000	1.73	md	11.0	91.0	0.05645	0.05	...	2022-12-24T01:49:11.962Z	6km WSW of Gilroy, CA	earthquake	0.28	0.750	0.
8	2022-12-24T01:32:52.642Z	58.187900	-155.358200	1.300000	0.80	ml	NaN	NaN	NaN	0.16	...	2022-12-24T01:35:26.317Z	87 km NW of Karluk, Alaska	earthquake	NaN	1.000	N
9	2022-12-24T01:19:05.660Z	19.257166	-155.388000	30.500000	2.17	md	29.0	132.0	NaN	0.12	...	2022-12-24T01:22:16.610Z	11 km ENE of Pāhala, Hawaii	earthquake	0.68	1.010	0.

10 rows × 22 columns

Next, we need to check if there are any NULL values and we do that by using the df.isnull() method. The df.isnull() method returns a DataFrame object in which all values have been replaced with the Boolean value True for NULL values and False otherwise.

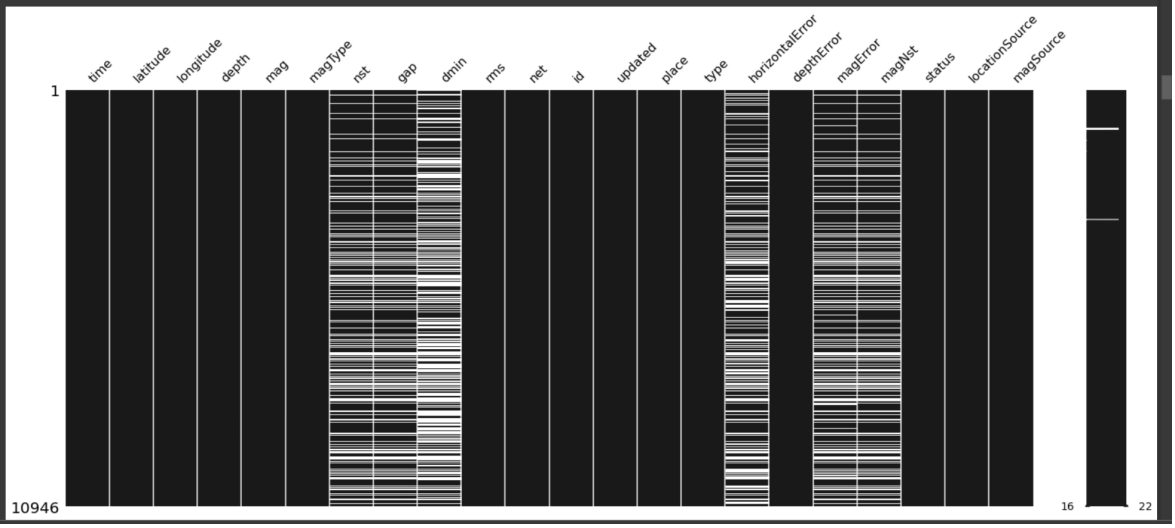
```
[232] # returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False.
      df.isnull()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalError	depthError	magError	magNst	status	l
0	False	False	False	False	False	False	True	True	True	False	...	False	False	False	True	False	True	True	False	
1	False	False	False	False	False	False	False	False	True	False	...	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	True	False	...	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10941	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	
10942	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	
10943	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	
10944	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	
10945	False	False	False	False	False	False	False	False	False	False	...	False	False	False	True	False	False	False	False	

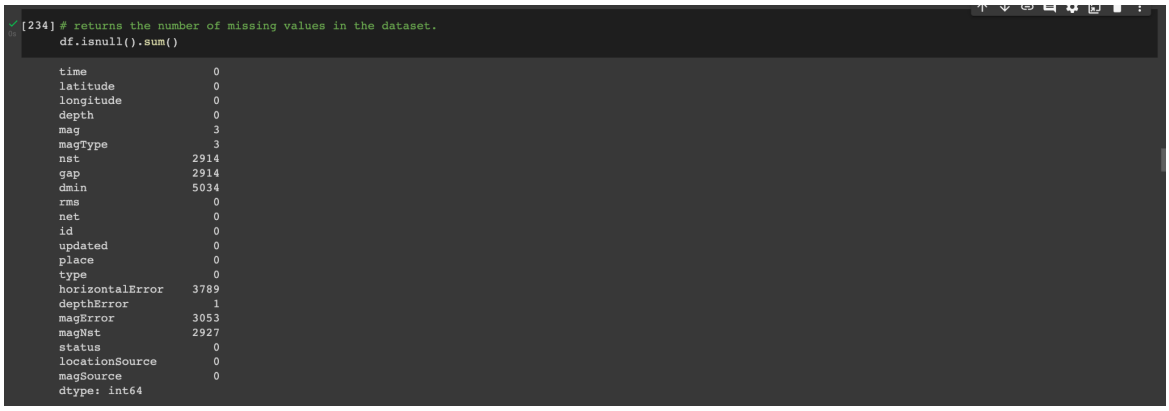
10946 rows × 22 columns

After displaying all the missing values from our DataFrame we visualized it by formatting it with a matrix visualizer. The msno.matrix nullity matrix is a data-dense display that allows you to quickly identify patterns in data completion. The missingno library that was imported in the start provides various visualizations that allow us to visualize and analyze missing values (NaNs/NULLs/None) in our dataset from various perspectives. This is very useful in dealing with missing data.

```
# showing missing value in each
msno.matrix(df, color="0.1")
plt.show()
```

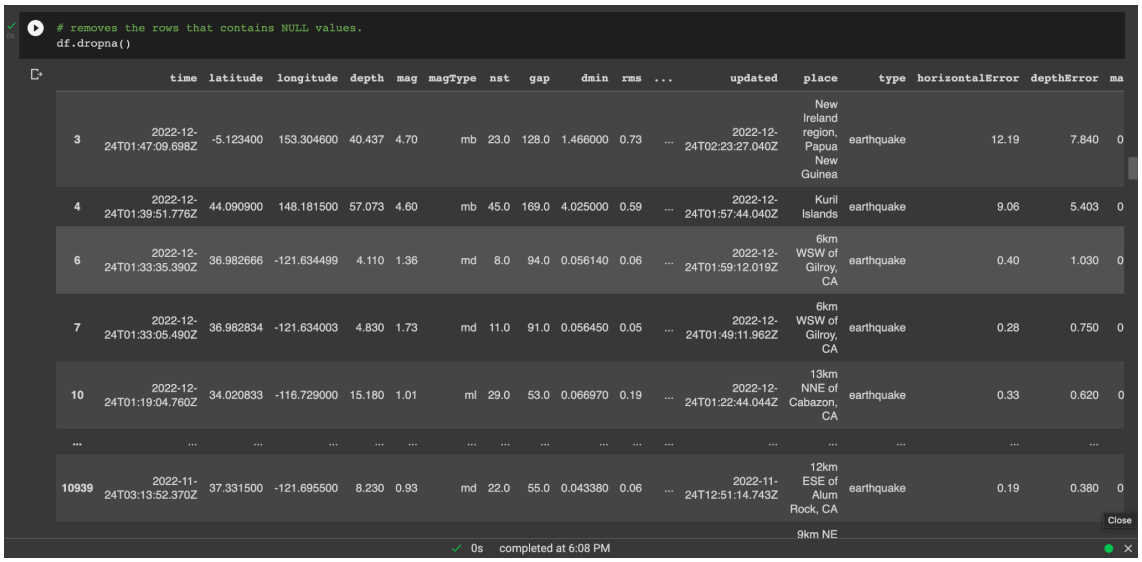


The function dataframe. isnull(). sum(). sum() returns the number of missing values in the dataset.



