Lab Report

Title: Lab0 – GitHub, ArcPro, Jupyter Notebooks, and ArcOnline

Notice: Dr. Bryan Runck
Author: Rob Hendrickson
Date: 9/21/2022

Project Repository: https://github.com/RwHendrickson/GIS5571/tree/main/Lab0
Google Drive Link: *<if applicable with data, notebooks, etc.>*
Time Spent: 5 hours

Abstract

This lab will go through the process of creating a buffer on a road network using three methods in the ESRI ecosystem: ArcPro, Jupyter Notebooks in ArcPro, and Jupyter Notebooks in ArcOnline.

Problem Statement

The Esri ecosystem has many different ways to access the same underlying functionality. In this lab, we will compare and contrast the same simple activity, buffering a road network using three different tools: ArcPro, Jupyter Notebooks in ArcPro, Jupyter Notebooks in ArcOnline.

Table 1. Brief description of requirements used in this lab.

#	Requirement	Defined As	(Spatial) Data	Attribute Data	Dataset	Preparation
1	Municipal Boundaries	Raw input dataset from Metropolitan Council	Polygons of Cities and Townships (CTU) geometries	CTU Name, CTU Type, CTU ID, etc	Mn GeoSpatial Commons	Data filtered to only include Minneapolis
2	Road network	Raw input dataset from MNDOT	Road geometry	Volume, Segment Name, etc.	Mn GeoSpatial Commons	Data was clipped to Minneapolis municipal boundary

Input Data

The datasets acquired for this lab were provided by the Metropolitan Council and Minnesota Department of Transportation (MNDoT). Both are in the NAD83 UTM 15N (EPSG:26915) coordinate reference system.

Counties and Cities & Townships 2010 of Twin Cities Metropolitan Area (Metropolitan Council) contains boundaries of counties, cities, and townships in the 7 county metro. This is assembled for the MetroGIS community.

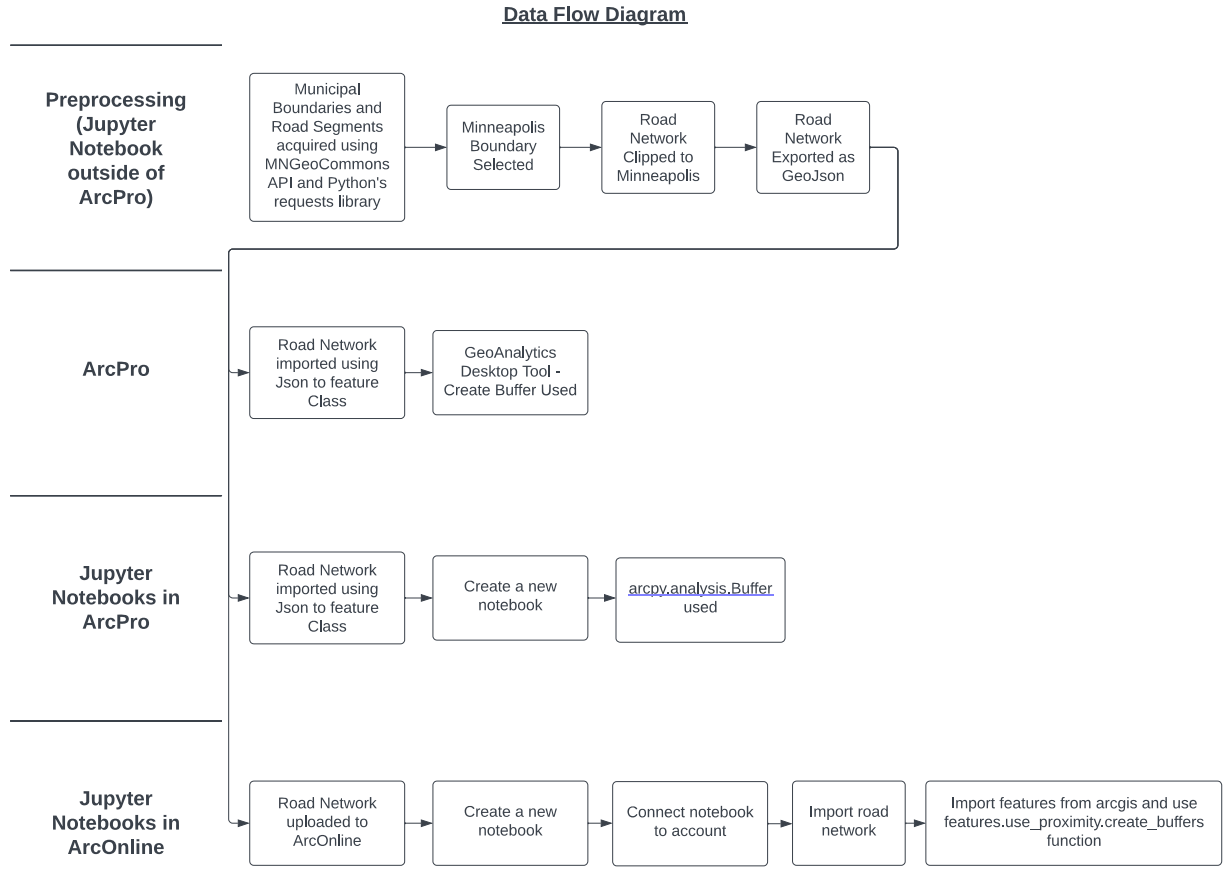
The road network from MNDoT contains the most current Annual Average Daily Traffic (AADT) road segments of the entire state of Minnesota. Often this data is used to assist in annual allocation of financial aid for road maintenance and construction.

