EARTHQUAKE PREDICTION MODEL USING MACHINE LEARNING

COLLEGE CODE: 8106
NAME: SARAVANAN.K
PHASE 1 -SUBMISSION

Abstract :

Earthquake prediction is a cruciat area of research in seismotogy and geophysics. In this project, we aim to develop a data-driven earthquake prediction model using Python and a Kaggte dataset containing historical earthquake data. The primary goal is to analyse patterns and build a machine learning model that can provide probabilistic earthquake predictions. Additionally, we will visualize earthquake occurrences on a world map to gain insights into their spatial distribution.

Modules:

Python libraries:

The following Python modules are needed to build an earthquake prediction model using the Kaggle earthquake dataset and visualize the results on a world map:

1. numpy - for scientific computing
2. pandas - for data manipulation and analysis
3. matplotlib - for data visuaiizat\on2
4. folium - for interactive map visualization
5. sCikit-learn - for machine learning

IMPORTING PYTHON LIBRARIES:

```
import numpy as np
import pandas as pd
import matplotIib.pypIot as plt
import folium
from skIearn.model selection import train test split
from skiearn.ensembIe import RandomForestRegressar
```

Metnods:

1. **Data Acquisition and Preprocessing:**

   Download the Kaggle dataset and obtain the required day from it. Pre-process the dataset, inc(uding data cleaning, handling missing values, and converting data types.

```
# Load the earthquake dataset

df = pd.read_csv('earthquake-database.csv')
```

2. **Exploratory Data Analysis (EDA):**

Perform statistical **analysis** to understand the distribution of earthquake magnitudes, depths, and locations. Visualize earthquake data with histograms, scatter plots, and time serias ptote.

3. **Feature Engineering:**

Extract relevant features from the dataset, such as earthquake location coordinates. time, magnitude, depth, and possibly external factors like geological data, weather, or tectonic plate information.

```
# Prepare the data
#Remove outliers
df = df[(df['Magnitude'] > 5) & (df['Oepth'] > 0)s
```

the data from tha csv file where magnitude is greater than 5 and depth is greater than 0.

4. **Data Splitting:**

Split the dataset into training and teMing subsets to evaluate model performance.

\# **Split the data into training and testing sets**

```
X train, X_test, y train, y_test = train_test_sptit(df[['Latitude, 'Longitude’]],
df{'f•\agnitude'}, test_size=0.25)
```

5. f4ode\ **Selection and Training:**

Choose appropriate machine learning models for earthquake prediction (e.g., regression, classification, time aeries forecasting). Train and fine-tune the selected models using the training data.

```
# Train tha maehine learning model

model = RandomForestRegressor()
model.fit(X_train, y_train)
```

WHY RANDOM FOREST REGRESSION MODEL:

A random forest regression model is easy to identify the struGtural safety status of buildings damaged by the earthquake is probabilistic. An earthquake’s Latitude, longitude. magnitude, and depth may be predicted using tha random forest algorithm.

A random forest with muftioutput technique is employed, with variables being each station’s recorded value and geographic position.

B. **Model Evaluation:**

Evaluate the model's performance using appropriate metrics i.e., Mean Absolute Error.

```
# Evaluate the model on the testing set
y red = model.predict(X test)
    print('Model accuracy:', np.mean(y_pred == y test))
```

7. Visualization with World I•Iap:

Use library Folium to create a world map. Plot earthquake data on the world map using latitude and longitude information. Customize markers to represent earthquake attributes such as magnitude. depth, and date.

fi Visualize the resutta on a wortd map

wortd_map = fotium.Map()

Fotium :

Folium makes it easy to visualize data that's been manipulated in python on an interactive leaflet map. It enables both the binding of data to a map for choropteth visualizations as passing as a HTML visualization as markers on the map.