```
[234] # returns the number of missing values in the dataset.
      df.isnull().sum()

time                 0
latitude             0
longitude            0
depth                0
mag                  3
magType              3
nst               2914
gap               2914
dmin              5034
rms                  0
net                  0
id                   0
updated              0
place                0
type                 0
horizontalError   3789
depthError           1
magError          3053
magNst            2927
status               0
locationSource       0
magSource            0
dtype: int64
```

The dropna() method deletes rows that have NULL values. Unless the inplace parameter is set to True, the dropna() method returns a new DataFrame object; otherwise, the dropna() method removes the data from the original DataFrame.



```
# removes the rows that contains NULL values.
df.dropna()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalError	depthError	ma
3	2022-12-24T01:47:09.698Z	-5.123400	153.304600	40.437	4.70	mb	23.0	128.0	1.466000	0.73	...	2022-12-24T02:23:27.040Z	New Ireland region, Papua New Guinea	earthquake	12.19	7.840	0
4	2022-12-24T01:39:51.776Z	44.090900	148.181500	57.073	4.60	mb	45.0	169.0	4.025000	0.59	...	2022-12-24T01:57:44.040Z	Kuril Islands	earthquake	9.06	5.403	0
6	2022-12-24T01:33:35.390Z	36.982666	-121.634499	4.110	1.36	md	8.0	94.0	0.056140	0.06	...	2022-12-24T01:59:12.019Z	6km WSW of Gilroy, CA	earthquake	0.40	1.030	0
7	2022-12-24T01:33:05.490Z	36.982834	-121.634003	4.830	1.73	md	11.0	91.0	0.056450	0.05	...	2022-12-24T01:49:11.962Z	6km WSW of Gilroy, CA	earthquake	0.28	0.750	0
10	2022-12-24T01:19:04.760Z	34.020833	-116.729000	15.180	1.01	ml	29.0	53.0	0.066970	0.19	...	2022-12-24T01:22:44.044Z	13km NNE of Cabazon, CA	earthquake	0.33	0.620	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
10939	2022-11-24T03:13:52.370Z	37.331500	-121.695500	8.230	0.93	md	22.0	55.0	0.043380	0.06	...	2022-11-24T12:51:14.743Z	12km ESE of Alum Rock, CA	earthquake	0.19	0.380	0

9km NE

Close

0s completed at 6:08 PM

The fillna() method replaces NULL values with the value specified which in this case we changed it to the number 0. Unless the inplace parameter is set to True, the fillna() method returns a new DataFrame object; otherwise, the fillna() method replaces the original DataFrame.

```
[236] # fills the NULL values with a 0
      df.fillna(0,inplace=True)
```

The df.replace() method replaces a string, regex, list, dictionary, series, number, and so on. After a thorough search of the entire DataFrame, every instance of the provided value is replaced. In this case we changed the letter “T” and “Z” from the “time” column into and empty space because it will make it harder for us if we did not do it as we will encounter an error that says “valueerror: could not convert string to float”.

```
# replacing the letters T and Z to a space
df = df.replace('Z', '', regex=True)
df = df.replace('T', ' ', regex=True)
print(df)
```

```
                      time   latitude   longitude      depth   mag  \
0      2022-12-24 02:23:14.097  56.963300 -155.453700  25.900000  2.70
1      2022-12-24 02:22:21.110  19.219999 -155.429993  33.400002  1.94
2      2022-12-24 01:50:43.200  19.248167 -155.395340  31.740000  2.15
3      2022-12-24 01:47:09.698  -5.123400  153.304600  40.437000  4.70
4      2022-12-24 01:39:51.776  44.090900  148.181500  57.073000  4.60
...                        ...        ...         ...        ...   ...
10941  2022-11-24 03:00:29.990  37.146667 -121.542333   3.710000  0.79
10942  2022-11-24 02:53:01.020  18.308833  -67.168667  14.960000  2.49
10943  2022-11-24 02:42:08.540  33.758000 -116.917000  10.620000  0.67
10944  2022-11-24 02:35:21.590  38.799333 -122.751167   1.510000  1.38
10945  2022-11-24 02:33:35.197  39.595700 -119.086100   0.000000  1.80

      magType   nst    gap      dmin     rms  ...                updated  \
0          ml   0.0    0.0  0.000000  0.3800  ...  2022-12-24 02:25:36.127
1          md  28.0  144.0  0.000000  0.1400  ...  2022-12-24 02:25:35.660
2          ml  41.0  134.0  0.000000  0.1400  ...  2022-12-24 01:56:14.230
3          mb  23.0  128.0  1.466000  0.7300  ...  2022-12-24 02:23:27.040
4          mb  45.0  169.0  4.025000  0.5900  ...  2022-12-24 01:57:44.040
...       ...   ...    ...       ...     ...  ...                      ...
10941      md  17.0   50.0  0.069260  0.1400  ...  2022-11-24 12:41:15.684
10942      md   3.0  328.0  0.306000  0.1300  ...  2022-11-24 04:00:27.327
10943      ml  21.0  129.0  0.070960  0.0900  ...  2022-11-28 20:36:10.575
10944      md  44.0   47.0  0.006801  0.0500  ...  2022-11-26 10:46:12.336
10945      ml  24.0  115.3  0.258000  0.2032  ...  2022-11-24 16:04:41.194

                                    place        type horizontalError  \
0                78 km W of Akhiok, Alaska  earthquake            0.00
1                 5 km ENE of Pāhala, Hawaii  earthquake            0.72
2                10 km ENE of Pāhala, Hawaii  earthquake            0.75
3      New Ireland region, Papua New Guinea  earthquake           12.19
4                           Kuril Islands  earthquake            9.06
```

0s completed at 6:08 PM

The duplicated() method returns a Series containing True and False values indicating which rows in the DataFrame are duplicated and which are not. The drop_duplicates() method removes duplicate rows.

```
[238] # prints the data that has a duplicate
      print(df.duplicated())
```

```
0        False
1        False
2        False
3        False
4        False
         ...
10941    False
10942    False
10943    False
10944    False
10945    False
Length: 10946, dtype: bool
```

```
[239] # dropping the duplicates
      df.drop_duplicates(inplace = True)
```

