**Hemanta Pokharel**

**Dr. Lei Wang**

**GEOG-4057**

**May 6, 2025**

**Project-2**

**Introduction**

Manually collecting elevation data for water boundaries is time-consuming. This project solves this problem by automating the process—fetching elevations from GEE, updating shapefiles, and providing an ArcGIS tool. It uses a CSV file containing water boundary locations taken from a GeoTIFF file named flood\_2class.tif. The goal is to write a Python script that fetches elevation data for these points from Google Earth Engine and saves the values to a shapefile. Finally, an ArcGIS toolbox will be created to give users an easy way to interact with the tool.

**Methods and Results**

**Step-1: Importing and Authenticating Google Earth Engine**

This step imports the Earth Engine (ee) library, authenticates the user with the GEE by signing in via browser and finally initializes the GEE session with a specific project (vegetation—sav).

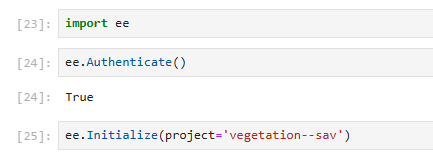


Figure 1: Importing and authenticating Google Earth Engine

**Step-2: Loading Elevation Data from GEE**

This step loads the USGS 3DEP 10m resolution dataset. And retrieve the metadata about the dataset (e.g., coordinate system, resolution).

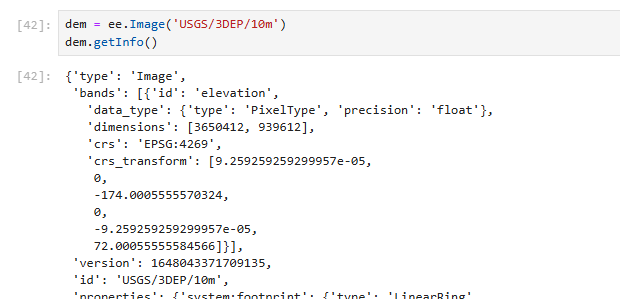


Figure 2: Loading Elevation Data from GEE

**Step-3: Elevation Extraction**

This code creates a geometry at coordinates [-91.0989573, 30.3529013], converts the point into a feature collection, and extracts the elevation value for the point using sampleRegions() and prints the result

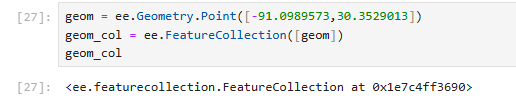


Figure 3: Elevation Extraction for a Single Point

**Step-4: Reading Water Boundary Data from CSV**

The code in this step reads the CSV file (boundary.csv) containing the water boundary points.



Figure 4: Converting CSV Data to a Shapefile

**Step-5: Converting CSV Data to a Shapefile**

This code converts the CSV data into a GeoDataFrame with point geometries. It sets the coordinate system that matches the raster and saves the point as a shapefile (boundary.shp).

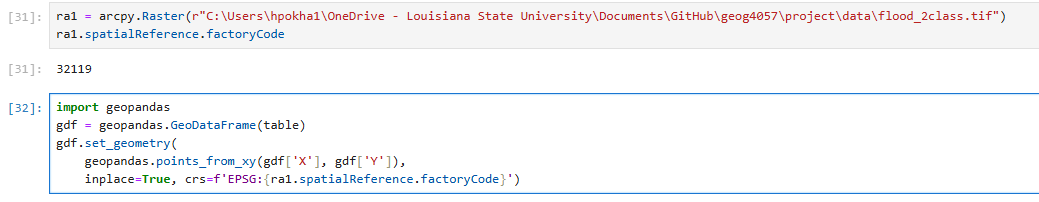


Figure 5: Converting CSV to a Shapefile

**Step-6: Adding an Elevation Field to the Shapefile**

This code uses ArcPy to add a new field (elevation) to the shapefile to store elevation values.

