**COURSE: ARTIFICIAL INTELLIGENCE**

**TITLE: EARTHQUAKE PREDICTION MODEL USING PYTHON**

**PHASE 1 SUBMISSION**

**Team members:**

**Shravya. G. P (au 311521106091)**

**Sakthi Sowmiya. S (au 311521106083)**

**Rekha Sekar (au 311521106075)**

**Saali Deivanai.L (au 311521106080)**

**PROBLEM DEFINITION:**

Predicting earthquakes using Python, or any other programming language, is a challenging scientific problem because short-term earthquake prediction remains uncertain and not currently feasible with high accuracy. Earthquake prediction often involves making forecasts about the occurrence, timing, and location of future seismic events, which is a complex task due to the unpredictable nature of earthquakes. Nevertheless, researchers use Python and various data analysis and machine learning techniques to work on related tasks like earthquake forecasting and seismic hazard assessment.

**DESIGN THINKING:**

**Earthquake Prediction Using Python:**

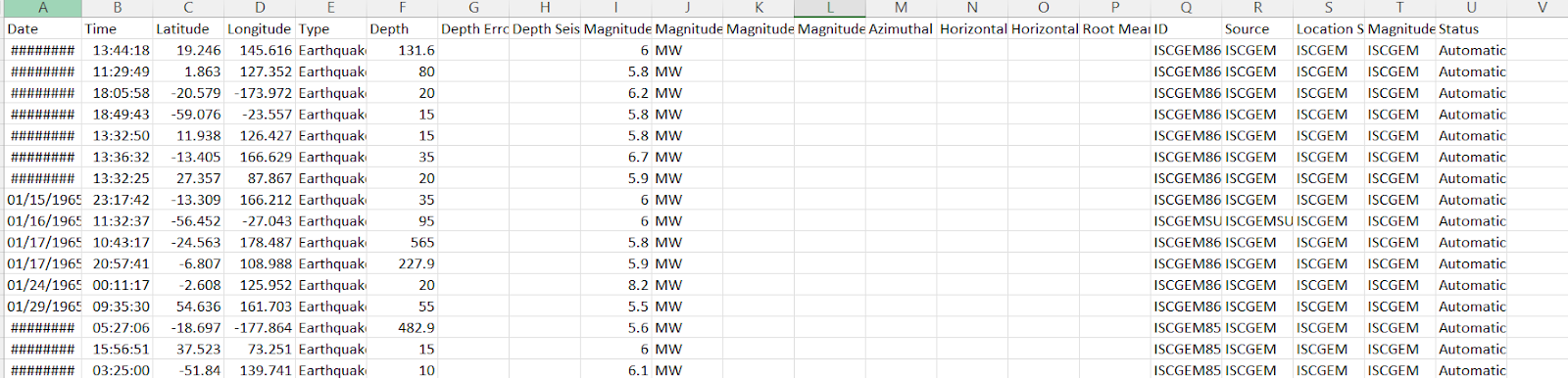
Earthquake prediction using Python refers to the application of Python programming and associated libraries and tools to analyze historical earthquake data, study seismic patterns, and assess long-term seismic hazards. While precise short-term earthquake prediction is a challenging goal, Python can be instrumental in various aspects of earthquake research, including:

* **Data Analysis:** Using Python libraries like Pandas, NumPy, and Matplotlib to clean, process, and visualize earthquake data for gaining insights into seismic events.
* **Machine Learning:** Employing machine learning algorithms, implemented in Python with libraries like Scikit-Learn or TensorFlow, to develop models for earthquake forecasting, aftershock prediction, or seismic hazard assessment.
* **Geospatial Analysis**: Utilizing Python libraries such as GeoPandas, Basemap, or Folium to perform geospatial analysis, visualize earthquake data on maps, and identify regions with high seismic risk.
* **Statistical Analysis**: Conducting statistical analyses in Python to identify patterns, correlations, and trends in earthquake occurrence and characteristics.
* Simulation: Using Python for physics-based simulations of seismic events and understanding how earthquakes propagate through geological structures.
* **Early Warning Systems**: Developing early warning systems in Python that provide real-time alerts and notifications to minimize the impact of earthquakes.
* Research and Education: Collaborating with researchers, educators, and institutions to contribute to earthquake science and education by sharing findings, creating educational materials, and promoting earthquake preparedness.

It's essential to emphasize that while Python is a valuable tool for earthquake-related research and analysis, the primary goal is often to better understand earthquake behavior, assess long-term seismic hazards, and improve earthquake preparedness, rather than making short-term earthquake predictions. Short-term earthquake prediction remains a complex and uncertain field of study in seismology.

**DATA SET:**

**DATA SET LINK:** [**https://www.kaggle.com/datasets/usgs/earthquake-database**](https://www.kaggle.com/datasets/usgs/earthquake-database)



**PROGRAM:**

from learntools.core import binder

binder.bind(globals())

from learntools.data\_cleaning.ex3 import \*

print("Setup Complete")

# modules we'll use

import pandas as pd

import numpy as np

import seaborn as sns

import datetime

# read in our data

earthquakes = pd.read\_csv("../input/earthquake-database/database.csv")

# set seed for reproducibility

np.random.seed(0)

# TODO: Your code here!

earthquakes['Date'].head()

# Check your answer (Run this code cell to receive credit!)

q1.check()

# Line below will give you a hint

#q1.hint()

Earthquakes[3378:3383]

date\_lengths = earthquakes.Date.str.len()

date\_lengths.value\_counts()

indices = np.where([date\_lengths == 24])[1]

print('Indices with corrupted data:', indices)

earthquakes.loc[indices]

# TODO: Your code here

earthquakes.loc[3378, "Date"] = "02/23/1975"

earthquakes.loc[7512, "Date"] = "04/28/1985"

earthquakes.loc[20650, "Date"] = "03/13/2011"

earthquakes['date\_parsed'] = pd.to\_datetime(earthquakes['Date'], format="%m/%d/%Y")

# Check your answer

q2.check()

# try to get the day of the month from the date column

day\_of\_month\_earthquakes = earthquakes['date\_parsed'].dt.day

# Check your answer

q3.check()

# TODO: Your code here!

sns.distplot(day\_of\_month\_earthquakes, kde=False, bins=31)

# Check your answer (Run this code cell to receive credit!)

q4.check()

**OUTPUT:**

0    01/02/1965

1    01/04/1965

2    01/05/1965

3    01/08/1965

4    01/09/1965

Date

10    23409

24        3

Name: count, dtype: int64