The describe() method returns the data in the DataFrame's description. If the DataFrame contains numerical data, the description includes the following details for each column: count - The number of values that are not empty. mean - The mean (average) value.

```
# describe the data
df.describe()
```

	latitude	longitude	depth	mag	nst	gap	dmin	rms	horizontalError	depthError	magError	magNst
count	10946.000000	10946.000000	10946.00000	10946.000000	10946.000000	10946.000000	10946.000000	10946.000000	10946.000000	10946.000000	10946.000000	10946.000000
mean	40.251389	-119.206851	24.01988	1.651484	16.599488	89.517521	0.364471	0.282491	1.113645	2.096111	0.204121	10.660515
std	20.067601	64.164776	51.04860	1.193518	21.303481	77.229754	1.730739	0.255797	2.651342	15.964088	0.437827	20.326689
min	-65.380700	-179.940400	-3.74000	-1.330000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	33.225542	-153.305600	2.98250	0.900000	0.000000	0.000000	0.000000	0.100000	0.000000	0.410000	0.000000	0.000000
50%	38.830002	-122.858833	8.86500	1.450000	10.000000	81.000000	0.007345	0.180000	0.270000	0.700000	0.122864	6.000000
75%	58.252167	-116.774833	21.87350	2.100000	24.000000	136.307500	0.069998	0.440000	0.600000	1.480000	0.210000	13.000000
max	78.699100	179.961700	654.23700	6.700000	492.000000	353.210000	39.621000	3.810000	48.990000	1604.000000	5.220000	540.000000

The crosstab() function performs a simple cross-tabulation of two (or more) variables. Unless an array of values and an aggregation function are passed, compute a frequency table of the factors by default. In this case, we used it to perform a tabulation of the magnitude and depth of our dataset.

```
# standarized data
df_outcome = pd.crosstab(index=df["mag"],
                         columns="count")

df_outcome
```

col_0	count
mag	
-1.33	1
-1.23	1
-1.22	1
-1.10	1
-1.08	1
...	...
5.90	3
6.00	1
6.30	2
6.37	1
6.70	1

621 rows × 1 columns

```
# standarized data
df_outcome = pd.crosstab(index=df["depth"],
                         columns="count")

df_outcome
```

col_0	count
depth	
-3.740	1
-3.740	1
-3.730	1
-3.720	1
-3.620	1
...	...
603.180	1
603.358	1
605.651	1
618.929	1
654.237	1

4153 rows × 1 columns

Now we are going to standardize the DataFrame.

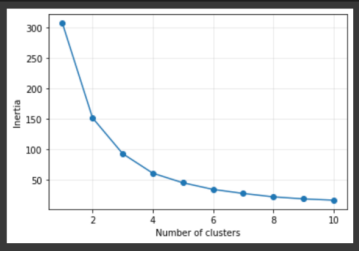
```
[244] # standarized data
      tempDf = df[['depth','mag']].copy()
      ms = MinMaxScaler()
      tempDf = ms.fit_transform(tempDf)
      tempDf = pd.DataFrame(tempDf, columns=['depth','mag'])
      tempDf.head()
```

	depth	mag
0	0.045047	0.501868
1	0.056446	0.407223
2	0.053923	0.433375
3	0.067141	0.750934
4	0.092424	0.738481

IV. Data Visualization

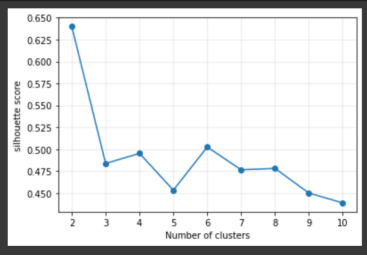
Then the elbow method is used to determine the optimal number of clusters in k-means clustering in our DataFrame.

```
# find optimum cluster amount: elbow method
inertia = []
for i in range(1,11):
    kMeans = KMeans(n_clusters = i)
    kMeans.fit(tempDf[['depth','mag']])
    inertia.append(kMeans.inertia_)
plt.plot(range(1,11), inertia, 'o-')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.grid(alpha = 0.3)
plt.show()
```



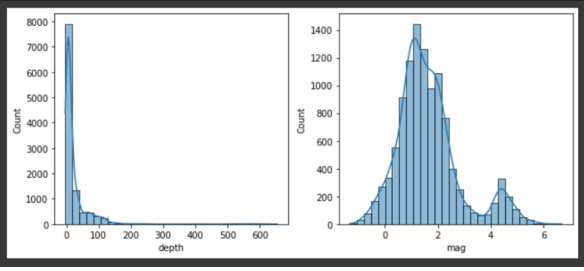
Next, we used the silhouette method to calculate silhouette coefficients for each point, which measure how similar a point is to its own cluster in comparison to other clusters by displaying a brief graphical representation of how well each object has been classified.

```
# find optimum cluster amount: sillhouette method
ss = []
for i in range(2,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(tempDf[['depth','mag']])
    cluster_labels = kmeans.labels_
    ss.append(silhouette_score(tempDf[['depth','mag']], cluster_labels))
plt.plot(range(2,11), ss, 'o-')
plt.xlabel('Number of clusters')
plt.ylabel('silhouette score')
plt.grid(alpha = 0.3)
plt.show()
```



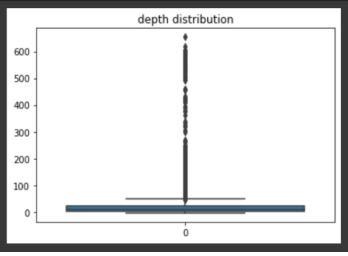
Plot the magnitude and depth into a histogram for visualization.

```
[247] # visualize the data
fig = plt.figure(figsize=(9,4), tight_layout=True)

for i, data in enumerate([df['depth'], df['mag']]):
    bins = 9 if i == 2 else 30
    ax = plt.subplot(1,2, i + 1)
    sns.histplot(data, bins=bins, kde=True)
```

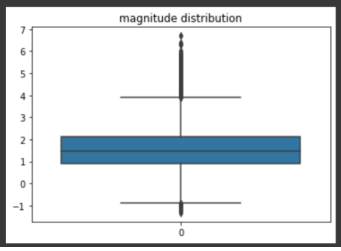


Plot the magnitude and depth into a box plot for visualization.

```
[248] # visualize the data
      sns.boxplot(data=df['depth'])
      plt.title('depth distribution')
      plt.show()
```




```
# visualize the data
sns.boxplot(data=df['mag'])
plt.title('magnitude distribution')
plt.show()
```



