**Visualization**

Here, we will visualize the earthquakes that have occurred all around the world.

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()

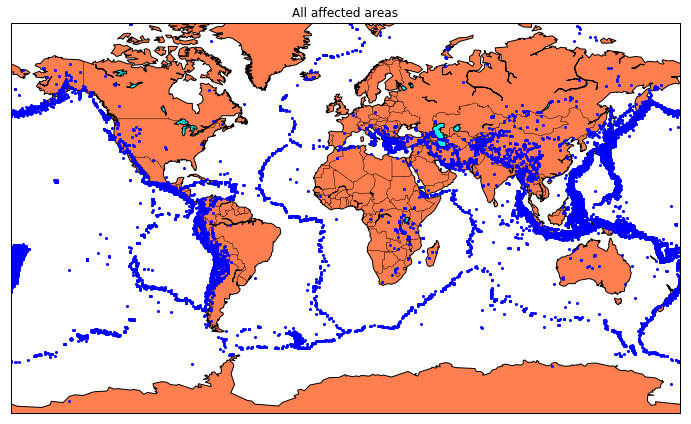
m.fillcontinents(color=’coral’,lake\_color=’aqua’)

m.drawmapboundary()

m.drawcountries()

plt.show()

Output:



**Splitting The Dataset**

Now we will split the dataset into a training and testing set.

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

Output:

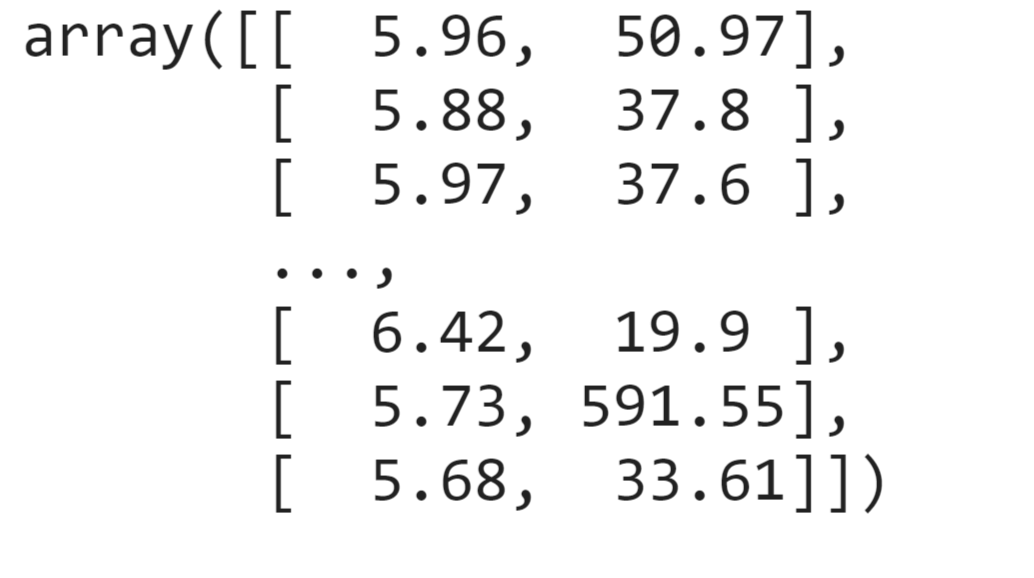
Earthquake Prediction Using Machine Learning

We will be using the RandomForestRegressor model to predict the earthquake, here will look for its accuracy.

Reg = RandomForestRegressor(random\_state=42)

Reg.fit(X\_train, y\_train)

Reg.predict(X\_test)

**Output:**

Reg.score(X\_test, y\_test)

Output:



86% of accuracy is quite high.

Now we will shift to GridSearch.

From sklearn.model\_selection import GridSearchCV

Parameters = {‘n\_estimators’:[10, 20, 50, 100, 200, 500]}

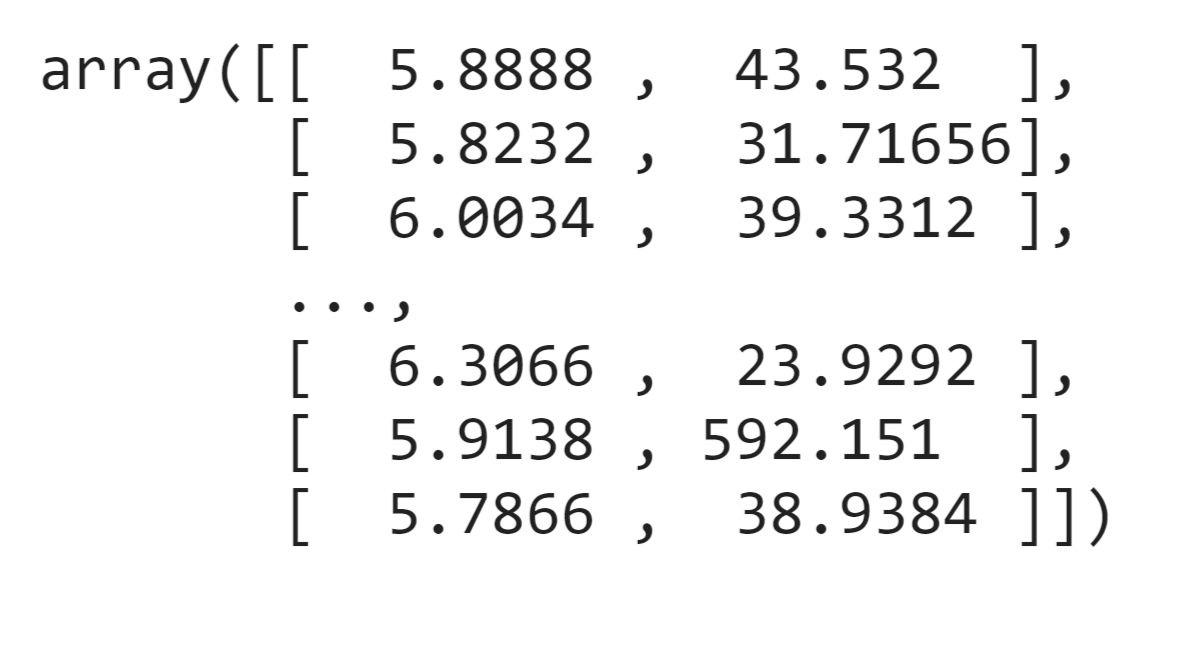
Grid\_obj = GridSearchCV(reg, parameters)

Grid\_fit = grid\_obj.fit(X\_train, y\_train)

Best\_fit = grid\_fit.best\_estimator\_

Best\_fit.predict(X\_test)

Output:

best\_fit.score(X\_test, y\_test)

Output:



Considering it’s a natural phenomenon, we have got a high accuracy number.