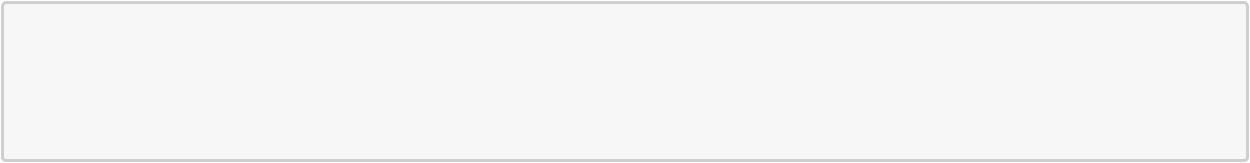
**Name:R.raja**

AI-Phase-4

October 24, 2023



* ]: Date:24 october 2023 Team ID:NM2023TMID345

Team Name:Proj\_227274\_team\_1

Project Name:Earthquake prediction model



[5]: **import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from mpl\_toolkits.basemap import** Basemap

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



* ]: *# Load your earthquake data into a Pandas DataFrame*



[6]: data = pd.read\_csv(r'C:\Users\91912\Desktop\AI\_Phase3/database.csv')



[7]: data.head()

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [7]: | Date | Time | Latitude | Longitude | Type | Depth | Depth Error \ |
| 0 | 01-02-1965 13:44:18 | | 19.246 | 145.616 | Earthquake | 131.6 | NaN |
| 1 | 01-04-1965 11:29:49 | | 1.863 | 127.352 | Earthquake | 80.0 | NaN |
| 2 | 01-05-1965 | 18:05:58 | -20.579 | -173.972 | Earthquake | 20.0 | NaN |
| 3 | 01-08-1965 | 18:49:43 | -59.076 | -23.557 | Earthquake | 15.0 | NaN |
| 4 | 01-09-1965 | 13:32:50 | 11.938 | 126.427 | Earthquake | 15.0 | NaN |

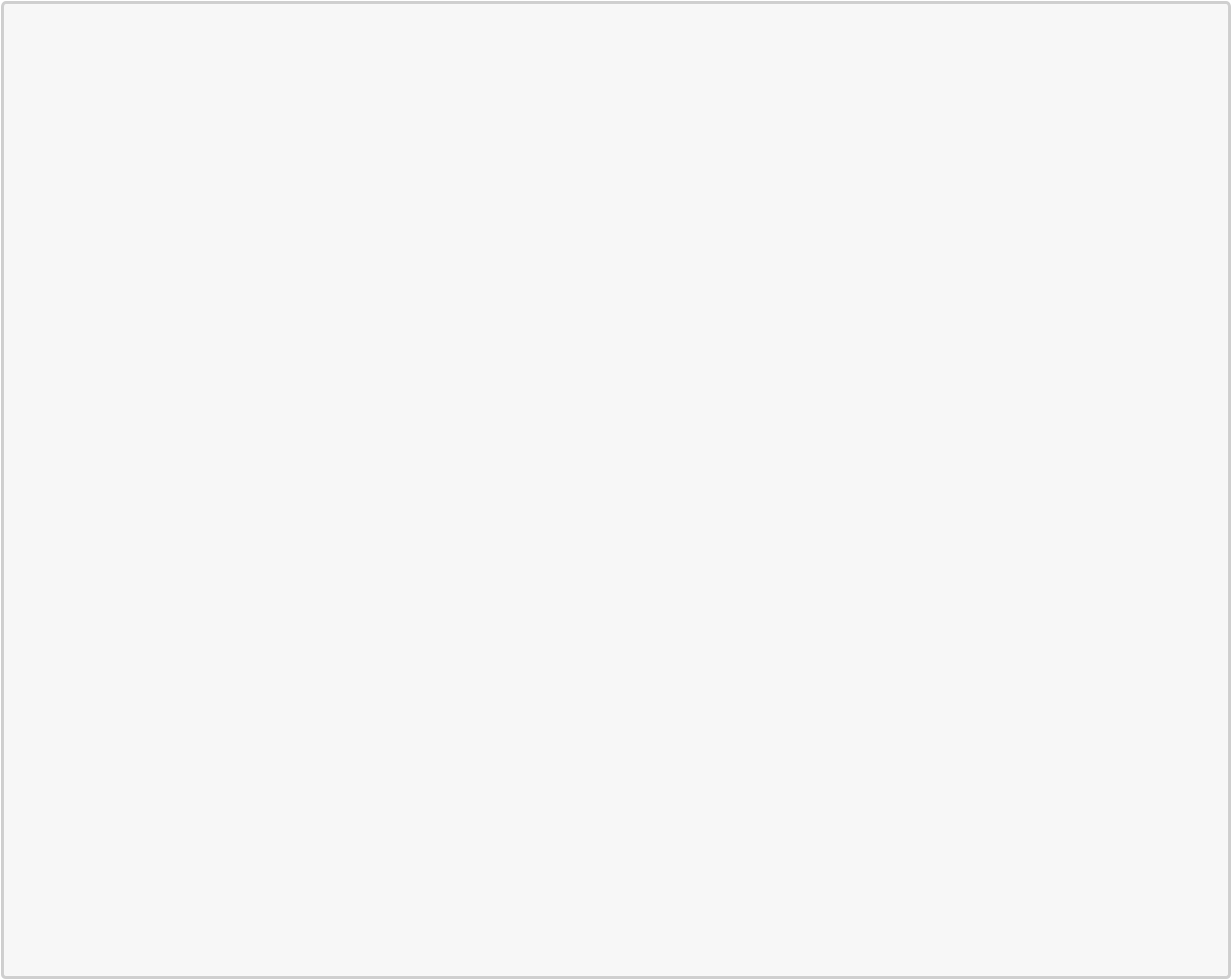
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Depth Seismic Stations | Magnitude Magnitude Type | | … \ |
| 0 | NaN | 6.0 | MW | … |
| 1 | NaN | 5.8 | MW | … |
| 2 | NaN | 6.2 | MW | … |
| 3 | NaN | 5.8 | MW | … |
| 4 | NaN | 5.8 | MW | … |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Magnitude Seismic Stations | Azimuthal Gap | Horizontal Distance \ |
| 0 | NaN | NaN | NaN |
| 1 | NaN | NaN | NaN |
| 2 | NaN | NaN | NaN |
| 3 | NaN | NaN | NaN |
| 4 | NaN | NaN | NaN |