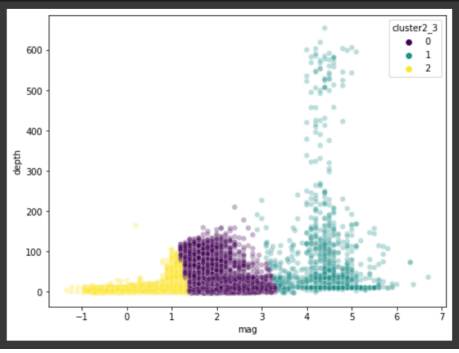
The next thing we did was apply k-means clustering and plot the results into a scatterplot. K-means clustering attempts to group similar types of items into clusters. It detects similarities between items and groups them into clusters.

```
# applying kmean clustering
kMeans = KMeans(n_clusters=3)
kMeans.fit(tempDf[['depth','mag']])
df['cluster2_3'] = kMeans.labels_
df.head()
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	place	type	horizontalError	depthError	magError	magNst	status
0	2022-12-24 02:23:14.097	56.963300	-155.453700	25.900000	2.70	ml	0.0	0.0	0.000	0.38	...	78 km W of Akhiok, Alaska	earthquake	0.00	0.700	0.000	0.0	automatic
1	2022-12-24 02:22:21.110	19.219999	-155.429993	33.400002	1.94	md	28.0	144.0	0.000	0.14	...	5 km ENE of Pāhala, Hawaii	earthquake	0.72	0.840	1.980	4.0	automatic
2	2022-12-24 01:50:43.200	19.248167	-155.395340	31.740000	2.15	ml	41.0	134.0	0.000	0.14	...	10 km ENE of Pāhala, Hawaii	earthquake	0.75	0.740	0.210	6.0	automatic
3	2022-12-24 01:47:09.698	-5.123400	153.304600	40.437000	4.70	mb	23.0	128.0	1.466	0.73	...	New Ireland region, Papua New Guinea	earthquake	12.19	7.840	0.129	18.0	reviewed
4	2022-12-24 01:39:51.776	44.090900	148.181500	57.073000	4.60	mb	45.0	169.0	4.025	0.59	...	Kuril Islands	earthquake	9.06	5.403	0.068	65.0	reviewed

5 rows × 23 columns

```
[251] # ploting result
      plt.figure(figsize = (8,6))
      sns.scatterplot(data = df, y = 'depth', x = 'mag', hue = 'cluster2_3', alpha = 0.3, palette = 'viridis')
      plt.show()
```



Here we are figuring out the cluster summary.

```
# cluster summary
tempList1 = ['depth','mag']
tempList2 = ['earthquakeType','dataAmount','percentage'] + tempList1
eT = 1
tempList3 = []
dflen = len(df)
for i in df['cluster2_3'].unique().tolist():
    tempList4 = []
    tempList4.append(i)
    typelen = len(df.loc[df['cluster2_3']==i])
    tempList4.append(typelen)
    typepercent = round((typelen/dflen)*100,2)
    tempList4.append(typepercent)
    for j in tempList1:
        q25 = str(round(df[j].loc[df['cluster2_3']==i].min(),2))
        q75 = str(round(df[j].loc[df['cluster2_3']==i].max(),2))
        if (q25==q75):
            tempList4.append(q25)
        else:
            tempList4.append("("+q25+") - ("+q75+")")
    tempList3.append(tempList4)
tempdf = pd.DataFrame(tempList3,columns = tempList2)
tempdf
```

	earthquakeType	dataAmount	percentage	depth	mag
0	0	4995	45.63	(-3.74) - (210.1)	(1.2) - (3.34)
1	1	1091	9.97	(-3.74) - (654.24)	(2.9) - (6.7)
2	2	4860	44.40	(-3.25) - (164.4)	(-1.33) - (1.37)

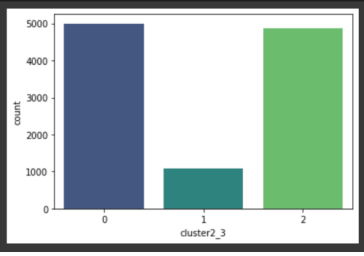
These are the results based on percentage.

```
[253] # result based on percentage
      tempdf.sort_values(by='percentage',ascending = False)
```

	earthquakeType	dataAmount	percentage	depth	mag
0	0	4995	45.63	(-3.74) - (210.1)	(1.2) - (3.34)
2	2	4860	44.40	(-3.25) - (164.4)	(-1.33) - (1.37)
1	1	1091	9.97	(-3.74) - (654.24)	(2.9) - (6.7)

Then we plot the data based on the cluster summary.

```
# plot the data
sns.countplot(data = df , x = 'cluster2_3', palette = 'viridis')
plt.show()
```



We made a geographical visualization for further analysis.

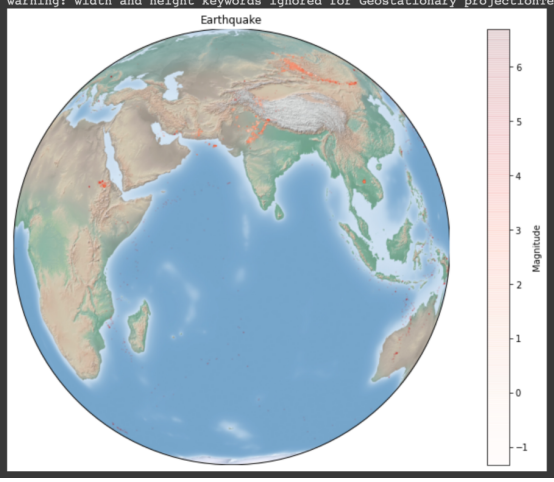
```
[255] # Geographical Visualization
      def background_map():
          plt.figure(figsize=(15,9))
          m = Basemap(projection='geos', resolution=None,width=6E6, height=4E6, lat_0=0, lon_0=70)
          m.shadedrelief(scale=0.7)
          return m

m = background_map()
```

```
warning: width and height keywords ignored for Geostationary projection
```

0s completed at 6:08 PM

```
m = background_map()
m.scatter(df['longitude'], df['latitude'], s=0.5 , c=df['mag'], cmap='Reds', latlon=True, alpha=0.1)
plt.colorbar(label='Magnitude')
plt.title('Earthquake')
```

```
warning: width and height keywords ignored for Geostationary projectionText(0.5, 1.0, 'Earthquake')
```



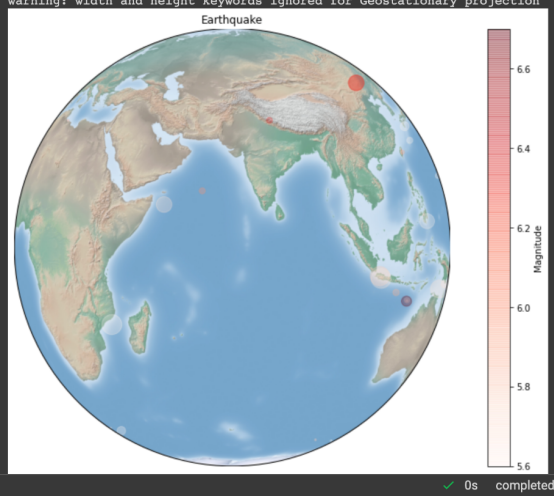

