**Earthquake prediction using python**

PHASE-3

Team 2

Team Members:

S.Premlatha(312821205030)

V.Keerthana(312821205014)

S. Ashika jubi(312821205003

R. Aravind(312821205001)

M. venu Aravind(312821205048)

Overview

**Phase 1:** This project aims to predict the magnitude and probability of earthquake occurring in a particular region using the data with various machine learning models to find which model is more accurate to accomplish this task.

**Phase2:** Our goal is to explain what are the steps involved in basic earthquake prediction model.

**Phase-3**

**Objectives:**

* To discover the areas of the world mostly affected by earthquakes
* To identify patterns in the timeline of earthquakes
* To determine if the year 2022 was significantly different to the previous years.

## Areas covered:

* Structuring & Cleaning data
* Plotting
* Scatter maps
* Hypothesis Testing

**import numpy as np**

**import pandas as pd**

In [3]:

**import os;**

os.listdir('/kaggle/input/earthquake-dataset')

Out[3]:

['earthquake\_data.csv', 'earthquake\_1995-2023.csv']

In [4]:

df = os.listdir('/kaggle/input/earthquake-dataset')

In [5]:

df = pd.read\_csv('../input/earthquake-dataset/earthquake\_data.csv')

In [6]:

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 782 entries, 0 to 781

**Data columns (total 19 columns):**

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 title 782 non-null object

1 magnitude 782 non-null float64

2 date\_time 782 non-null object

3 cdi 782 non-null int64

4 mmi 782 non-null int64

5 alert 415 non-null object

6 tsunami 782 non-null int64

7 sig 782 non-null int64

8 net 782 non-null object

9 nst 782 non-null int64

10 dmin 782 non-null float64

11 gap 782 non-null float64

12 magType 782 non-null object

13 depth 782 non-null float64

14 latitude 782 non-null float64

15 longitude 782 non-null float64

16 location 777 non-null object

17 continent 206 non-null object

18 country 484 non-null object

dtypes: float64(6), int64(5), object(8)

memory usage: 116.2+ KB

**df.shape**

Out[7]:

(782, 19)

In [8]:

df.dtypes

Out[8]:

title object

magnitude float64

date\_time object

cdi int64

mmi int64

alert object

tsunami int64

sig int64

net object

nst int64

dmin float64

gap float64

magType object

depth float64

latitude float64

longitude float64

location object

continent object

country object

**dtype**: object

***df.info()***

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 782 entries, 0 to 781

**Data columns** (total 19 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 title 782 non-null object

1 magnitude 782 non-null float64

2 date\_time 782 non-null object

3 cdi 782 non-null int64

4 mmi 782 non-null int64

5 alert 415 non-null object

6 tsunami 782 non-null int64

7 sig 782 non-null int64

8 net 782 non-null object

9 nst 782 non-null int64

10 dmin 782 non-null float64

11 gap 782 non-null float64

12 magType 782 non-null object

13 depth 782 non-null float64

14 latitude 782 non-null float64

15 longitude 782 non-null float64

16 location 777 non-null object

17 continent 206 non-null object

18 country 484 non-null object

dtypes: float64(6), int64(5), object(8)

* memory usage: 116.2+ KB

**df. isnull():**

df.duplicated()

Out[11]:

0 False

1 False

2 False

3 False

4 False

...

777 False

778 False

779 False

780 False

781 False

Length: 782, dtype: bool

In [12]:

df.duplicated().sum()

**df.isnull().sum()**

Out[14]:

title 0

magnitude 0

date\_time 0

cdi 0

mmi 0

alert 367

tsunami 0

sig 0

net 0

nst 0

dmin 0

gap 0

magType 0

depth 0

latitude 0

longitude 0

location 5

continent 576

country 298

dtype: int64

Null fields in 'alert' --> will replace null with 'Unknown'  
'location'  
'continent'  
'country'  
--> will not use these 3 fields directly

## Duplicate and null report

**df.duplicated()**

Out[11]:

0 False

1 False

2 False

3 False

4 False

...

