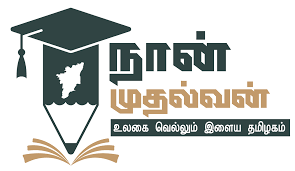
 **4222-SURYA GROUP OF INSTITUTIONS**  

**VIKARAVANDI -605 652**

**PROJECT NAME:**

**EARTHQUAKE-PREDICTION-USING-PYTHON**

**PREPARED BY:**

**K.ARUN**

**REG NO:422221106003**

**ECE DEPARTMENT**

**EARTHQUAKE-PREDICTION-USING-PYTHON**

**INTRODUCTION:**

Earthquake prediction is a challenging and complex task that is still an active area of research. It is a way to predict the magnitude of earthquake based on parameters such as longitude, latitude, depth, and duration magnitude, country. These approaches are based on the analysis of seismic data, historical earthquake data, and other relevant factors. People used to minimize loss of life and property.

**ML MODELS USED:**

* Linear Regression
* Decision Tree
* K-Nearest Neighbors

**STEPS TAKEN:**

* Data source
* Feature exploration
* Visualization
* Data splitting
* Training and evaluation

**DATA SOUCE:**

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
import os  
print(os.listdir("../input"))

['database.csv']

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SI NO | Date | Time | Latitude | Longitude | Depth | Magnitude |
| 0 | 01/02/1965 | 13:44:18 | 19.246 | 145.616 | 131.6 | 6.0 |
| 1 | 01/04/1965 | 11:29:49 | 1.863 | 127.352 | 80.0 | 5.8 |
| 2 | 01/05/1965 | 18:05:58 | -20.579 | -173.972 | 20.0 | 6.2 |
| 3 | 01/08/1965 | 18:49:43 | -59.076 | -23.557 | 15.0 | 5.8 |
| 4 | 01/09/1965 | 13:32:50 | 11.938 | 126.427 | 15.0 | 5.8 |

**FEATURE EXPLORATION:**

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')

Figure out the main features from earthquake data and create a object of that features, namely, Date, Time, Latitude, Longitude, Depth, Magnitude.

data = data[['Date', 'Time', 'Latitude', 'Longitude', 'Depth', 'Magnitude']]  
data.head()

Out[4]:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Date | Time | Latitude | Longitude | Depth | Magnitude |
| 0 | 01/02/1965 | 13:44:18 | 19.246 | 145.616 | 131.6 | 6.0 |
| 1 | 01/04/1965 | 11:29:49 | 1.863 | 127.352 | 80.0 | 5.8 |
| 2 | 01/05/1965 | 18:05:58 | -20.579 | -173.972 | 20.0 | 6.2 |
| 3 | 01/08/1965 | 18:49:43 | -59.076 | -23.557 | 15.0 | 5.8 |
| 4 | 01/09/1965 | 13:32:50 | 11.938 | 126.427 | 15.0 | 5.8 |