```
[258] high_earthquake = df.loc[df['mag'] > 5.5]
```

```
[259] mms = MinMaxScaler()
```

```
[260] scaled_depth = mms.fit_transform(high_earthquake['depth'].values.reshape(-1,1)).ravel() * 500
```

```
m = background_map()
m.scatter(high_earthquake['longitude'], high_earthquake['latitude'], s=scaled_depth, c=high_earthquake['mag'], cmap='Reds', latlon=True, alpha=0.3)
plt.colorbar(label='Magnitude')
plt.title('Earthquake')
plt.show()
```

```
warning: width and height keywords ignored for Geostationary projection
```



0s completed at 6:08 PM

V. Methodology Evaluation

To begin, split the data into Xs and Ys, which will be the model's input and output, respectively. In this case, the inputs are time, latitude, and longitude, and the outputs are magnitude and depth. Split the Xs and Ys into two groups: train and test with validation. The training dataset contains 80% of the data, while the test dataset contains 20%. We used pd.to_numeric to convert whatever strings in the data into numeric values. Then we replaced the value with 0 instead.

```
[262] # Splitting the Data
      X = df[['time', 'latitude', 'longitude']]
      Y = df[['mag', 'depth']]

[263] from sklearn.linear_model import LinearRegression

      X = X.apply(pd.to_numeric, errors='coerce')
      Y = Y.apply(pd.to_numeric, errors='coerce')

[264] X.fillna(0, inplace=True)
      Y.fillna(0, inplace=True)
```

```
[265] from sklearn.model_selection import train_test_split

      X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state=0)
      clf = LinearRegression().fit(X_train, Y_train)

      X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
      print(X_train.shape, X_test.shape, Y_train.shape, X_test.shape)

      (8756, 3) (2190, 3) (8756, 2) (2190, 3)
```

We used the RandomForestRegressor model to predict the outputs, and we see a strange prediction with a score above 80% that can be assumed to be the best fit but is not due to its predicted values.

```
[266] from sklearn.ensemble import RandomForestRegressor

      reg = RandomForestRegressor(random_state=42)
      reg.fit(X_train, Y_train)
      reg.predict(X_test)

      array([[ 0.761  , 15.968  ],
             [ 0.58625,  0.7502 ],
             [ 0.4954 ,  7.3565 ],
             ...,
             [ 1.365  , 69.022  ],
             [ 4.78   , 10.0513 ],
             [ 2.3058 , 45.71371]])
```

```
[267] reg.score(X_test, Y_test)

      0.8030913165278764
```

```
[268] from sklearn.model_selection import GridSearchCV

      parameters = {'n_estimators':[10, 20, 50, 100, 200, 500]}

      grid_obj = GridSearchCV(reg, parameters)
      grid_fit = grid_obj.fit(X_train, Y_train)
      best_fit = grid_fit.best_estimator_
      best_fit.predict(X_test)

      array([[ 0.7506    , 15.882     ],
             [ 0.63760967,  0.837272  ],
             [ 0.50982   ,  7.28122   ],
             ...,
             [ 1.341     , 69.533     ],
             [ 4.7564    , 11.632276  ],
             [ 2.38566   , 48.58997   ]])
```

```
[269] best_fit.score(X_test, Y_test)

      0.8050404031185664
```

Import SciPy for the basic statistic of this project, we used SciPy stats or also known as ANOVA one-way analysis to obtain probabilistic distributions. And stats f_oneway functions takes the groups as input and returns ANOVA F and p value then prints it.

```
[46] import scipy.stats as stats

     fvalue, pvalue = stats.f_oneway(df['depth'], df['mag'])
     print(fvalue, pvalue)

     2100.487654564308 0.0
```

Installing SHAP package.

```
!pip install shap

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting shap
  Downloading shap-0.41.0-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (575 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 575.9/575.9 KB 28.4 MB/s eta 0:00:00
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.8/dist-packages (from shap) (21.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages (from shap) (1.3.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from shap) (1.7.3)
Collecting slicer==0.0.7
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.8/dist-packages (from shap) (1.0.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from shap) (1.23.5)
Requirement already satisfied: numba in /usr/local/lib/python3.8/dist-packages (from shap) (0.56.4)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.8/dist-packages (from shap) (2.2.0)
Requirement already satisfied: tqdm>4.25.0 in /usr/local/lib/python3.8/dist-packages (from shap) (4.64.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging>20.9->shap) (3.0.9)
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/python3.8/dist-packages (from numba->shap) (0.39.1)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.8/dist-packages (from numba->shap) (6.0.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from numba->shap) (57.4.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas->shap) (2022.7)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->shap) (3.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn->shap) (1.2.0)
Collecting numpy
  Downloading numpy-1.22.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.9/16.9 MB 67.3 MB/s eta 0:00:00
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.7.3->pandas->shap) (1.15.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from importlib-metadata->numba->shap) (3.11.0)
Installing collected packages: slicer, numpy, shap
  Attempting uninstall: numpy
    Found existing installation: numpy 1.23.5
    Uninstalling numpy-1.23.5:
      Successfully uninstalled numpy-1.23.5
Successfully installed numpy-1.22.4 shap-0.41.0 slicer-0.0.7
```

Import the SHAP package and create the explainer. We used the Shapley value to find the average of all the marginal contributions to all possible coalitions of the dataset. The computation time increases exponentially with the number of features in the data. it shows you exactly which features had the most influence on the model's prediction for a single observation.

```
[48] import shap

     explainer = shap.TreeExplainer(reg)
```

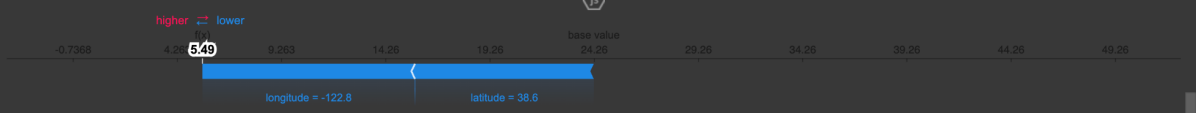
Use the explainer to explain predictions and calculate SHAP values. Choose a random row to test the data out and here we used row 777 where the longitude and latitude are on the lower side then we plot it.