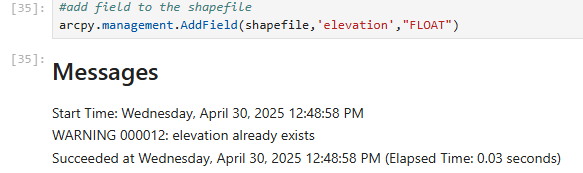


Figure 6: Adding an Elevation Field to the Shapefile

**Step-7: Extracting Elevation Values from GEE and Updating the Shapefile**

This code iterates over each point in the shapefile using a ArcPy UpdateCursor. For each point it extracts the (X,Y), queries GEE for the elevation at that point. Updates the elevation field in the shapefile with the retrieved value.

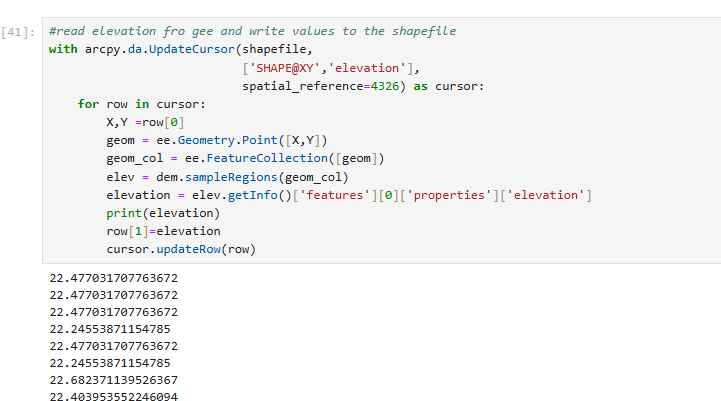
****

Figure 7: Extracting Elevation Values from GEE and Updating the Shapefile

**Creating a Function and using it in Python Toolbox**

Now, the above workflow is converted into a reusable function “get\_elevation\_from\_gee” in the csv\_gee.py. Then the python toolbox is created. This toolbox takes the CSV points and convert it to the shapefile, queries GEE for elevations in batch and updates the shapefile with elevation values. As a result, a shapefile with an elevation field is populated by a Google Earth Engine Data. See figure 8 and 9.