B. Plotting the data.

```
# Plot the training data
for i in range(len(X_train)):
folium.CircIeMarker(
location= [X_train.iloC[i, 0], X_train.iIoc[i, 1j],
 popup= str (X train.iloc[L 0]) + ', ' + str(X_train.iloc|i, 1]),
coIor='bIue',
fiII_coIor='bIue',
fiII_opacity=0.5
).add_to(worId_map)

# Plot the testing data
for i in range(Ien(X_test)):
foIium.CircIeMarker(
location= [X_test.iIoc[i, 0], X_test.hoc[i, 1]],
    popup= str (X_test.iIoc[i, 0]) + ',' + str(X_test.iloc[i, 1]),
coIor='red',
fiII_coIor='red',
fiII_opacity=0.5
).add_to(worId_map)

world_map.save('earthquake rediction_map.html')
```

```
import pandas as pd

import numpy aa np

import matplotlib.pyplot aa pit

!pip ínstall folium
```

```
Collecting folium
```

```
osiDg cached folium-0.l4.0-py2.py3-none-aOy.whl (102 kB)
Collecting branca>=0,6.0 (from folium)
Psing cached branca-0.6.0-py3-none-any.whl (24 kB)
Requirement already satisfied: jinja2>=2.9 in
c:\user6\sriOi\,conda\lib\site-packages (from folium) (3.1.2}
Requirement already satisfied: numpy in c:\users\srini\.conda\lib\site-
packages {from folium) (1,24.3)
Requirement already satisfied: requests in c:\users\srini\.conda\lib\site-
packages (from folium) (2.31.0)
Requirement already satisfied: Markupsate·=2.D in
c:\users\srini\.conda\lib\sire-packages ([rom jinja2>=2.9- folium) (2.1.1)
Requirement already satisfied: charset-normalizer‹4,>=2 in
c:\users\srini\,conda\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\srini\.conda\lib\site-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<2,>=l.2l.l in
c:\users\srini\.conda\lib\site-packages {from requests->fOlium) (1.26.16)
Requirement already satisfied: certifi>=2017,4.17 im
c:\users\sriDi\.conda\lib\site-packages (from requests->folium)
(2023.7.22) Installing collected packages: braoca, folium
```

Successfully insea12ed branea- 0 . 6 . 0 £olium - 0 . 14 . 0

```
import lolium
```

```
from sk1earn.ensemble import RandomForestRegressor
```

```
df pd.read_csv(r'/Dsers/srini/OneDrive/Documents/database.csv')
print(df)
```

```
          Date Time Latitude Longitude                    Type Oeptfl \
0     01/02/1965   13:44:18 19.2460 145.6160 Earthquake  121.60
1     01/04/1965 11:29:49 1.8630 127.3520          Earthquake 80.00
2     01/05/196s    18.05.5B -a0.6790 -172.9780 Earthquake    20.00
3     01/08/196s    18.49:42 -59.0760 -22.5570 Earthquake     15.00
4     01/09/196s Earth le 13:32:50  11.9380  126.4270          15.0D

234 07  12/28/2016  0B:22:12  38.3917  -118.a941  Earthquak  12.30
234 0B  12/28/2016  09:l3:47  38.3777  -118.8957  e          B.B0
234 09  12/28/2016  l2:18:51  36.9179 140.4262    Earthquak  10.00
23410   12/29/2016  22:30:19  -9.02BE3  1I8.6629  e          79.00
234 11  12/30/2016  20:08:28  y.3973   141.4T02   Earthquak  11,94
                                                  e
                                                  Earthquak
                                                  e
                                                  Earthquak
                                                  e

     Depth Error Depth seismic Stations Nagnitude Magnitude Type.  \
0        NaN              NaN       6.0          MW
1        NaN              NaN       5.8          MW  ...
2        Na               NaN       6.2          M
3        N                NaN       B.8          W
4        NaN              NaN       5.B          MW
         N                                       W
234 07   1.2              40.0      5 C          ML  ...
```

23408 2.0 33.0 5.5 ML ...
23409 1.8 NaN 5.9 MWW
23410 1.8 NaN 6.3 MWW
23411 2.2 NaN 5.5 MB ...

Magnitude Seismic Stations Azimutbal Gap Horizontal Distance \
0 maN NaN NaN
1 NaN NaN NaN
2 NaN NaN NaN
3 san NaN NaN
4 NaN NaN NaN

23407 1.8 0 42.47 0,120
23408 18.0 48.58 0.129
23409 NaN 91.00 0.992
23410 NaN 26.00 3.553
23411 428.0 9?.00 0.681

Horizontal Error Root Mean Square ID Source \
0 NaN NaN ISCGEM860706 ISCGEM
1 NaN NaN ISCGEN860737 ISCGEM
2 NaN NaN ISCGEM860762 ISCGEM
3 NaN NaN ISCGEM860856 ISCGEM
4 Ra 1'4 RAM ISCGEM860890 ISCGEM

23407 NaN O.189B Nm00570710 NN
23408 NaN 0.2187 Nm00570744 NN
23409 4.8 1.5200 DS10007NAR US
23410 6.0 1,4300 US10007NTD US
23411 4.5 0.9100 US10007NTL0 US

Location Source Nagnitude Source status
0 ISCGEM ISCGEM Automatic
1 ISC€EM ISCGEM Automatic
2 ISC€EM ISCGEM Automatic
3 ISCGEM ISCOEM Automatic
4 ISCGEM ISCGEM Automatic