```
df.loc[[777]]
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	place	type	horizontalError	depthError	magError	magNst	status
777	2022-12-21 04:18:46.820	38.601002	-122.797836	4.51	1.22	md	7.0	150.0	0.1141	0.09	...	6km NNE of Windsor, CA	earthquake	1.38	2.65	0.05	5.0	automatic

1 rows × 23 columns

```
[50] choosen_instance = X.loc[[777]]
     shap_values = explainer.shap_values(choosen_instance)
     shap.initjs()
     shap.force_plot(explainer.expected_value[1], shap_values[1], choosen_instance)
```



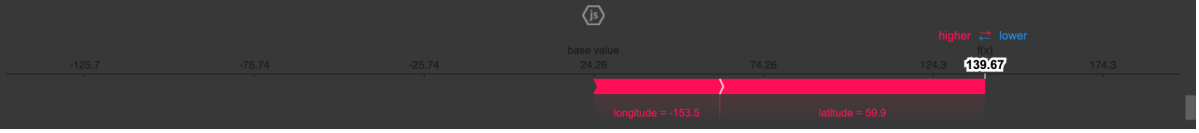
Repeat the previous step once more by using a different row, in this case, we used row 333, where the longitude and latitude are on the higher side, then plot it.

```
[51] df.loc[[333]]
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	place	type	horizontalError	depthError	magError	magNst	status	locat
333	2022-12-22 15:45:16.413	59.896	-153.509	136.7	2.4	ml	0.0	0.0	0.0	0.41	...	35 km ENE of Pedro Bay, Alaska	earthquake	0.0	0.6	0.0	0.0	reviewed	

1 rows × 23 columns

```
choosen_instance = X.loc[[333]]
shap_values = explainer.shap_values(choosen_instance)
shap.initjs()
shap.force_plot(explainer.expected_value[1], shap_values[1], choosen_instance)
```



Printing just the time, latitude, longitude, magnitude and depth of the dataset because that is what we are going to be using for the next few steps.

```
[53] df[['time','latitude','longitude','mag','depth']]
```

	time	latitude	longitude	mag	depth
0	2022-12-24 02:23:14.097	56.963300	-155.453700	2.70	25.900000
1	2022-12-24 02:22:21.110	19.219999	-155.429993	1.94	33.400002
2	2022-12-24 01:50:43.200	19.248167	-155.395340	2.15	31.740000
3	2022-12-24 01:47:09.698	-5.123400	153.304600	4.70	40.437000
4	2022-12-24 01:39:51.776	44.090900	148.181500	4.60	57.073000
...	...	...	...	...	...
10941	2022-11-24 03:00:29.990	37.146667	-121.542333	0.79	3.710000
10942	2022-11-24 02:53:01.020	18.308833	-67.168667	2.49	14.960000
10943	2022-11-24 02:42:08.540	33.758000	-116.917000	0.67	10.620000
10944	2022-11-24 02:35:21.590	38.799333	-122.751167	1.38	1.510000
10945	2022-11-24 02:33:35.197	39.595700	-119.086100	1.80	0.000000

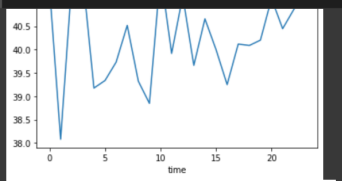
10946 rows × 5 columns

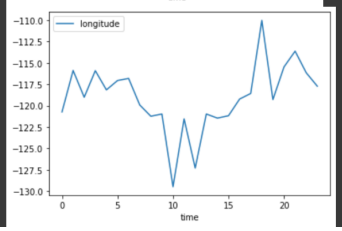
This function converts a scalar, array-like, Series or DataFrame/dict-like to a pandas datetime object. If 'coerce', then invalid parsing will be set as NaT.

```
[112] df['time'] = pd.to_datetime(df['time'], errors='coerce')
```

Plotting the time in hours and finding the mean according to 4 categories which are latitude, longitude, depth and magnitude.

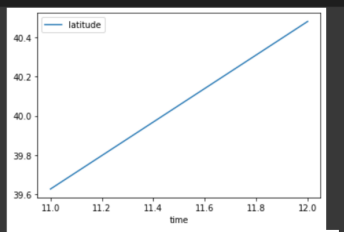
```
[172] groupedByTime = df.groupby([df['time'].dt.hour]).mean()

groupedByTime.plot.line(y='latitude')
groupedByTime.plot.line(y='longitude')
groupedByTime.plot.line(y='depth')
groupedByTime.plot.line(y='mag')
```

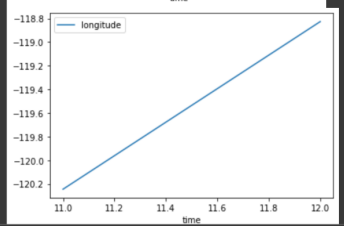


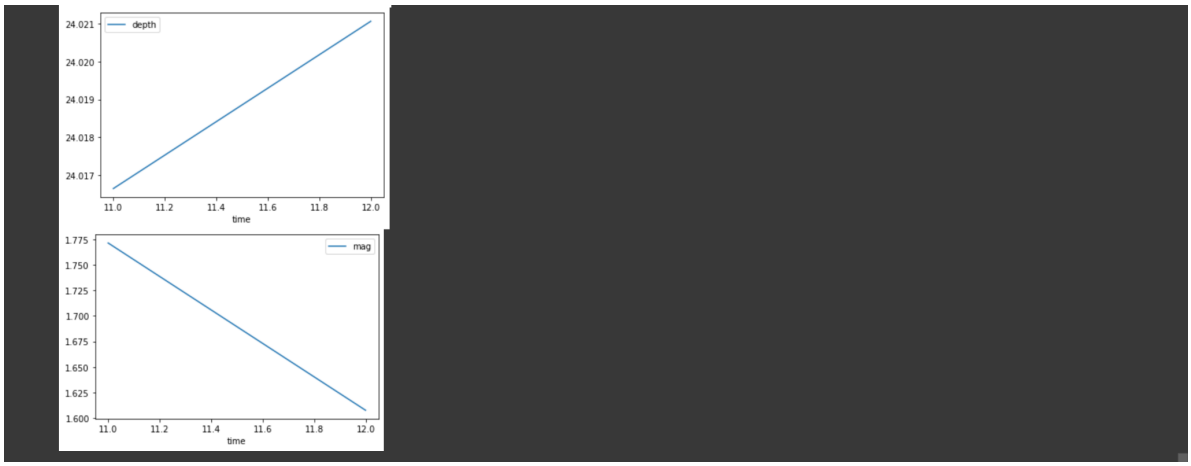


Plotting the time in a month and finding the mean according to 4 categories which are latitude, longitude, depth and magnitude.

