

ARTIFICIAL INTELLIGENCE

PHASE-4 SUBMISSION

EARTHQUAKE PREDICTION USING PYTHON

By this process, we can visualize earthquake data on a world map and split the data for machine learning. Begin by installing Folium with 'pip', then import Pandas and Folium. Load your earthquake dataset using Pandas, specifying your data source. Create an interactive map with Folium, positioning it according to your requirements. Customize map markers, each representing an earthquake event, with details like magnitude and depth. Save the map as an HTML file. After this, open the HTML file in a web browser to view the data on a world map. For machine learning, import 'train_test_split' from scikit-learn and define your features and target variable. Utilize 'train_test_split' to divide your data into training and testing sets, facilitating model development and evaluation.

Preprocess data in Python – Step by step:

1. Install Folium
2. Import Necessary Libraries
3. Load Earthquake Data
4. Create a Map Object
5. Add Markers to the Map
6. Save the Map to an HTML File
7. View the Map

Splitting the Data into Training and Testing Sets:

1. Import Necessary Libraries
2. Define Features and Target Variable
3. Split the Data

1. Visualizing the Data on a World Map:

Install Folium: To use Folium, you need to install it using pip.

```
pip install folium
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.10/dist-packages (0.14.0)
Requirement already satisfied: branca>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from folium) (0.6.0)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from folium) (3.1.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from folium) (1.23.5)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from folium) (2.31.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->folium) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->folium) (2023.7.22)
```

2. Import Necessary Libraries:

Import the required Python libraries, including Folium and Pandas.

```
import folium
import pandas as pd
```

3. Load Earthquake Data:

Load your earthquake dataset using Pandas. Replace 'earthquake_data.csv' with the actual path to your dataset.

```
earthquake_data = pd.read_csv('earthquake_data.csv')
```

4. Create a Map Object:

You need to create a map object. You can specify the starting location and zoom level according to your preference.

```
[14] for index, row in earthquake_data.iterrows():
      folium.Marker(
          location=[row['Latitude'], row['Longitude']],
          popup=f"Magnitude: {row['Magnitude']}, Depth: {row['Depth']}",
          icon=folium.Icon(color='red' if row['Magnitude'] >= 6 else 'green')
      ).add_to(earthquake_map)
```

5. Add Markers to the Map:

Loop through your earthquake dataset and add a marker for each earthquake event. You can customize the marker's appearance, popup information, and color based on earthquake magnitude.

```
[15] earthquake_map.save('earthquake_map.html')
```

6. Save the Map to an HTML File:

Save the map with all the markers to an HTML file that can be opened in your web browser.

```
[7] from sklearn.model_selection import train_test_split
```

7. View the Map:

Open the generated 'earthquake_map.html' file in your web browser to view the earthquake

data on a world map.

```
✓ [8] x = earthquake_data[['Latitude', 'Longitude', 'Depth']] # Adjust columns as needed  
    y = earthquake_data['Magnitude']
```

Splitting the Data into Training and Testing Sets:

1.Import Necessary Libraries:

Import the scikit-learn library for data splitting.

```
✓ [8] x = earthquake_data[['Latitude', 'Longitude', 'Depth']] # Adjust columns as needed  
    y = earthquake_data['Magnitude']
```

2.Define Features and Target Variable:

Determine the features (X) and the target variable (y) for your earthquake prediction model. Typically, you'd use attributes like 'Latitude,' 'Longitude,' 'Depth,' etc., as features, and 'Magnitude' as the target variable.

3.Split the Data:

Use the “train_test_split” function to split your data into training and testing sets. Specify the test size (e.g., 20% for testing) and a random seed for reproducibility.

```
✓ [9] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```