Table 2. Table describing the datasets used in this lab.

#	Title	Purpose in Analysis	Link to Source
1	Metropolitan CTUs	Raw input dataset to determine boundary of Minneapolis (from Metropolitan Council)	Mn GeoSpatial Commons
2	Minnesota Roads	Raw input dataset for routing analysis from MNDOT	Mn GeoSpatial Commons

Methods

Figure 1. A data flow diagram of this lab.



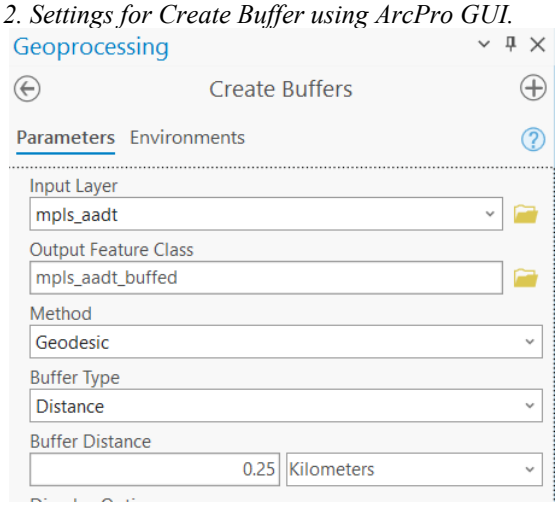
Preprocessing

The preprocessing of the data for this project was done in a previous project of mine and can be found here: https://github.com/RwHendrickson/MappingGZ/tree/main/Prototype/Notebooks/CleaningData under the 'Boundary' and 'Traffic' folders. The data for this project was retrieved using the Python requests library. The Minneapolis municipal boundary was selected from the Metropolitan Council's CTUs dataset and used to clip the road network for the entire state of Minnesota. All of this was performed using the Geopandas library and then exported in a GeoJson format.

ArcPro Method

The preprocessed geojson was imported into ArcPro using the JSON to Features conversion tool. Once here, the GeoAnalytics Desktop Tool, Create Buffers, was used to create 250 meter buffers around all road segments. The parameters can be seen in figure 2.

Figure 2. Settings for Create Buffer using ArcPro GUI.



Jupyter Notebook in ArcPro Method

The preprocessed geojson was imported into ArcPro using the JSON to Features conversion tool. Once here, a new Jupyter Notebook was initialized and the Buffer function from Arcpy's analysis module was used. The parameters can be seen in figure 3.

Figure 3. Creating a buffer using Jupyter Notebooks in ArcPro.

```
In [1]:  # This is creating a buffer in ArcPro Notebooks

         arcpy.analysis.Buffer(in_features = 'mpls_aadt',
                               out_feature_class='mpls_aadt_buffed_nb',
                               buffer_distance_or_field='250 Meters',
                               method='Geodesic')
```

```
Out[1]:
```

Messages

Start Time: Wednesday, September 21, 2022 10:09:03 PM

Succeeded at Wednesday, September 21, 2022 10:09:09 PM (Elapsed Time: 6.75 seconds)

Jupyter Notebook in ArcOnline Method

The preprocessed geojson was uploaded onto ArcOnline. Once here, a new Jupyter Notebook was initialized and connected to a user account. Then the road network was imported into the notebook and create_buffer function from arcgis's features.use_proximity module was used. The parameters can be seen in figure 4.