```
[174] groupedByTime = df.groupby([df['time'].dt.month]).mean()

groupedByTime.plot.line(y='latitude')
groupedByTime.plot.line(y='longitude')
groupedByTime.plot.line(y='depth')
groupedByTime.plot.line(y='mag')
```







Related Work

((find 15))

[1] Chelidze, T., Kiria, T., Melikadze, G., Jimsheladze, T., & Kobzev, G. (2022). Earthquake Forecast as a Machine Learning Problem for Imbalanced Datasets: Example of Georgia, Caucasus. *Frontiers in Earth Science*, *10*. https://doi.org/10.3389/feart.2022.847808

In this article, they use a machine learning approach for the M>3 earthquake forecasting problem. They train time series on monitoring water level variation. Compared to our dataset, we also use training time but we monitor the earthquake using the magnitude and also the depth of the earthquake itself. For the methodology, they use several methods such as complexity analysis and machine learning in the earthquake forecast. In this method, for the training set, they used the regional seismic catalog and several other predictors. For our datasets, we analyze our data since there is still a lot of missing value, and we need to do data cleaning. This article also uses another method which is earthquake forecast in the laboratory and in numerical models.

[2] Xiangrong, X. (2020). Visual Analysis of World Earthquakes based on Data Science and Statistical Methods. *Journal of Physics: Conference Series*, *1684*(1), 012031. https://doi.org/10.1088/1742-6596/1684/1/012031

In this article, they talked about how earthquakes are severe geological disaster that frequently results in significant casualties as we mentioned in our problem analysis and as a result, it is critical to investigate the law of earthquake occurrence and implement appropriate mitigation measures in order to reduce losses and casualties. In this study, they used data science methods and tools to investigate the frequency of earthquake occurrence which is similar to ours as we are trying to find how much of an impact do the categories of the earthquake does. This paper that they made is used as a practice of enforcing the workflow of data science, which uses computer algorithms to generate fast data statistics and conduct visualization processing to investigate the law of earthquake occurrence.

[3] Tocheport, A., Rivera, L., & Chevrot, S. (2007). A systematic study of source time functions and moment tensors of intermediate and deep earthquakes. *Journal of Geophysical Research*, *112*(B7). https://doi.org/10.1029/2006jb004534

In this article, they developed an inversion algorithm to determine the Source Time Function and Moment Tensor of intermediate and deep earthquakes from teleseismic body wave records. They estimated the complete moment tensor solution, the pure deviatoric solution, and the double-couple solution using three different inversions. All of the calculations are extremely simple, and it is especially unnecessary to compute synthetic seismograms. The method necessitates well-isolated phases at various stations, limiting its application to intermediate and deep events. The algorithm is applied to FDSN broadband records of worldwide intermediate and deep seismicity (depth [100 km] of magnitude greater than 6.5) from 1990 to 2005. The source time functions are compared to other studies of intermediate and deep events. They tried to find an algorithm that detects an upcoming earthquake while ours determine the effects that it has on the earthquake.

[4] Kurata, N., & Ise, T. (2022). Thematic and Country-Specific Characteristics of Research on the Great East Japan Earthquake: An Analysis Using Data Science Methods. *Open Journal of Social Sciences*, *10*(11), 244–256. https://doi.org/10.4236/jss.2022.1011017

In this article, the Great East Japan Earthquake of 2011 had far-reaching consequences in a variety of ways because it was a complex disaster which we briefly talked about when we compared our dataset to the Japanese dataset. In addition to the earthquake, the tsunami and nuclear accident had far-reaching consequences for human lives, health, the economy, and the environment. In this study, they used data science as well to examine over 20,000 academic records about the Great East Japan Earthquake, while we did not use academic records, the aim of their research is somewhat similar to ours as they are also trying to examine the earthquakes that happen in Japan. The characteristics of many research fields have been revealed as a result of text mining. They discovered characteristics of countries that conducted disaster studies by collecting studies by country and research subject. They discovered that countries in the same Asian region as Japan, as well as countries prone to earthquakes and tsunamis, have a high level of research interest.

[5] Mousavi, S. M., Sheng, Y., Zhu, W., & Beroza, G. C. (2019). STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI. *IEEE Access*, *7*, 179464–179476. https://doi.org/10.1109/access.2019.2947848

Understanding the properties of earthquakes using the machine learning technique helps people to understand the relationship and cover the data pattern. They are using the Standford earthquake dataset for their research on the seismic problem. A problem and the subject of active research is the effective extraction of as much valuable information as possible from the recorded signals and the possibility of acquiring fresh insight. What is interesting here, they are using their own data set which is basically a large-scale global labelled data set of an earthquake and non-earthquake signals recorded by seismic instruments.

[6] Johnson, P. A., Rouet-Leduc, B., Pyrak-Nolte, L. J., Beroza, G. C., Marone, C. J., Hulbert, C., Howard, A., Singer, P., Gordeev, D., Karaflos, D., Levinson, C. J., Pfeiffer, P., Puk, K. M., & Reade, W. (2021). Laboratory earthquake forecasting: A machine learning competition. *Proceedings of the National Academy of Sciences*, *118*(5). https://doi.org/10.1073/pnas.2011362118

The researchers of this paper used Kaggle, Google's machine learning competition platform, to engage the global ML community in a competition to develop and improve data analysis approaches for a forecasting problem involving laboratory earthquake data, their aim is to forecast earthquakes while ours is to see if anything effects the earthquakes after it has happened. Based on only a small portion of the laboratory seismic data, the competitors were tasked with predicting the time remaining before the next earthquake of successive laboratory quake events. Over 400 computer programs were created and shared by the more than 4,500 participating teams in openly accessible notebooks. The winning teams used unexpected strategies based on rescaling failure times as a fraction of the seismic cycle and comparing input distribution of training and testing data, in addition to the now well-known features of seismic data that map to fault criticality in the laboratory. The competition serves as a pedagogical tool for teaching ML in geophysics, in addition to yielding scientific insights into fault processes in the laboratory and their

relationships with the evolution of the statistical properties of the associated seismic data. The approach could serve as a model for other competitions in geosciences or other fields of study to help engage the ML community in important problems.

[7] Dey, B., Dikshit, P., Sehgal, S., Trehan, V., & Kumar Sehgal, V. (2022). Intelligent solutions for earthquake data analysis and prediction for future smart cities. *Computers & Industrial Engineering*, *170*, 108368. https://doi.org/10.1016/j.cie.2022.108368

