# EARTHQUAKE PREDICTION MODEL USING MACHINE LEARNING-PYTHON

## Objective:

An earthquake is the sudden release of strain energy in the Earth's crust, resulting in waves of shaking that radiate outwards from the earthquake source.

## Problem Statement :

- An earthquake is a violent and abrupt shaking of the ground, caused by movement between tectonic plates along a fault line in the earth's crust.
- When the stored energy beneath the crust is suddenly released as an earthquake, the crust's response to the changing stress beneath it is not directly proportional.

- An earthquake prediction must define 3 elements:
  1) the date and time
  2) the location
  3) the magnitude
- In order to anticipate earthquakes, machine learning may be used to examine seismic data trends.
- Earthquakes can result in the ground shaking, soil liquefaction, landslides, fissures, avalanches, fires and tsunamis.
- The environmental effects of it are that including surface faulting, tectonic uplift and subsidence, tsunamis, soil liquefaction, ground resonance, landslides and ground failure, either directly linked to a quake source or provoked by the ground shaking.
- The main problem occur in earthquake occurs loss of natural,human and environment surrounding.It leads to disruption in economic activities.
- Seismologists study earthquakes by looking at the damage that was caused and by using seismometers. A seismometer is an instrument that records the shaking of the Earth’s surface caused by seismic waves. The term seismograph usually refers to the combined seismometer and recording device.

## Design thinking:

Earthquakes were once thought to result from supernatural forces in the prehistoric era. Aristotle was the first to identify earthquakes as a natural occurrence and to provide some potential explanations for them in a truly scientific manner. One of nature’s most destructive dangers is earthquakes. Strong earthquakes frequently have negative effects.

A lot of devastating earthquakes occasionally occur in nations like Japan, the USA, China, and nations in the middle and far east. Several major and medium-sized earthquakes have also occurred in India, which have resulted in significant property damage and fatalities. One of the most catastrophic earthquakes ever recorded occurred in Maharastra early on September 30, 1993. One of the main goals of researchers studying earthquake seismology is to develop effective predicting methods for the occurrence of the next severe earthquake event that may allow us to reduce the death toll and property damage.

STEPS TO IMPLEMENT:

- ✓ First create a dataset using kaggle website
- ✓ From 1990-2023 dataset
- ✓ https://www.kaggle.com/datasets/alessandrolobello/the-ultimate-earthquake-dataset-from-1990-2023
- ✓ I will start this task to create a model for earthquake prediction by importing the necessary python libraries:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt
```

✓ Now let’s load and read the dataset.

✓ data = pd.read_csv("database.csv")

✓ data.columns

✓ Index([‘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’],

✓ Dtype=’object’)

✓ Now let’s see the main characteristics of earthquake data and create an object of these characteristics, namely, date, time, latitude, longitude, depth, magnitude:

✓

✓ Data = data[[‘Date’, ‘Time’, ‘Latitude’, ‘Longitude’, ‘Depth’, ‘Magnitude’]]

✓ Data.head()

| ✓ date | Time | Latitude | Longitude | Depth | Magnitude | |
|---|---|---|---|---|---|---|
| **0** | 01/02/1965 | 13:44:18 | 19.246 | 145.616 | 131.6 | 6 |
| **1** | 01/04/1965 | 11:29:49 | 1.863 | 127.352 | 80.0 | 5 |
| **2** | 01/05/1965 | 18:05:58 | -20.579 | -173.972 | 20.0 | 6 |
| **3** | 01/08/1965 | 18:49:43 | -59.076 | -23.557 | 15.0 | 5 |
| **4** | 01/09/1965 | 13:32:50 | 11.938 | 126.427 | 15.0 | 5 |

Since the data is random, so we need to scale it based on the model inputs. In this, we convert the given date and time to Unix time which is in seconds and a number. This can be easily used as an entry for the network we have built:

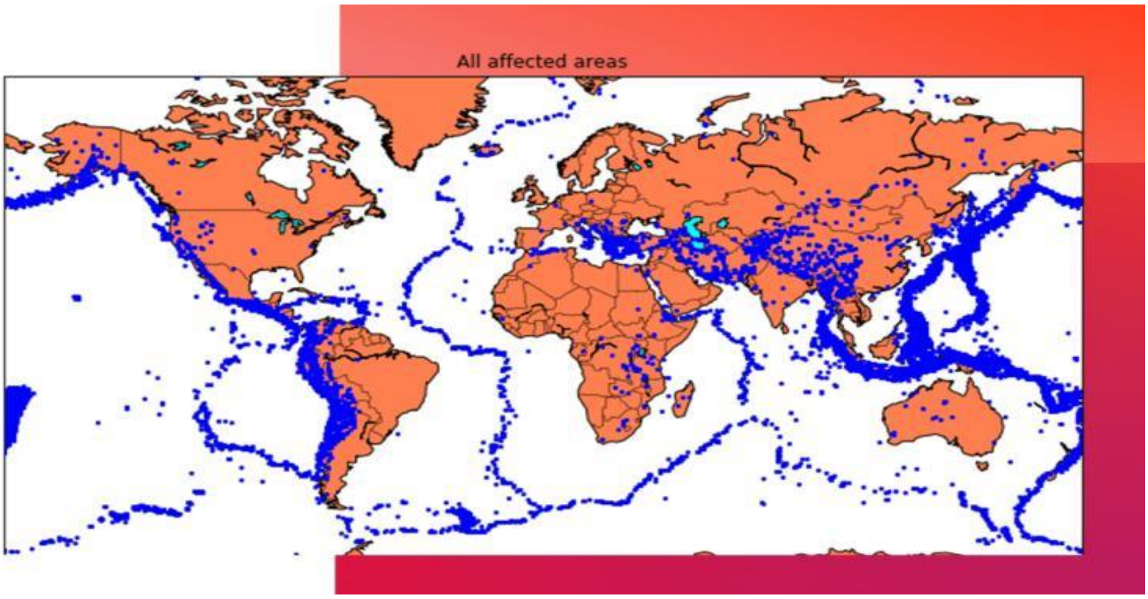
```
import datetime
import time