Figure 4. Creating a buffer using a Jupyter Notebook on ArcGIS Online.

```
In [13]:
# Added Road Network
# Title: mpls_aadt | Type: Feature Service | Owner: hendr862_UMN
mpls_aadt = gis.content.get("cb436450fa6845bbbcd855678ead881b")

In [26]:
from arcgis import features

features.use_proximity.create_buffers(input_layer = mpls_aadt,
                                      output_name = 'mpls_aadt_buffed',
                                      distances = [250], units = 'Meters')

Out[26]:
```

mpls_aadt_buffed

Feature Layer Collection by hendr862_UMN
Last Modified: September 22, 2022
0 comments, 0 views

Results

Each of the methods were able to create the desired buffers of the road network. However, each had it's own pros and cons (table 3).

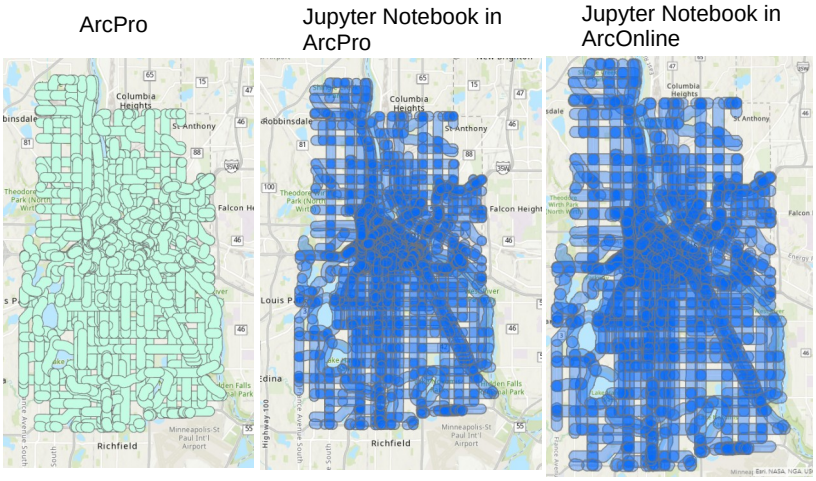
Table 2. Table describing the pros and cons of each method.

Method	Speed	Number of Steps	Need Software on Local Machine	Documentation (rating 0 - 5)	Reproducible & Scalable
ArcPro	6 seconds	6	Yes	4	No
Notebook in ArcPro	6 seconds	7	Yes	4	Yes
Notebook in ArcOnline	20 seconds	9	No	3	Yes (but costs money?)

Results Verification

Visually all the methods appear to have produced the same results (figure 5). They also contained the same number of elements.

Figure 5. Visualization of results across methods.



Discussion and Conclusion

GitHub:

Getting my repository set up for this course wasn't too difficult as I have used Github in previous projects.

Method Comparison:

In reflecting upon these various methods, I would be more inclined to use Jupyter Notebooks through ArcPro moving forward. ArcPro alone involves too much button pushing in my opinion and ArcOnline's credit system freaks me out a little (though it does provide space for some serious collaboration!).

My preference to date for doing GIS work has been to use Jupyter Notebooks on their own with FOSS code, however, I'm beginning to understand the power of certain tools that have been developed by ESRI and have also grown more familiar with their GUI this semester. Their arcpy syntax and way of managing spatial data is going to take some getting used to, but I'm excited to see how automation and ArcPro can work in concert together!

References

N/A

Self-score

Category	Description	Points Possible	Score
Structural Elements	All elements of a lab report are included **(2 points each)**: Title, Notice: Dr. Bryan Runck, Author, Project Repository, Date, Abstract, Problem Statement, Input Data w/ tables, Methods w/ Data, Flow Diagrams, Results, Results Verification, Discussion and Conclusion, References in common format, Self-score	28	**28**
Clarity of Content	Each element above is executed at a professional level so that someone can understand the goal, data, methods, results, and their validity and implications in a 5 minute reading at a cursory-level, and in a 30 minute meeting at a deep level **(12 points)**. There is a clear connection from data to results to discussion and conclusion **(12 points)**.	24	**12**
Reproducibility	Results are completely reproducible by someone with basic GIS training. There is no ambiguity in data flow or rationale for data operations. Every step is documented and justified.	28	**21**
Verification	Results are correct in that they have been verified in comparison to some standard. The standard is clearly stated **(10 points)**, the method of comparison is clearly stated **(5 points)**, and the result of verification is clearly stated **(5 points)**.	20	**10**
		100	**71**