777 False

778 False

779 False

780 False

781 False

Length: 782, dtype: bool

In [12]:

**df.duplicated().sum()**

**Date time transformations:**

df['date\_time'] = pd.to\_datetime(df['date\_time'], dayfirst=True)

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 782 entries, 0 to 781

Data columns (total 19 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 title 782 non-null object

1 magnitude 782 non-null float64

2 date\_time 782 non-null datetime64[ns]

3 cdi 782 non-null int64

4 mmi 782 non-null int64

5 alert 782 non-null object

6 tsunami 782 non-null int64

7 sig 782 non-null int64

8 net 782 non-null object

9 nst 782 non-null int64

10 dmin 782 non-null float64

11 gap 782 non-null float64

12 magType 782 non-null object

13 depth 782 non-null float64

14 latitude 782 non-null float64

15 longitude 782 non-null float64

16 location 777 non-null object

17 continent 206 non-null object

18 country 484 non-null object

dtypes: datetime64[ns](1), float64(6), int64(5), object(7)

memory usage: 116.2+ KB

## Data Visualisation

In [21]:

import seaborn as sns

import matplotlib.pyplot as plt

*# finding the number of earthquakes per country, to determine the most seismogenic countries:*

countries = df['country'].value\_counts()

print(countries)

country

Indonesia 110

Papua New Guinea 56

Chile 34

Vanuatu 27

Solomon Islands 22

Japan 21

Mexico 20

Peru 20

Philippines 17

United States of America 17

Russia 15

People's Republic of China 12

Fiji 9

New Zealand 9

Afghanistan 6

Taiwan 6

Ecuador 6

Myanmar 5

United Kingdom of Great Britain and Northern Ireland (the) 5

Iran 5

India 5

Greece 5

Pakistan 4

Colombia 4

Panama 4

Nepal 4

Turkey 4

Brazil 3

Bolivia 3

Costa Rica 3

Antarctica 3

Argentina 2

Haiti 2

Kyrgyzstan 1

Martinique 1

Mozambique 1

Algeria 1

Tonga 1

Canada 1

Tanzania 1

South Georgia and the South Sandwich Islands 1

Nicaragua 1

Tajikistan 1

Italy 1

Botswana 1

Guatemala 1

Venezuela 1

Mongolia 1

El Salvador 1

Name: count, dtype: int64

top\_10\_countries.plot(kind='bar')

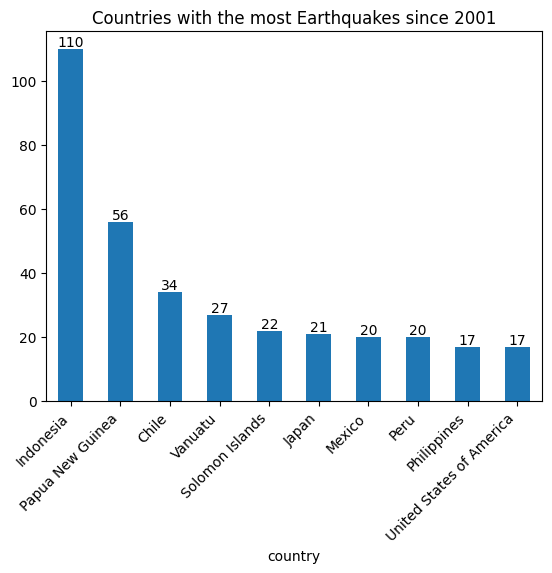
**for index, value in enumerate(top\_10\_countries):**