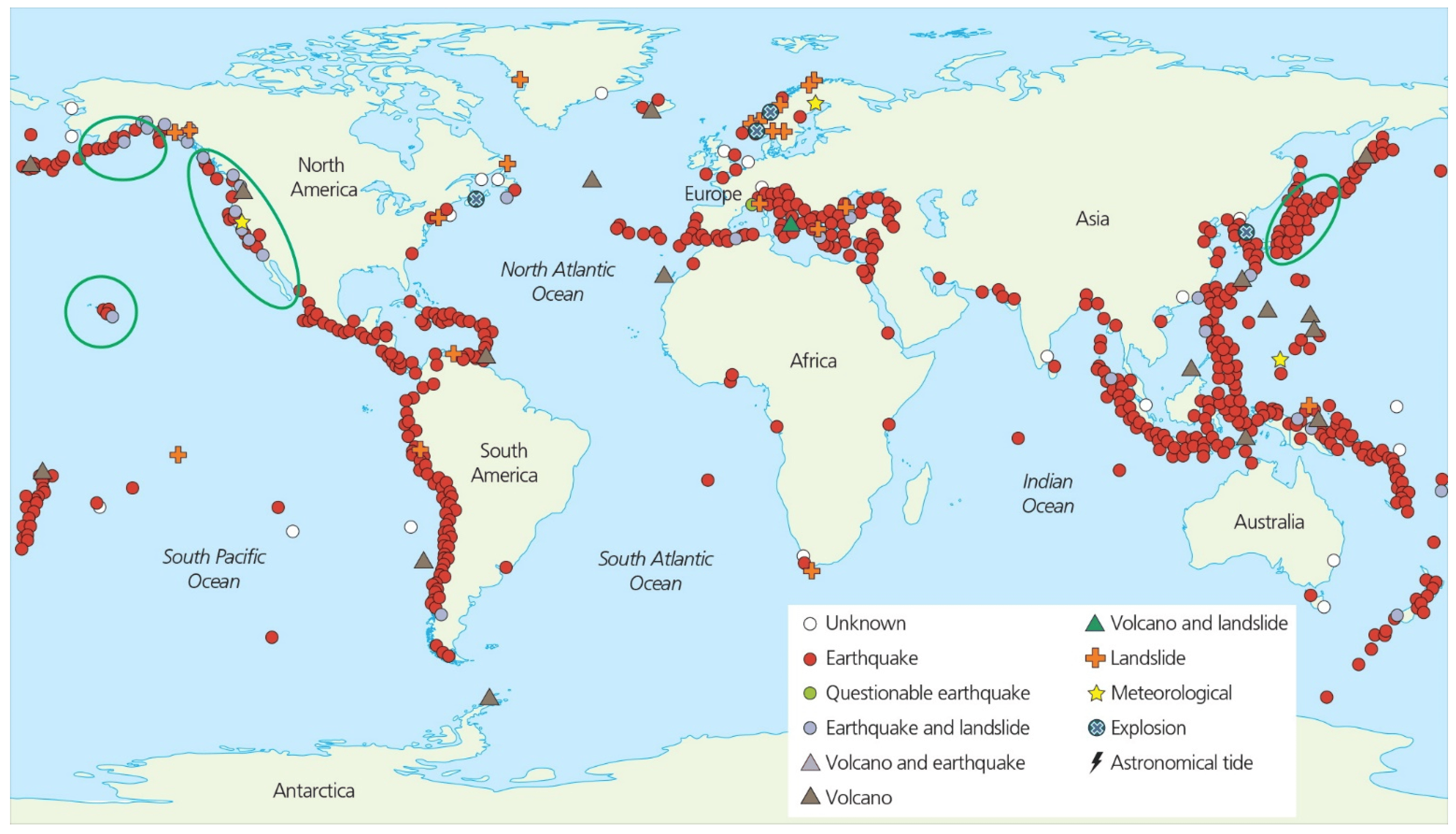
**Visualization:**

Here, all the earthquakes from the database in visualized on to the world map which shows clear representation of the locations where frequency of the earthquake will be more.

In [8]:

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)  
In [9]:

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:**

The data split was 90% train and 10% test.



**TRAINING AND EVALUATION**

# demonstrate that the train-test split procedure is repeatable

from sklearn.datasets import make\_blobs

from sklearn.model\_selection import train\_test\_split

# create dataset

X, y = make\_blobs(n\_samples=100)

# split into train test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=1)

# summarize first 5 rows

print(X\_train[:5, :])

# split again, and we should see the same split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=1)

# summarize first 5 rows

print(X\_train[:5, :])

[[-2.54341511 4.98947608]

[ 5.65996724 -8.50997751]

[-2.5072835 10.06155749]

[ 6.92679558 -5.91095498]

[ 6.01313957 -7.7749444 ]]

[[-2.54341511 4.98947608]

[ 5.65996724 -8.50997751]

[-2.5072835 10.06155749]

[ 6.92679558 -5.91095498]

[ 6.01313957 -7.7749444 ]]