timestamp = []
for d, t in zip(data['Date'], data['Time']):
    try:
        ts = datetime.datetime.strptime(d+' '+t, '%m/%d/%Y %H:%M:%S')
```

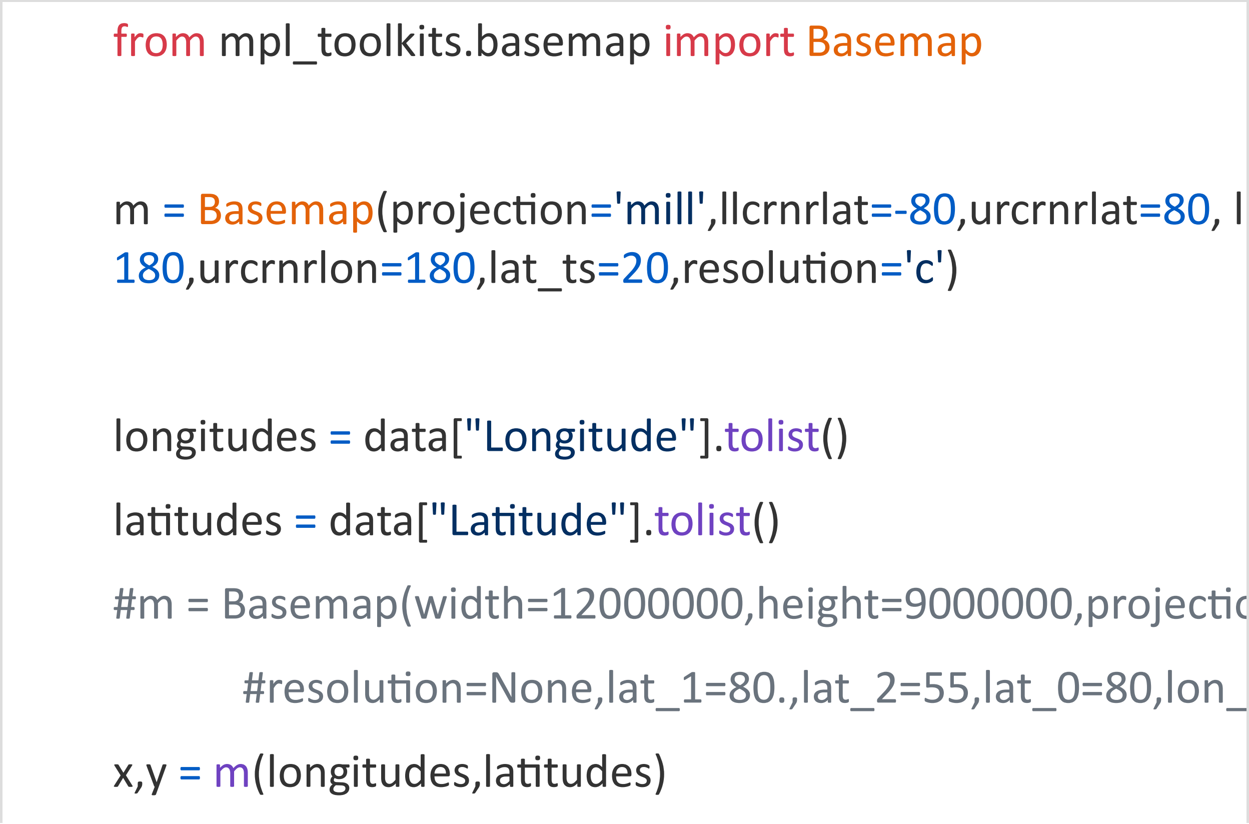
```
timestamp.append(time.mktime(ts.timetuple()))
    except ValueError:
        # print('ValueError')
        timestamp.append('ValueError')
timeStamp = pd.Series(timestamp)
data['Timestamp'] = timeStamp.values
final_data = data.drop(['Date', 'Time'], axis=1)
final_data = final_data[final_data.Timestamp != 'ValueError']
final_data.head()
```

| | Latitude | Longitude | Depth | Magnitude | Timestamp |
|---|---|---|---|---|---|
| **0** | 19.246 | 145.616 | 131.6 | 6.0 | -1.57631e+08 |
| **1** | 1.863 | 127.352 | 80.0 | 5.8 | -1.57466e+08 |
| **2** | -20.579 | -173.972 | 20.0 | 6.2 | -1.57356e+08 |
| **3** | -59.076 | -23.557 | 15.0 | 5.8 | -1.57094e+08 |
| **4** | 11.938 | 126.427 | 15.0 | 5.8 | -1.57026e+08 |

## Data visualation:



Now, before we create the earthquake prediction model, let’s visualize the data on a world map that shows a clear representation of where the earthquake frequency will be more:



```
from mpl_toolkits.basemap import Basemap


m = Basemap(projection='mill',llcrnrlat=-80,urcrnrlat=80, llcrnrlon=
180,urcrnrlon=180,lat_ts=20,resolution='c')


longitudes = data["Longitude"].tolist()

latitudes = data["Latitude"].tolist()

#m = Basemap(width=12000000,height=9000000,projection='lcc',

      #resolution=None,lat_1=80.,lat_2=55,lat_0=80,lon_0=-107.)

x,y = m(longitudes,latitudes)
```

```
fig = plt.figure(figsize=(12,10))
plt.title("All affected areas")
m.plot(x, y, "o", markersize = 2, color = 'blue')
m.drawcoastlines()
m.fillcontinents(color='coral',lake_color='aqua')
m.drawmapboundary()
m.drawcountries()
plt.show()
```

## Data splitting:

Now, to create the earthquake prediction model, we need to divide the data into Xs and ys which respectively will be entered into the model as inputs to receive the output from the model.

Here the inputs are TImestamp, Latitude and Longitude and the outputs are Magnitude and Depth. I’m going to split the xs and ys into train and test with validation. The training set contains 80% and the test set contains 20%:

```
X = final_data[['Timestamp', 'Latitude', 'Longitude']]
y = final_data[['Magnitude', 'Depth']]
from sklearn.cross_validation import train_test_split
```

```
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
```

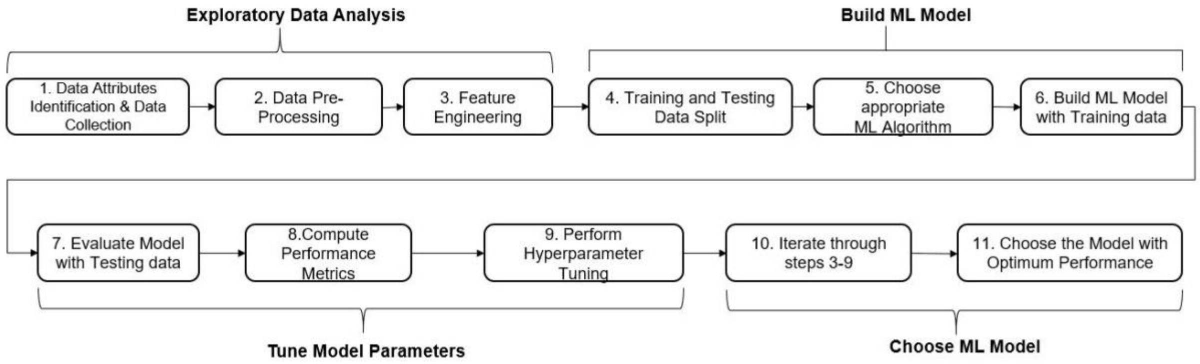
## Data Exploration:

A Machine Learning project is as good as the foundation of data on which it is built. In order to perform well, machine learning data exploration models must ingest large quantities of data, and model accuracy will suffer if that data is not thoroughly explored first. Data exploration steps to follow before building a machine learning model include:

- Variable identification: define each variable and its role in the dataset
- Univariate analysis: for continuous variables, build box plots or histograms for each variable independently; for categorical variables, build bar charts to show the frequencies
- Bi-variable analysis - determine the interaction between variables by building visualization tools
- ~Continuous and Continuous: scatter plots
- ~Categorical and Categorical: stacked column chart
- ~Categorical and Continuous: boxplots combined with swarmplots
- Detect and treat missing values
- Detect and treat outliers

## Model development:

**Building an ML Model** requires splitting of data into two sets, such as ‘training set’ and ‘testing set’ in the ratio of 80:20 or


Exploratory Data Analysis
Build ML Model
1. Data Attributes Identification & Data Collection
2. Data Pre-Processing
3. Feature Engineering
4. Training and Testing Data Split
5. Choose appropriate ML Algorithm
6. Build ML Model with Training data
7. Evaluate Model with Testing data
8.Compute Performance Metrics
9. Perform Hyperparameter Tuning
10. Iterate through steps 3-9
11. Choose the Model with Optimum Performance
Tune Model Parameters
Choose ML Model

70:30; A set of supervised (for labelled data) and unsupervised (for unlabeled data) algorithms are available to choose from depending on the nature of input data and business outcome to predict. In case of labelled data, it is recommended to choose logistic regression algorithm if the outcome to predict is of binary (0/1) in nature; choose decision tree classifier (or) Random Forest® classifier (or) KNN if the outcome to predict is of multi-class (1/2/3/...) in nature; and choose linear regression algorithm (e.g., decision tree regressor, random forest regressor) if the outcome to predict is continuous in nature. The clustering (Unsupervised) algorithm is preferred to analyze unlabeled / unstructured (text) data (e.g., k-means clustering). Artificial Neural Network (ANN) algorithms are suggested to analyze other unstructured (image/voice) data types, such as Convolutional Neural Networks (CNN) for image recognition and Recurrent Neural Networks (RNN) for voice recognition and Natural Language Processing (NLP). The model is built using training dataset and make prediction using test dataset. Use of deep learning (neural networks) models is preferred over regression models (ML models) for better performance as these models introduce extra layer of non-linearity with the introduction of Activation Function (AF).