**PROJECT IDEA AND FUTURE WORKS**

**1.Project Idea:**

I want to create a general plot class which can handle the most common format of geospatial data like **csv, tsv, vector(.json,kml,shapefile), raster(geotiff)** . This is especially useful for torchgeo especially in scenarios like:

* `Plot` class makes it easier to visualize various data types (satellite imagery, vector data, multi banded data etc.,) allowing the user to work with the visualized raw and processed geospatial data for further procedures.
* This allows to perform quick data transformations (normalization, equalization etc) as the model is getting trained.

For now, I have created a main plot class for csv, tsv, geotiff format which I attached in the chat. Here are some important features that describe the structure of the three plot classes individually.

* csv\_plot: In the given code, there is a class named `DataPlotter` which allows interactive visualization of data provided in the CSV format in the form of various types of plots (1D, 2D, and 3D). It uses a Pandas DataFrame to first load the data and then the user decides what kind of plot to create. The user can choose the type of the plot to be created from a chart, such as a histogram, spline, box, scatter, bar, heatmap, 3D scatter, or surface plots among others. The user can also select which columns to plot, configure the data to be within certain ranges, and specify the title of the plots. The code also makes sure that the users entered valid options, processes the data according to the selections made and then the plots are generated and displayed using the Matplotlib and Seaborn libraries. The program provides the user with a graphical input in order to visualize the data in accordance to user needs.
* tsv\_plot: The same structure goes with the tsv\_plot. But, the code is made flexible to handle tsv files.
* geotiff\_plot: This implementation contains a complete package for visualization and processing of GeoTIFF files which encompass normalization, histogram equalization and multi-band stacking. The main structure is composed of functions that allow to explain auxiliary concepts such as re-scaling raster data, histogram modifying, and simple displaying of GeoTIFF files, be them single bands or multi-band composites. The interacting function `display\_geotiff` enables for display of the raster data with additional parameters and options such as whether the bands should go through normalization or histogram equalization. The code also allows for the combining of a number of single-band geotiff images into one image where it will be displayed as a multi-band image. Files, folders or their combinations and parts can be displayed as complete images or in a graph format whereby images or certain bands can be stacked, equalized or distinctly displayed. The program is easy to use since it allows a great deal of options including viewing the data in turn at the user which also allows the user to interact and view different transformations of the data.

**2.Future work:**

* To extend for other most common geospatial data formats: vector data (.json, kml, shapefile), hdf5, netcdf.
* Also extend for other features in the code like interactive visualisation like folium maps.
* Also, to ensure that the codes work for intersection and union part between different data formats.
* Also, to make all the code for data formats in a single main plot class, if possible and create a **dataviz toolkit.**

**Comments/opinions:** I look forward for your valuable comments or suggestions on this idea as well as developing it in a particular direction.