23407 NN NN Reviewed
23408 NN NN Reviewed
23409 US US Reviewed
23410 US DS Reviewed
23411 US DS Reviewed

(23412 rows x 21 columns)

```
df = df[(df['Magnitude') > 5) & (df['Depth') > 0))
```

```
print(df)
```

```
                Date Time Latitude Loogitude Type Depth \
U       DI/02/1965 13:44:18 19.2460 145.6160 Earthquake 131.60
1       01/04/1965 II: 29 : 49 I . 863D 127 . 3520 Earthquake 80.00
2       01/05/1965 GB : 05 : 58 -2Q . 579D - 173.3720 Earthquake Z0.00
3       0 / 0 8 /1 9 d 5 18:49- 42 -59.0750 -22. 5510 Earthquake 15.00
4       D1/09/1965 12 : 3Z 50 11 . 938 0 126. 4270 Earthquake 15.00

23407 12/29/2016     0B: 22:12    28.3917 38.    -118.894I   Earthquake   12.30
2 3 4 DR   I2 / 2 8 /2 0 I 6 12   D9 : I3 ' 47   3777  36.   -1I8. 8957   Earthquake   8.80
2 3 4 0 9   / 2 B / 2 01 6 12   12 : *8 : 51   9179  -9.   140. 4262   Earthquake   10.00
2 3 41 D   /2 9 / 2 0 I 6 12 /   22 : 30 : I9   0283  37  .   118.6639   Earthquak   79.00
2 3 4 11   3 D / 2 0 I 6   2D : 08 : 28   2973   141. 4103   e   11.94
                                              Earthquak
       Depth Error Depth Seismic Stations Magnitude Magnitude Type      \
                                              e
0            NaN                NaN       6.0            MW
1            NaN                NaN       5.8            MW
2            NaN                NaN       6.2            MW
```

3	NaN	NaN	5.8	MW	...
4	NaN	NaN	5.8	MW	...
234 07	1.2	40.0	5.6	ML	...
23 4 08	2.0	33.0	5.5	ML	
2 34 0 9	1.8	NaN	8.9	Mww	
2 34 10	1.8	NaN	6.3	Mww	
23 411	2.2	NaN	5.5	MB	

Magnitude Seismic Stations Azimuthal Gap Horizontal Distance \

1 NaN NaN NaN
2 NaN NaN
3 NaN NaN NaN
4 NaN NaN NaN
NaN

234 07	8.0	42.47	0.120
23 4	18.0	48.58	0.129
08 23	NaN	9i.00	0.992
4 09 23410	NaN	26.00	3.5b3
23411	428.0	97.00	0.68i

Borizootal Error Root Mean Square LD Source \
0 *Ra* 1'4 *RAM* ISCGEM860706 ISCGEM
1 NaN NaN ISCGFMB60737 ISCGEN
2 NaN NaN ISCGEMB6O762 ISCGEM
3 NaN NaN ISCGEMB6O856 ISCGEM
4 NaN NaN ISCGEMB60890 ISCGEM

234 07	Na	0.1898	NN00570710	NN
234 0B	N	0.Zi87	NN00570744	NN
234 09	Na 4.8	1.5200	DSI0007NAF	US
23 41 0	N 6.0	1.4300	DSI0007NL	DS
234 11	4.5	0.9i00	0 oSI0007mT	oS

Location Source Magnitude source Status
0 ISCGEM ISCGEM Automatic
1 ISCGEM ISCGEM Automatic
2 ISCGEM ISCGEM Automatic
3 ISCGEM ISCGEM Automatic
4 ISCGEN ISCGEM Automatic

234 07 NN m Reviewed

Z3408	NN	NN	Rev ie we d
23409	US	US	Reviewe d
23410	US	US	Reviewe d
22411	US	US	Reviewe d

[23239 rows x 21 columns)

```
X train, X test, y train, y test = train test split(df[[' atitude',
'Longitude']], df['Magnitude'], test size=0.25)
```

```
model =
RandomFOrestRegressor()
model.fitIX_train, y train)
```

RandomForestRegressor()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

RandomForestRegressor

RandomForestRegressor( )

```
ypred = model.predict(X test)
print('Model accuracy: ', np.mean(ypred == y test))
```

```
Model accuracy: 0.0D0172Il70395869I9IZ
```

```
world_map = folium.Map()
```