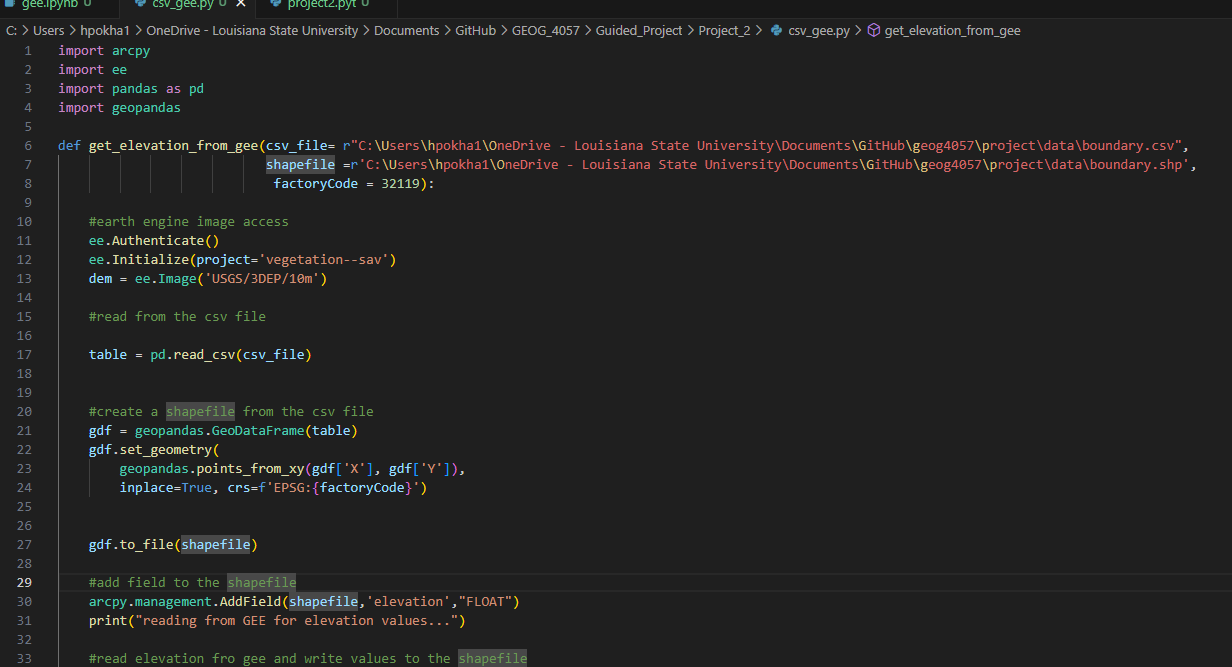


Figure 8: Creating a function

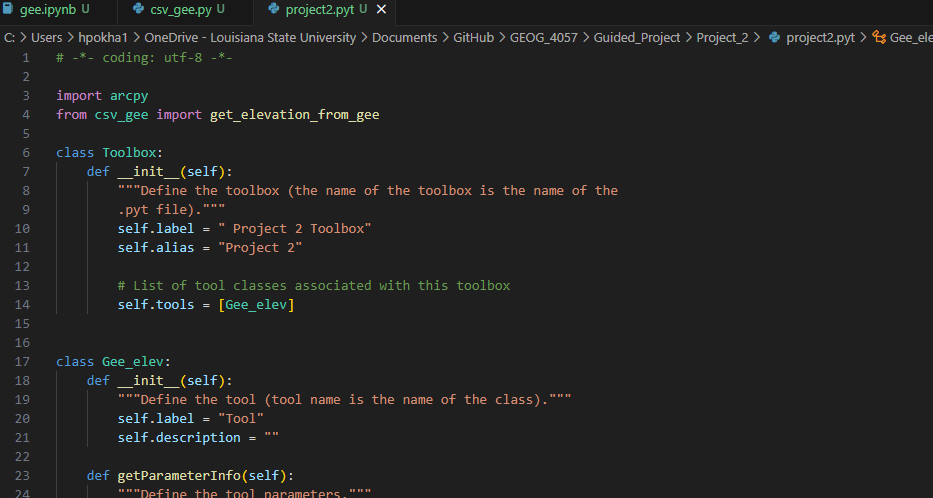


Figure 9: Creating a toolbox

**Running a toolbox**