The analysis and prediction of earthquakes for smart cities are critical because all critical infrastructure, such as drinking water resources, mobile networks, healthcare, power grid, and transportation, can affect the social-economic balance of a place that has been impacted by natural disasters. Earthquake Environmental Effects are the effects that an earthquake has on the environment as a result of the sudden movement of tectonic plates (EEE). The main contribution of this paper revolves around earthquakes and their analysis. Earthquakes are caused by the movement of tectonic plates on the lithosphere. These movements occur as a result of frictional stress between gliding plate borders, which leads to fault line failure. The sudden shaking of the plates creates seismic waves by releasing a specific type of energy in the Lithosphere. Earthquakes are also classified into different classes based on their magnitude, ranging from minor to major. They went in depth with earthquakes are their different classes and also the impact earthquakes have towards the environment meanwhile we just touched upon it briefly and focused on the data analyzing and finding out whether or not then depth or magnitude effects earthquakes in any way.

[8] Murti, M. A., Junior, R., Ahmed, A. N., & Elshafie, A. (2022). Earthquake multi-classification detection based velocity and displacement data filtering using machine learning algorithms. *Scientific Reports*, *12*(1), 21200. https://doi.org/10.1038/s41598-022-25098-1

This article analyzes and predicts earthquakes for smart cities because all critical infrastructure, such as drinking water resources, mobile networks, healthcare, power grid, and transportation, can affect the social-economic balance of a place that has been impacted by natural disasters. Earthquake Environmental Effects are the effects that an earthquake has on the environment as a result of the sudden movement of tectonic plates (EEE). The main contribution of this paper revolves around earthquakes and their analysis. Earthquakes are caused by the movement of tectonic plates on the lithosphere. This research paper is similar to the one above [7] where they focused more on the environmental aspect while we focused on data analyzing of what effects the earthquake.

[9]K. M. Asim, F. Martínez-Álvarez, A. Basit, and T. Iqbal, "Earthquake magnitude prediction in Hindukush region using machine learning techniques," *Natural Hazards*, vol. 85, no. 1, pp. 471–486, Sep. 2016, doi: 10.1007/s11069-016-2579-3.

This article is focusing on earthquake magnitude prediction specifically in the Hindukush region using the machine learning technique. Eight seismic indicators that were mathematically derived from the local earthquake database have been used to make predictions. Compared to our research, our prediction is taken from the data visualization we make. For the machine learning technique, they are using 4 different types of machine learning techniques which are, pattern recognition neural network, recurrent

neural network, random forest and linear programming boost ensemble classifier. This is quite different compared to our work because, for the machine learning technique, we are using unsupervised learning which is clustering.

[10]S. Mangalathu, H. Sun, C. C. Nweke, Z. Yi, and H. V. Burton, "Classifying earthquake damage to buildings using machine learning," *Earthquake Spectra*, vol. 36, no. 1, pp. 183–208, Jan. 2020, doi: 10.1177/8755293019878137.

For this research article, Individual building damage must be visually identified and classified, which can take a lot of time and people resources and continue for months after the incident. They are using machine learning techniques such as discriminant analysis, *k*-nearest neighbours, decision trees, and random forests, to rapidly predict earthquake-induced building damage. If we compare using our research, there are some similarities in the machine learning technique used. For example the random forest, we are using a random forest model to predict the outputs.

[11]G. L. Mao, T. P. Ferrand, J. Li, B. Zhu, Z. Xi, and M. Chen, "Unsupervised machine learning reveals slab hydration variations from deep earthquake distributions beneath the northwest Pacific," *Communications Earth & Environment*, vol. 3, no. 1, pp. 1–9, Mar. 2022, doi: 10.1038/s43247-022-00377-x.

The comprehensive knowledge of these systems is hindered by detection constraints. They estimate the b values of deep earthquakes in the northwest Pacific Plate, clustered in four regions, using the Japan Meteorological Agency (JMA) database. JMA is expected to deliver accurate and timely information to governmental organizations and citizens with the objectives of preventing and mitigating natural disasters as the sole national body in charge of issuing weather/tsunami warnings and advisories. In this article, they are using clustering for the region, and each cluster is correlated with different slab hydration states controlled by oceanic plate features and fabrics orientation and distribution. This article is working using the depth of the earthquake which is also similar to our work. We also use depth to finish our work. We also attempt to group similar types of items into clusters. It detects similarities between items and groups them into clusters.

[12] S. M. Mousavi and G. C. Beroza, "A Machine‐Learning Approach for Earthquake Magnitude Estimation," *Geophysical Research Letters*, vol. 47, no. 1, Jan. 2020, doi: 10.1029/2019gl085976.

In this article convolutional and recurrent neural networks were used to create a regressor (MagNet) that is not sensitive to data normalization, allowing waveform amplitude information to be used during training. Direct learning of site- and distance-dependent functions from training data is possible for neural networks. They tested their network on both local and duration magnitude scales.

Evaluation Method

The purpose of our research is to determine whether the magnitude or depth will affect earthquakes in any category since the magnitude and depth of an earthquake take an important part in a natural disaster. And finding out whether or not the magnitude or depth effects earthquakes at all is important because magnitude and depth are the two basic features of an earthquake that is crucial in understanding plate tectonics as well as the earthquake's potential hazards. We managed to come to the conclusion that the magnitude and depth do not effect earthquakes in a major way as we have elaborated in this research paper.

Results and Discussion

Understanding plate tectonics and earthquake danger requires a fundamental understanding of an earthquake's magnitude and depth. The potential for destruction increases with the magnitude and shallowness of the earthquake. From the methods shown before, we can see that The intensity of an earthquake increases with its Richter scale magnitude, which also increases the amount of damage. The strength of shaking from an earthquake diminishes with increasing distance from the earthquake's source.

Conclusion and Recommendation

From the methodology that has been done for this research, we can conclude that the null hypothesis is not true, because the magnitude and depth did have an effect on an earthquake. For the recommendation for further work, we would plot the tectonic plates so we can see the movements since movements are mostly seen in the tectonic boundaries and see the correlation between earthquakes and tectonic plates.

References

[1]F. Zulfikar, “10 Negara Paling Rawan Gempa Bumi, Indonesia Termasuk?,” *detikedu*. https://www.detik.com/edu/detikpedia/d-6361792/10-negara-paling-rawan-gempa-bumi-indonesia-termasuk (accessed Jan. 15, 2023).

[2]“Magnitude Types | U.S. Geological Survey,” *www.usgs.gov*. https://www.usgs.gov/programs/earthquake-hazards/magnitude-types

[3]“geospatial-data,” *colorado.posit.co*. https://colorado.posit.co/rsc/jupyter-geospatial/geospatial-data.html (accessed Jan. 15, 2023).

[4] P. Borman, “Earthquake, Magnitude,” *Encyclopedia of Solid Earth Geophysics*, pp. 243–254, 2021, doi: 10.1007/978-3-030-58631-7_3.

[5]I. M. Korrat, A. Lethy, M. N. ElGabry, H. M. Hussein, and A. S. Othman, “Discrimination Between Small Earthquakes and Quarry Blasts in Egypt Using Spectral Source Characteristics,” *Pure and Applied Geophysics*, vol. 179, no. 2, pp. 599–618, Jan. 2022, doi: 10.1007/s00024-022-02953-w.

[6] Administrator, & Administrator. (2021, February 13). *Earthquakes Research Paper*. IResearchNet. https://www.iresearchnet.com/research-paper-examples/history-research-paper/earthquakes-research-paper/

[7] Wisher, B., & Lomnitz, C. (2023). *Earthquakes*. ResearchGate; ResearchGate. https://www.researchgate.net/publication/288969202_Earthquakes

Github

https://github.com/raissaazaria/FoDS_FP