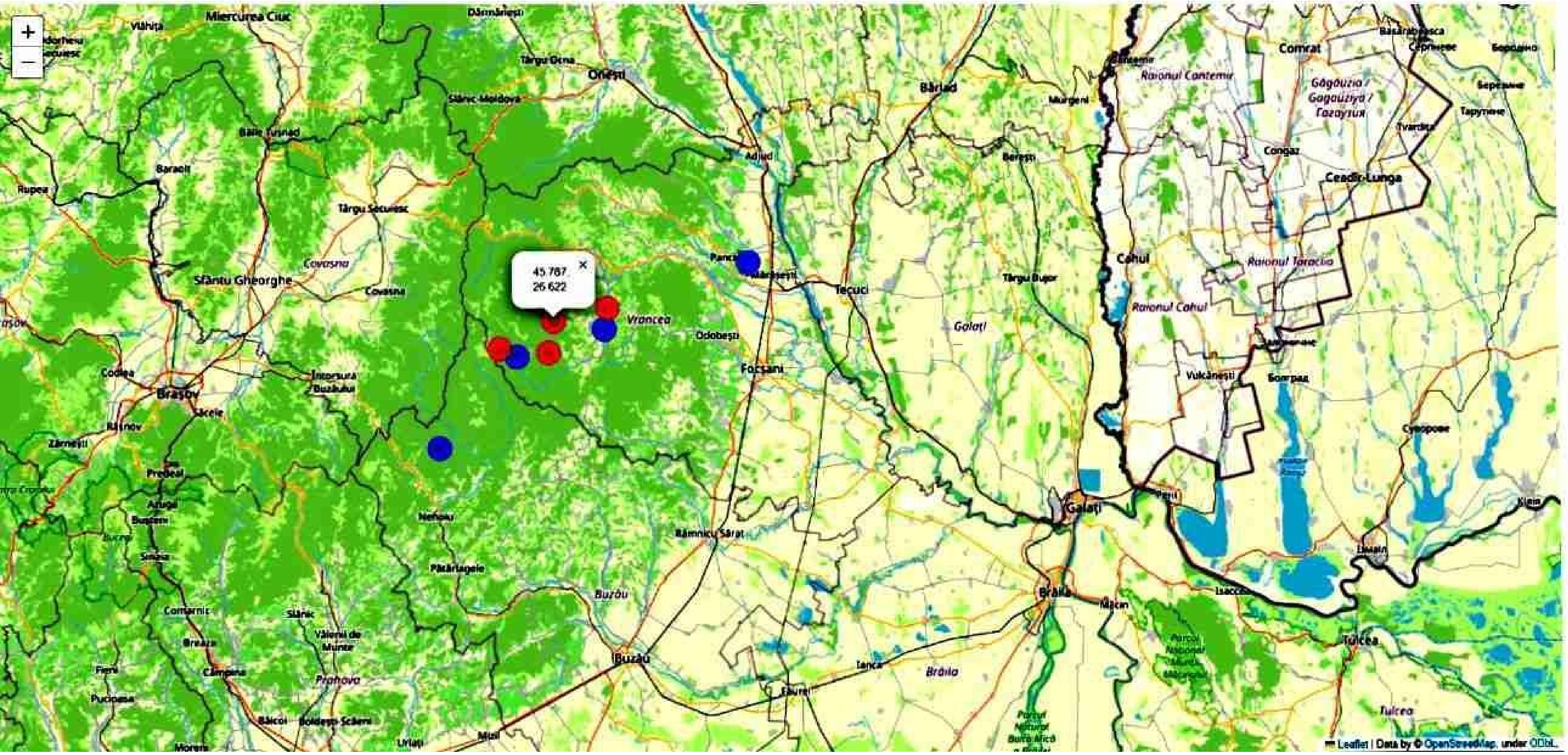
```
print(world_map)
```

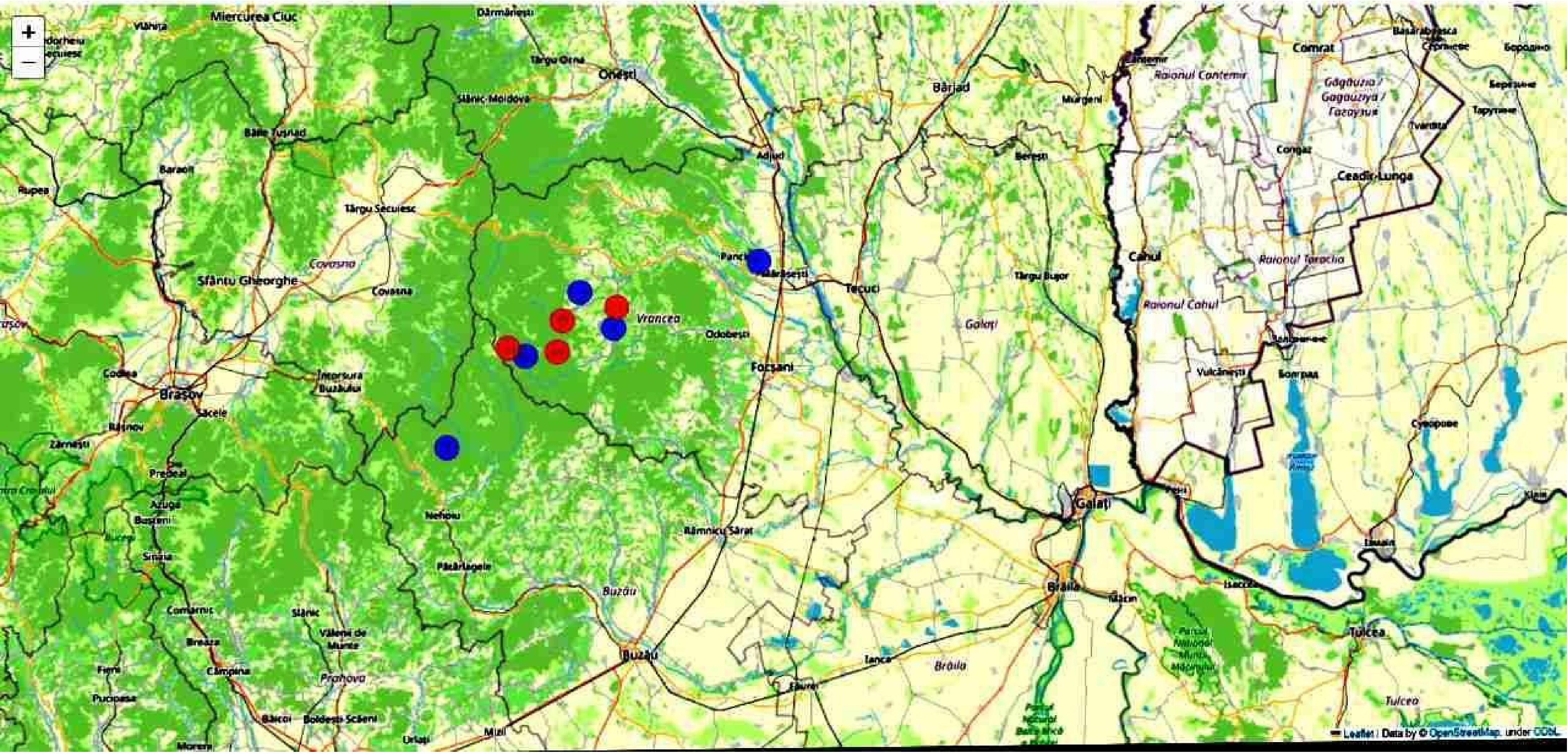
```
<folium.folium.Map Object at 0x0000026DBB0A4850>
```

```
for i in raoge(len(X traio)) :
folium.CircleMarkerl
location= (X train.iloc[i, D), X train.iloc[i, 1)),
                                        popup= str (X_train.iloc[i, 0)) + ', ' + str(X train.iloc(i, lj),
color='biue',
fill color='blue',
fill_opacity=D.5
       ).add to(world map)
```