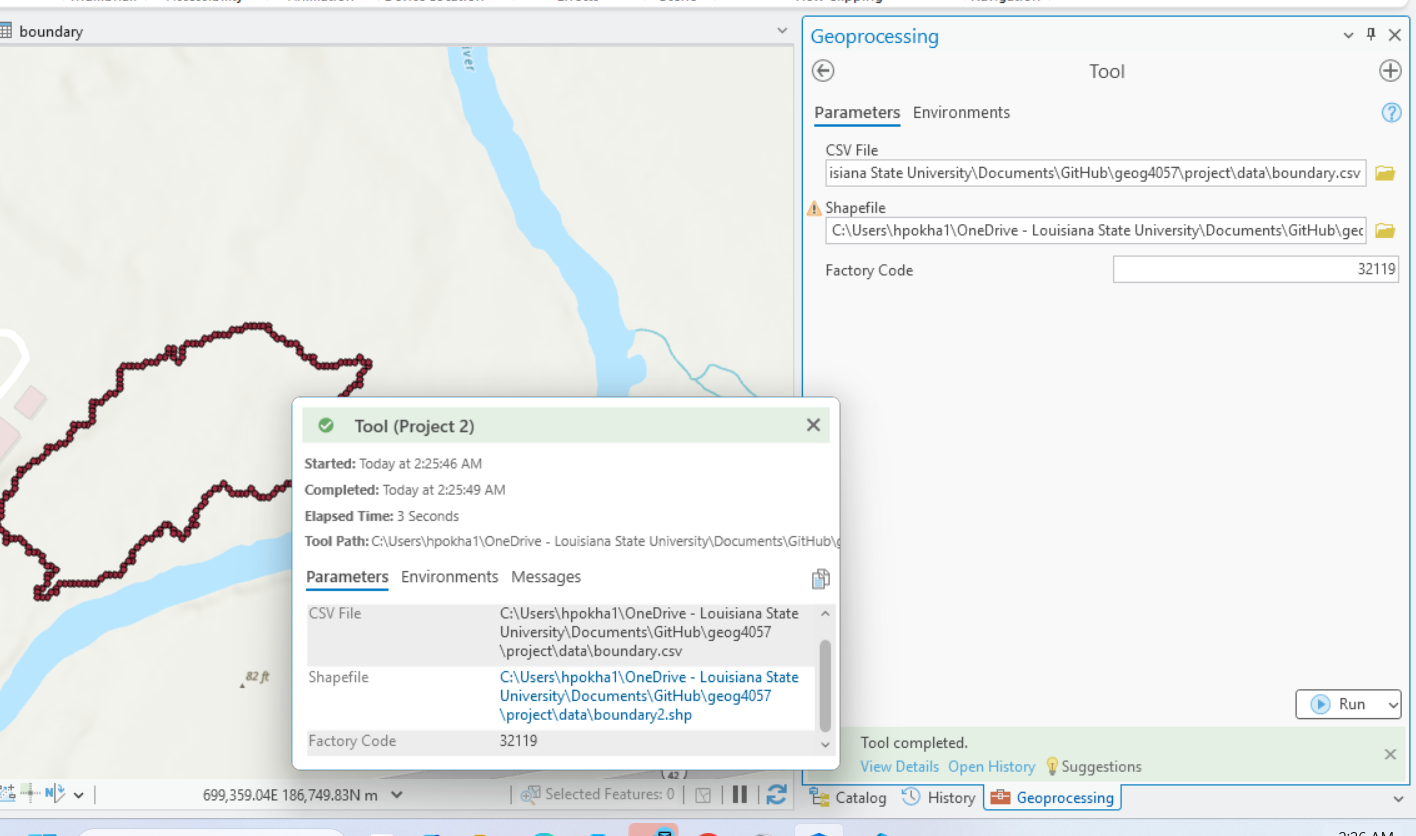


Figure 10: Reference for a successful run of the toolbox

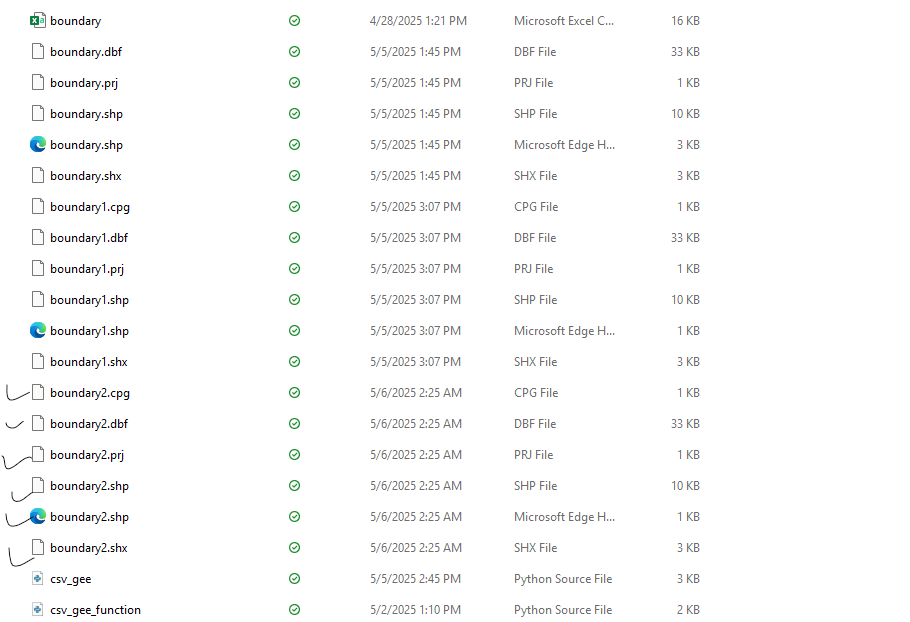


Figure 11: Files created from the toolbox.