1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Horizontal Error | Root Mean Square | ID | Source Location Source \ | |
| 0 | NaN | NaN | ISCGEM860706 | ISCGEM | ISCGEM |
| 1 | NaN | NaN | ISCGEM860737 | ISCGEM | ISCGEM |
| 2 | NaN | NaN | ISCGEM860762 | ISCGEM | ISCGEM |
| 3 | NaN | NaN | ISCGEM860856 | ISCGEM | ISCGEM |
| 4 | NaN | NaN | ISCGEM860890 | ISCGEM | ISCGEM |
|  | Magnitude Source | Status |  |  |  |
| 0 | ISCGEM | Automatic |  |  |  |
| 1 | ISCGEM | Automatic |  |  |  |
| 2 | ISCGEM | Automatic |  |  |  |
| 3 | ISCGEM | Automatic |  |  |  |
| 4 | ISCGEM | Automatic |  |  |  |

[5 rows x 21 columns]



[8]: **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')

min\_timestamp = datetime.datetime(1970, 1, 1)

max\_timestamp = datetime.datetime(2038, 1, 19)

**if** min\_timestamp <= ts <= max\_timestamp:

timestamp.append(time.mktime(ts.timetuple()))

**else**:

timestamp.append('OutofRange')

**except ValueError**:

* *print('ValueError')* timestamp.append('ValueError')

data['Timestamp'] = timestamp

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()

[8]: Latitude Longitude Type Depth Depth Error \

7 -13.309 166.212 Earthquake 35.0 NaN

2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 8 | -56.452 | -27.043 | Earthquake | 95.0 | NaN |
| 9 | -24.563 | 178.487 | Earthquake | 565.0 | NaN |
| 10 | -6.807 | 108.988 | Earthquake | 227.9 | NaN |
| 11 | -2.608 | 125.952 | Earthquake | 20.0 | NaN |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Depth Seismic Stations | Magnitude Magnitude Type | | Magnitude Error \ |
| 7 | NaN | 6.0 | MW | NaN |
| 8 | NaN | 6.0 | MW | NaN |
| 9 | NaN | 5.8 | MW | NaN |
| 10 | NaN | 5.9 | MW | NaN |
| 11 | NaN | 8.2 | MW | NaN |