**plt.text(index, value, str(value), ha='center', va='bottom')**

**plt.title('Countries with the most Earthquakes since 2001')**

**plt.xticks(rotation=45, horizontalalignment='right')**

**plt.show()**

****

**continents = (df['continent'].value\_counts())**

**print(continents)**

continent

Asia 100

South America 55

North America 34

Europe 10

Oceania 4

Africa 3

Name: count, dtype: int64

**import plotly.express as px**

**fig = px.scatter\_geo(**

**df,**

**lat='latitude',**

**lon='longitude',**

**color='magnitude',**

**size='depth',**

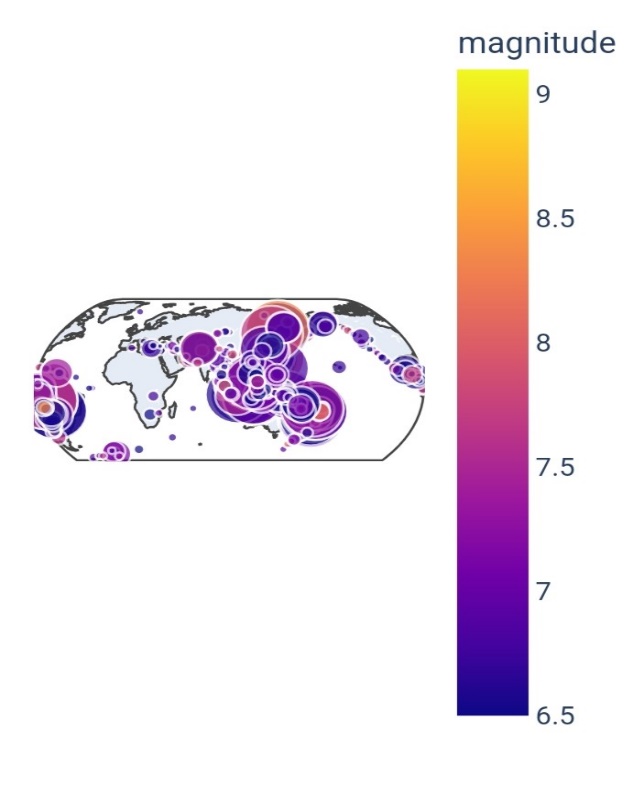
**hover\_name='location',**

**projection='natural earth',**

**title='Earthquake Occurrences Worldwide'**

**)**

**fig.show()**

****

**Earthquake hotspots:**

* Western Pacific Faultline
* Andes faultline in Latin America, extending into the Caribbean plate Very few major events since 2001 in Africa and Australia

Events in Europe only in the Eastern Mediterranean

***# finding the years with the largest average magnitude***

**avg\_magnitude\_per\_year = df.groupby('year')['magnitude'].mean().reset\_index().sort\_values(by='magnitude', ascending=False)**

**avg\_magnitude\_per\_year**

Out[27]:

|  |  |  |
| --- | --- | --- |
|  | year | magnitude |
| 8 | 2009 | 7.161538 |
| 11 | 2012 | 7.070968 |
| 6 | 2007 | 7.054054 |
| 20 | 2021 | 7.052381 |
| 0 | 2001 | 7.028571 |
| 9 | 2010 | 7.004878 |
| 10 | 2011 | 6.988235 |
| 3 | 2004 | 6.959375 |
| 17 | 2018 | 6.953488 |
| 15 | 2016 | 6.944186 |
| 4 | 2005 | 6.942857 |
| 5 | 2006 | 6.942308 |
| 19 | 2020 | 6.911111 |
| 7 | 2008 | 6.900000 |
| 1 | 2002 | 6.900000 |
| 14 | 2015 | 6.898113 |
| 12 | 2013 | 6.890566 |
| 2 | 2003 | 6.889032 |
| 18 | 2019 | 6.860606 |
| 13 | 2014 | 6.843750 |
| 21 | 2022 | 6.812500 |
| 16 | 2017 | 6.811111 |

***Conclusion:***

***Earthquake prediction remains a complex problem with significant uncertainty. Reliable earthquake prediction is still an ongoing area of research, and even the best models have limitations. Always ensure that you're using data from authoritative sources and consider collaborating with experts in seismology for more accurate insights.***

***THANK YOU…***