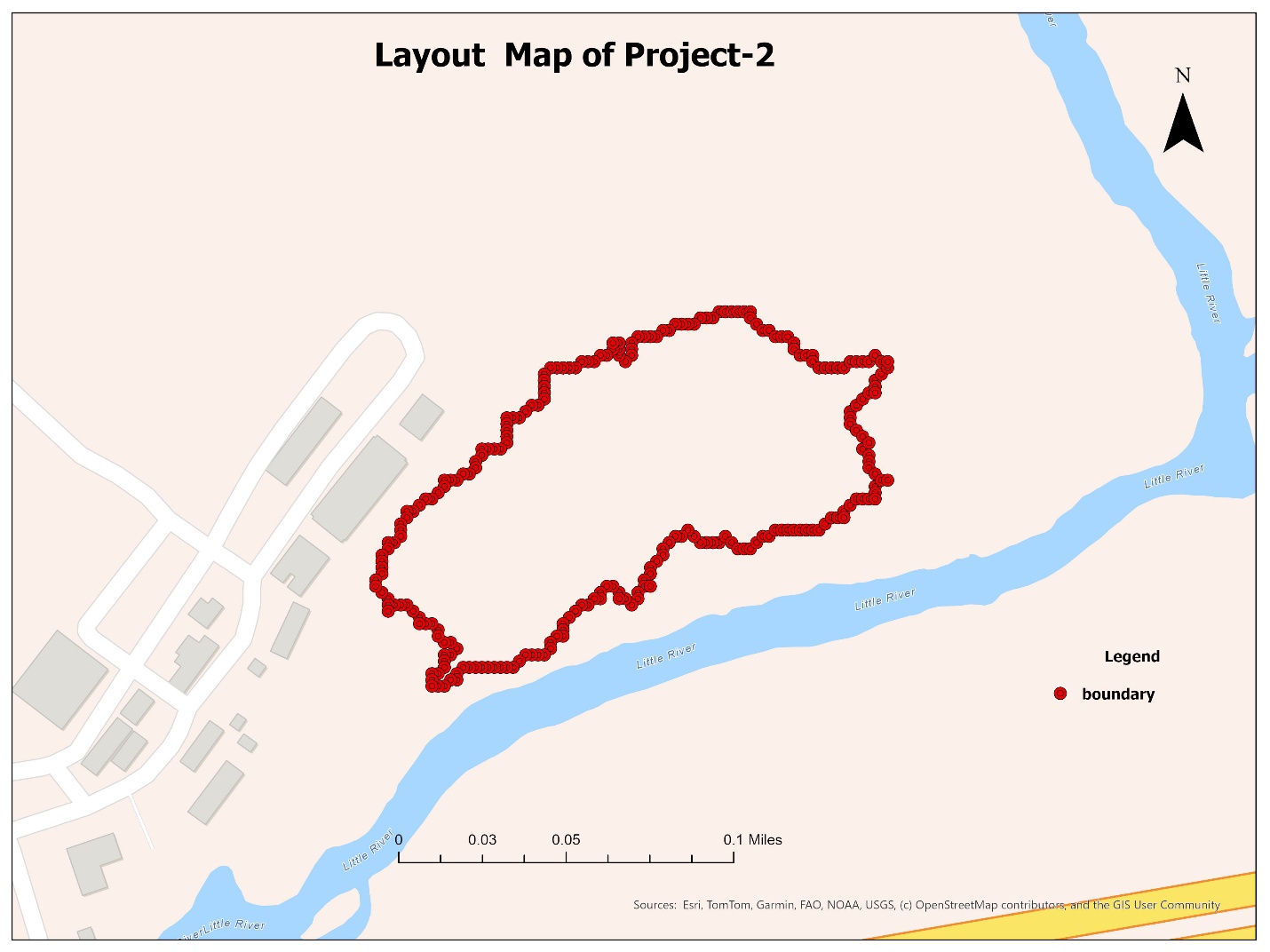


Figure 12: Layout map of the output of the Project-2

**Conclusion**

The project successfully developed a python toolbox that automates the elevation data extraction for the boundary using Google Earth Engine and ArcGIS. The workflow converts the CSV data to shapefile, retrieves USGS 3DEP elevations, and packages the process into an ArcGIS toolbox.