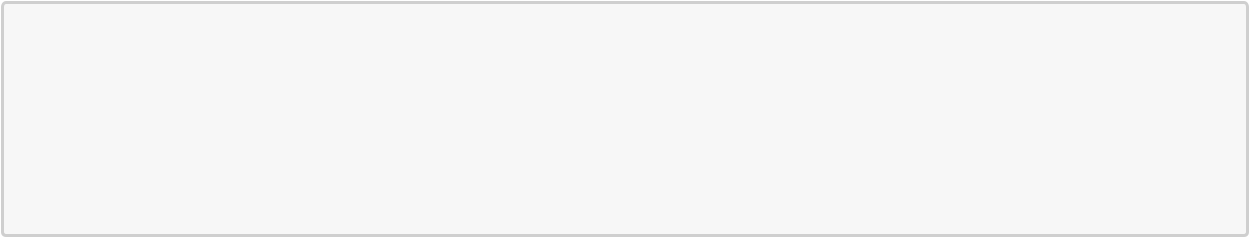
|  |  |  |  |
| --- | --- | --- | --- |
|  | Magnitude Seismic Stations | Azimuthal Gap | Horizontal Distance \ |
| 7 | NaN | NaN | NaN |
| 8 | NaN | NaN | NaN |
| 9 | NaN | NaN | NaN |
| 10 | NaN | NaN | NaN |
| 11 | NaN | NaN | NaN |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Horizontal Error | Root Mean Square | ID | Source \ |
| 7 | NaN | NaN | ISCGEM861111 | ISCGEM |
| 8 | NaN | NaN | ISCGEMSUP861125 | ISCGEMSUP |
| 9 | NaN | NaN | ISCGEM861148 | ISCGEM |
| 10 | NaN | NaN | ISCGEM861155 | ISCGEM |
| 11 | NaN | NaN | ISCGEM861299 | ISCGEM |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Location Source Magnitude Source | | Status | Timestamp |
| 7 | ISCGEM | ISCGEM | Automatic | OutofRange |
| 8 | ISCGEM | ISCGEM | Automatic | OutofRange |
| 9 | ISCGEM | ISCGEM | Automatic | OutofRange |
| 10 | ISCGEM | ISCGEM | Automatic | OutofRange |
| 11 | ISCGEM | ISCGEM | Automatic | OutofRange |



* ]: *# Visualizing the data on a world map*

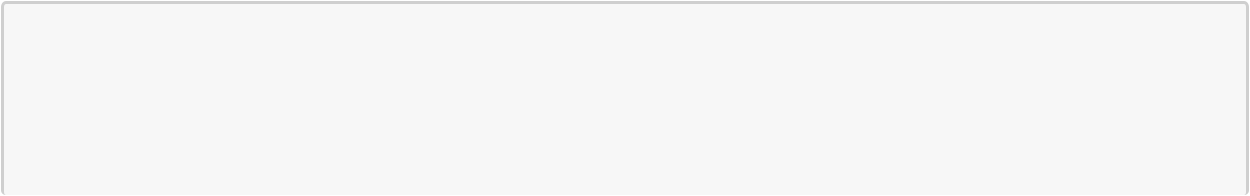


[9]: m = Basemap(projection='mill', llcrnrlat=-80, urcrnrlat=80, llcrnrlon=-180,␣ ↪urcrnrlon=180, lat\_ts=20, resolution='c')

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

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

x, y = m(longitudes, latitudes)



[10]: fig = plt.figure(figsize=(12,10))

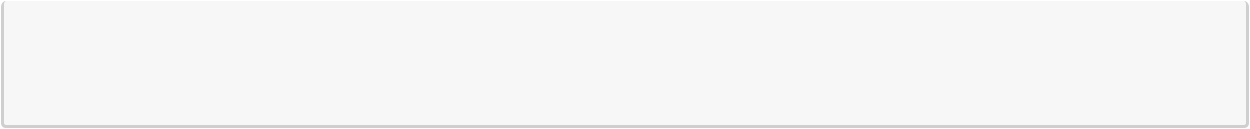
plt.title("All affected areas")

m.plot(x, y, "o", markersize = 2, color = 'blue')

m.drawcoastlines()

m.fillcontinents(color='coral',lake\_color='aqua')

