# **Map Visualization And Data Splitting**

I'd be happy to guide you through the process of visualizing earthquake data on a world map and splitting it into training and testing sets. However, since we're in a text-based environment and I can't directly create documents or code, I can provide you with the steps to do this:

1. **Visualizing Data on a World Map:** To visualize earthquake data on a world map, you'll need to use data visualization libraries. Python has some excellent libraries for this, including Folium, Baseman, and Plotly. Here's a general outline of the steps:
   * Install the required libraries, e.g., Folium or Plotly.
   * Load your earthquake data, which typically includes latitude and longitude information.
   * Use the chosen library to create a map and plot the earthquake data points on it.

If you're using Python, you can create Python scripts to achieve this.

1. **Splitting Data into Training and Testing Sets:** To split your earthquake data into training and testing sets, you'll need to use libraries like NumPy and scikit-learn. Here are the steps:
   * Import the necessary libraries.
   * Load your earthquake data.
   * Use scikit-learn's **train\_test\_split** function to split your data into training and testing sets.

Here's some sample code:

python Copy code

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

# Load your data into X (features) and y (labels/targets).

X, y = load\_earthquake\_data()

# Split the data into training and testing sets (e.g., 80% for training and 20% for testing).

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

Make sure to replace **load\_earthquake\_data()** with your data loading function.

You can then continue building your earthquake prediction model using these training and testing datasets. If you need specific code or more detailed guidance, please provide more information about the tools or programming language you're using.

import folium

import pandas as pd

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

# Load earthquake data from a CSV file (replace 'earthquake\_data.csv' with your file).

earthquake\_data = pd.read\_csv('earthquake\_data.csv')

# Create a map centered at a specific location (e.g., coordinates of a city or region).

m = folium.Map(location=[0, 0], zoom\_start=3)

# Plot earthquake data on the map.

for index, row in earthquake\_data.iterrows():

folium.CircleMarker(

location=[row['Latitude'], row['Longitude']],

radius=5,

color='red',

fill=True,

fill\_color='red'

).add\_to(m)

# Save the map to an HTML file for visualization.

m.save('earthquake\_map.html')

# Split data into training and testing sets.

X = earthquake\_data[['Latitude', 'Longitude']]

y = earthquake\_data['Magnitude']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print("Training data shape:", X\_train.shape, y\_train.shape)

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

In this code:

1. Replace ‘earthquake\_data.csv’ with the path to your earthquake data file.
2. Adjust the map center location and zoom level accordingly.
3. It creates a Folium map with circle markers for each earthquake data point and saves it as an HTML file for visualization.
4. The data is split into training and testing sets using scikit-learn’s` train\_test\_split’ function, and the shapes of the resulting datasets are printed.

Make sure to have Folium and scikit-learn installed in your Python environment. You can customize this code to suit your specific dataset and requirements.