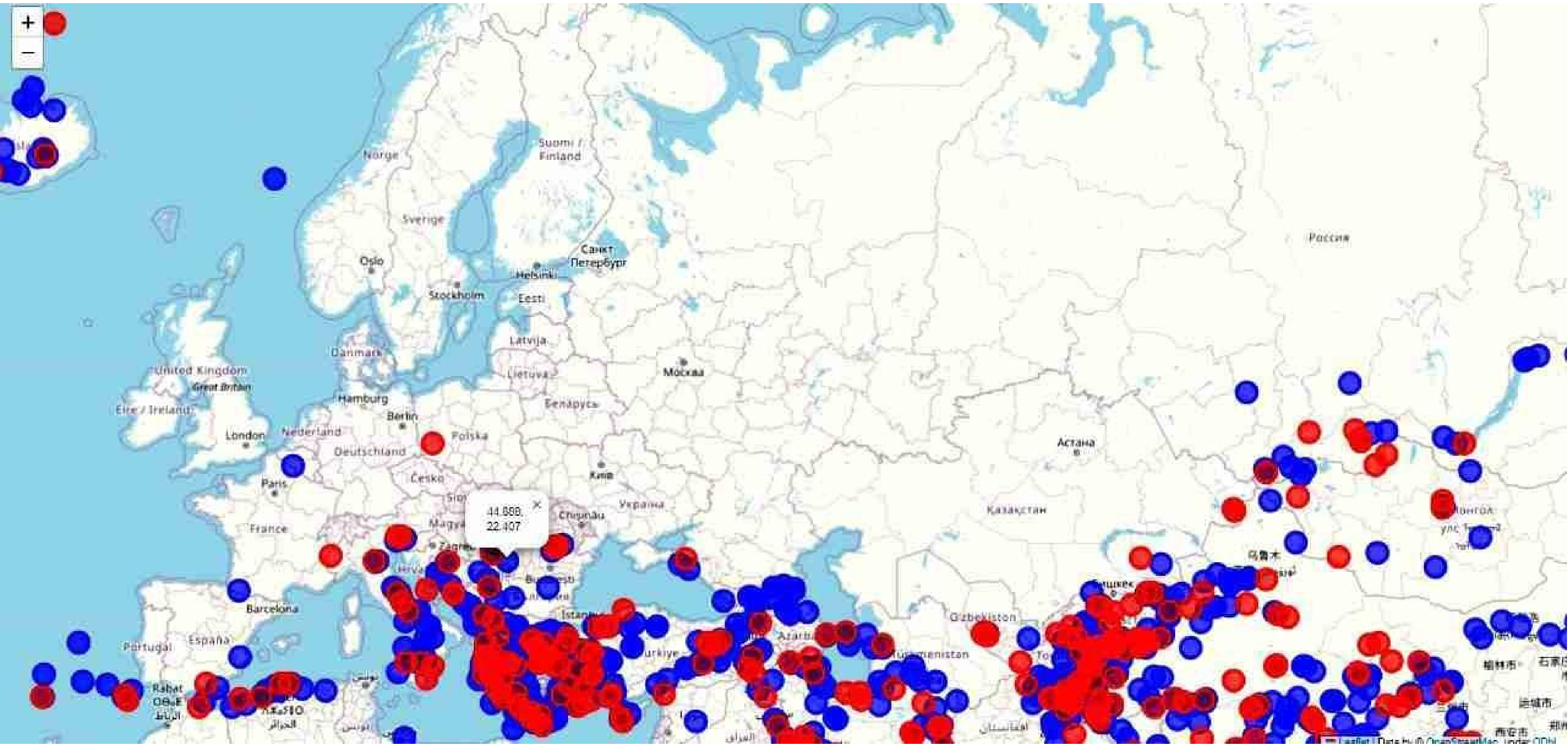
```
for i in range(len(X test)) :
      folium.CircleMarker(
      location= (X test.iloc(i, 0), X_test.iloc(i, 1)),
                       popup= str (X_test.iloc[i, 0)) + ', ' + str(X test.iloc(i, 1)),
              color='red',
              fill        color='red',
              fill_opacity=D.5
       ).add to(world map)

world map.save('earthquake_prediction map.html')
```