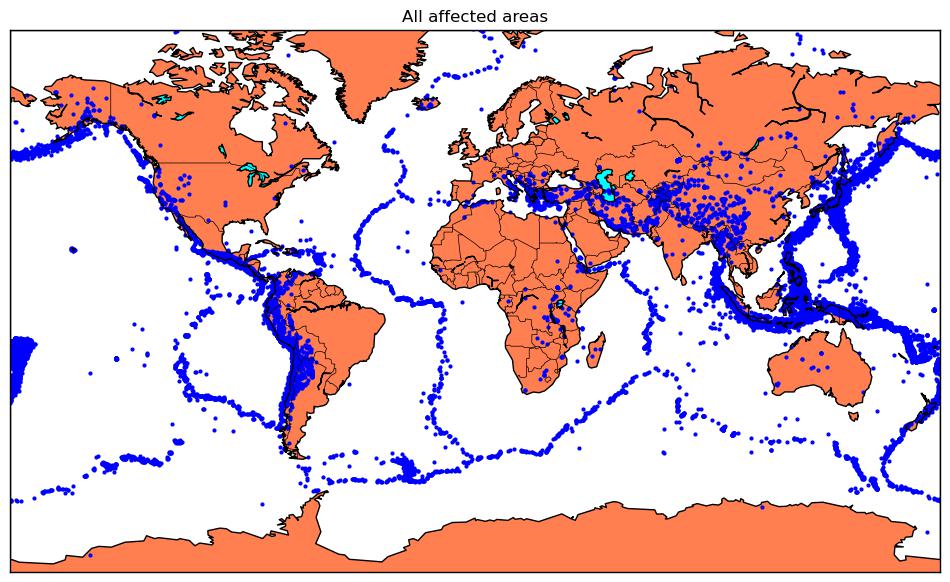
3



m.drawmapboundary()

m.drawcountries()

plt.show()



* ]: *# Splitting it into training and testing sets*



[5]: **import pandas as pd**

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



* ]: *# Load your earthquake data into a Pandas DataFrame*



[20]: data = pd.read\_csv(r'C:\Users\91912\Desktop\AI\_Phase3/database.csv')



* ]: *# Define your feature columns and target columns*



[17]: X = data[['Latitude', 'Longitude']]

y = data[['Magnitude', 'Depth']]



* ]: *# Split the data into training and testing sets*

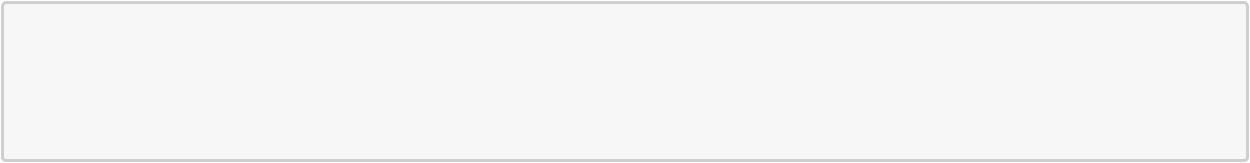


[18]: X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,␣ ↪random\_state=42)



[ ]: *# Print the shapes of the training and testing sets to verify the split*

4



[19]: print("X\_train shape:", X\_train.shape)

print("X\_test shape:", X\_test.shape)

print("y\_train shape:", y\_train.shape)

print("y\_test shape:", y\_test.shape)

X\_train shape: (18729, 2)

X\_test shape: (4683, 2)

y\_train shape: (18729, 2)

y\_test shape: (4683, 2)



* ]: *# using a Decision Tree regressor for your earthquake prediction model.*



[21]: **from sklearn.tree import** DecisionTreeRegressor

**from sklearn.metrics import** mean\_squared\_error, mean\_absolute\_error, r2\_score



* ]: *# Create a Decision Tree regressor*



[23]: tree\_model = DecisionTreeRegressor(random\_state=42)



* ]: *# Fit the Decision Tree model to the training data*



[24]: tree\_model.fit(X\_train, y\_train)

[24]: DecisionTreeRegressor(random\_state=42)



* ]: *# Make predictions on the testing data*



[25]: y\_pred\_tree = tree\_model.predict(X\_test)



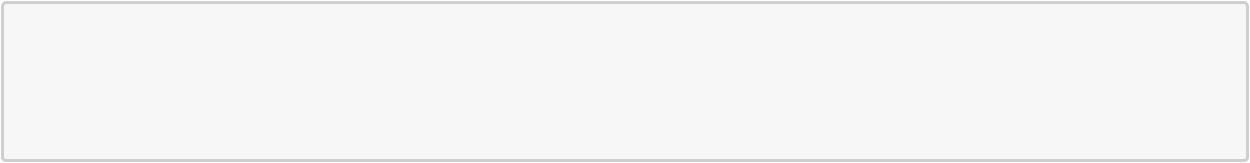
* ]: *# Evaluate the Decision Tree model*



[26]: mse\_tree = mean\_squared\_error(y\_test, y\_pred\_tree)

mae\_tree = mean\_absolute\_error(y\_test, y\_pred\_tree)

r2\_tree = r2\_score(y\_test, y\_pred\_tree)



[27]: print("Decision Tree Model:")

print("Mean Squared Error:", mse\_tree)

print("Mean Absolute Error:", mae\_tree)

print("R-squared Score:", r2\_tree)

Decision Tree Model:

Mean Squared Error: 1678.1163417214373

Mean Absolute Error: 12.115441063420867

R-squared Score: -0.03476557402029423

5