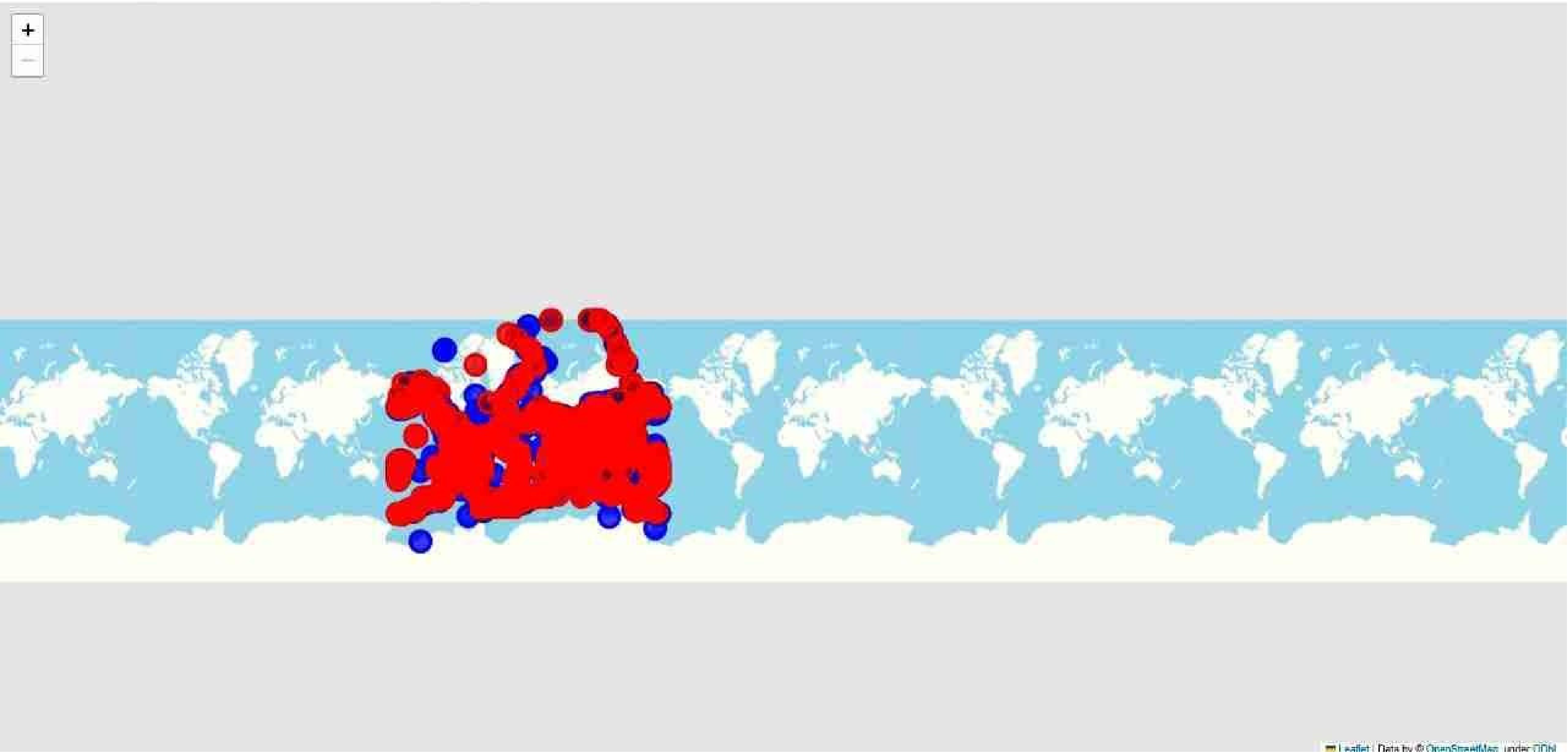

45.787,
26.622
Miercurea Ciuc
Vlăhița
Dărmănești
Târgu Ocna
Onești
Slănic-Moldova
Băile Tușnad
Baraolt
Rupea
Târgu Secuiesc
Covasna
Sfântu Gheorghe
Codlea
Brașov
Săcele
Râșnov
Zărnești
Predeal
Azuga
Bușteni
Sinaia
Comarnic
Breaza
Câmpina
Prahova
Băicoi
Boldești-Scăeni
Vălenii de Munte
Slănic
Nehoiu
Pătârlagele
Buzău
Urlați
Mizil
Adjud
Panciu
Mărășești
Tecuci
Vrancea
Odobești
Focșani
Râmnicu Sărat
Bârlad
Murgeni
Berești
Târgu Bujor
Galați
Brăila
Măcin
Isaccea
Tulcea
Ianca
Făurei
Cahul
Raionul Cantemir
Raionul Cahul
Raionul Taraclia
Comrat
Găgăuzia /
Gagauziya /
Гагаузия
Congaz
Ceadir-Lunga
Tvardița
Vulcănești
Болград
Reni
Ismail
Leaflet | Data by © OpenStreetMap, under ODbL.

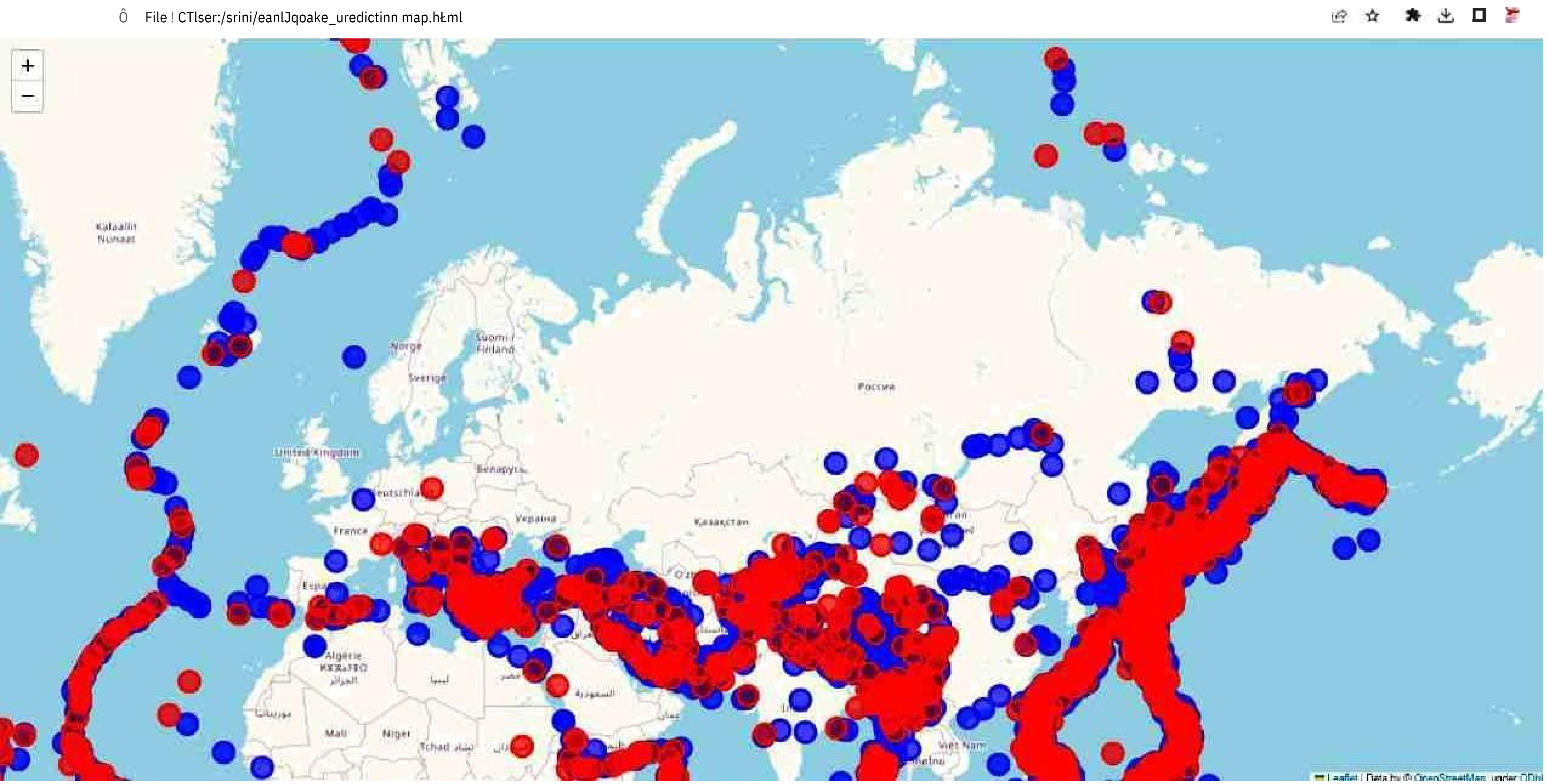


Miercurea Ciuc
Vlăhița
Dărmănești
Târgu Ocna
Onești
Slănic-Moldova
Băile Tușnad
Baraolt
Rupea
Târgu Secuiesc
Sfântu Gheorghe
Covasna
Covasna
Vrancea
Odobești
Adjud
Mărășești
Tecuci
Focșani
Codlea
Brașov
Săcele
Întorsura Buzăului
Râșnov
Zărnești
Predeal
Azuga
Bușteni
Sinaia
Nehoiu
Pătârlagele
Buzău
Buzău
Râmnicu Sărat
Comarnic
Slănic
Vălenii de Munte
Breaza
Câmpina
Prahova
Fieni
Pucioasa
Băicoi
Boldești-Scăeni
Urlați
Mizil
Moreni
Bârlad
Murgeni
Berești
Târgu Bujor
Galați
Galați
Brăila
Brăila
Ianca
Făurei
Măcin
Isaccea
Tulcea
Tulcea
Cahul
Raionul Cahul
Raionul Cantemir
Raionul Taraclia
Vulcănești
Comrat
Congaz
Ceadîr-Lunga
Găgăuzia / Gagauziya / Гагаузия
Izmail
Leaflet | Data by © OpenStreetMap, under ODbL



Norge
Suomi / Finland
Sverige
Oslo
Санкт Петербург
Helsinki
Stockholm
Eesti
Россия
Latvija
Danmark
United Kingdom
Great Britain
Lietuva
Москва
Éire / Ireland
Hamburg
Berlin
Беларусь
Nederland
London
Polska
Астана
Deutschland
Česko
Київ
Paris
Україна
Қазақстан
44.888,
22.407
France
Chișinău
Zagreb
Barcelona
España
Portugal
Oʻzbekiston
Rabat


+
−
Leaflet | Data by © OpenStreetMap, under ODbL


File ! CTlser:/srini/eanlJqoake_uredictinn map.hŁml
+
−
Kalaallit
Nunaat
Norge
Suomi /
Finland
Sverige
Россия
United Kingdom
Україна
Казахстан
France
Algérie
Mali
Niger
Tchad تشاد
Viet Nam
Leaflet | Data by © OpenStreetMap, under