BUILDING AN ACCESSIBLE MACHINE LEARNING WORKFLOW FOR GEOSPATIAL ANALYSIS

Nathalie Redick, Matthew Tarling

Dept. of Earth and Planetary Sciences, McGill University

MOTIVATIONS

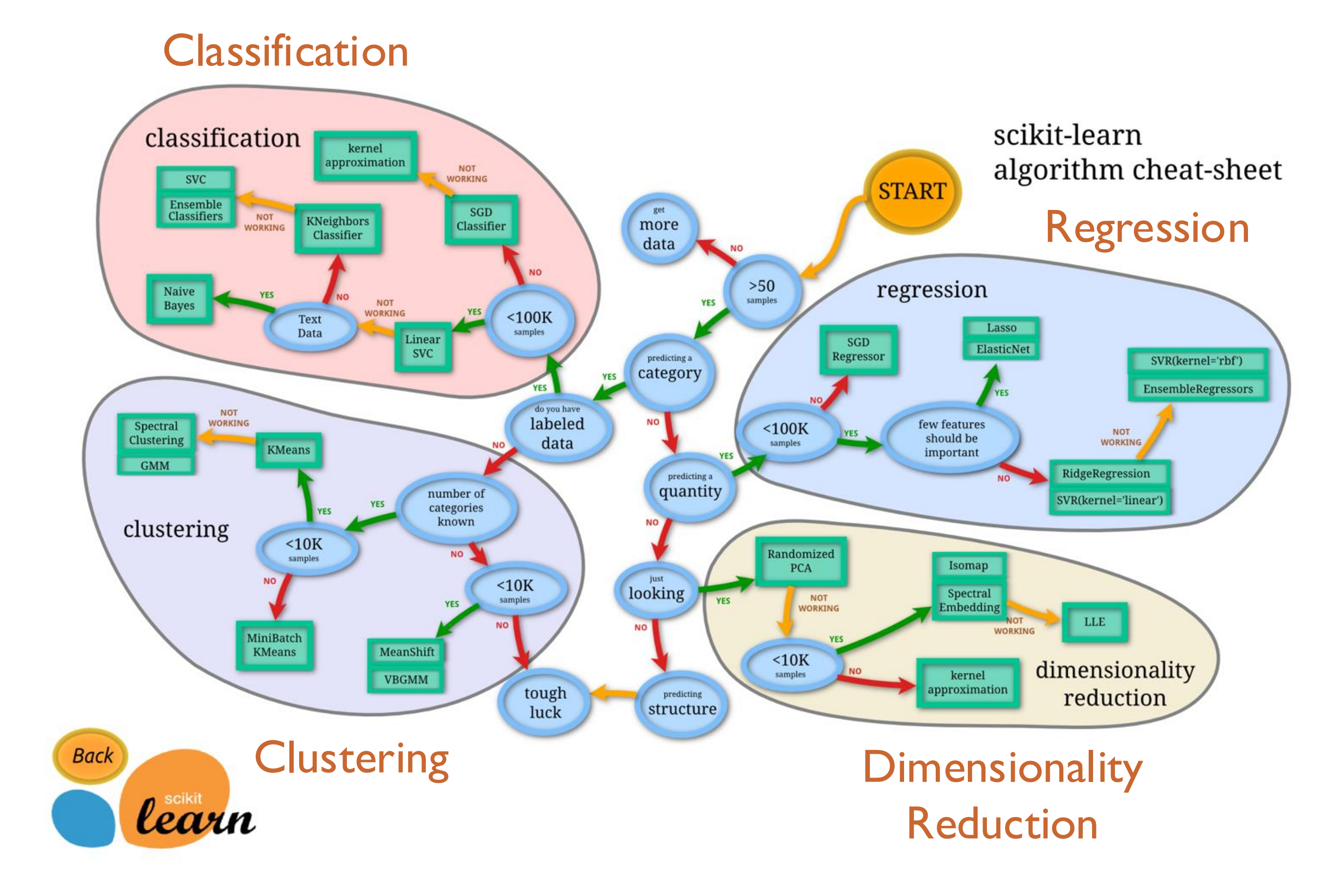
- Machine learning (ML) is becoming an increasingly popular tool across many fields, but is currently under-utilised in geospatial analysis
- ML can support research by speeding up complex calculations, providing new insights, automating tedious tasks, etc.
- Geospatial analysis relies on high resolution and high dimensional data that is time and resource intensive to process → ML can help

BARRIERS TO ML FOR GEOSCIENCE

1. Programming skill (Python).
   - Python is currently the most popular language for scientific ML applications with the most documentation and support
2. Understanding and applying ML algorithms and best practices .
   - How do you pick which model to use?
3. Lack of code libraries to support geospatial ML.
   - Geospatial data is typically provided in formats that requires a unique approach to data manipulation and processing
4. Access to computing resources.
   - Papers often tout models that were trained for long periods of time on remote super computers, which can be very expensive
5. Geospatial analysis is inherently visual, coding is not.

MOTIVATIONS

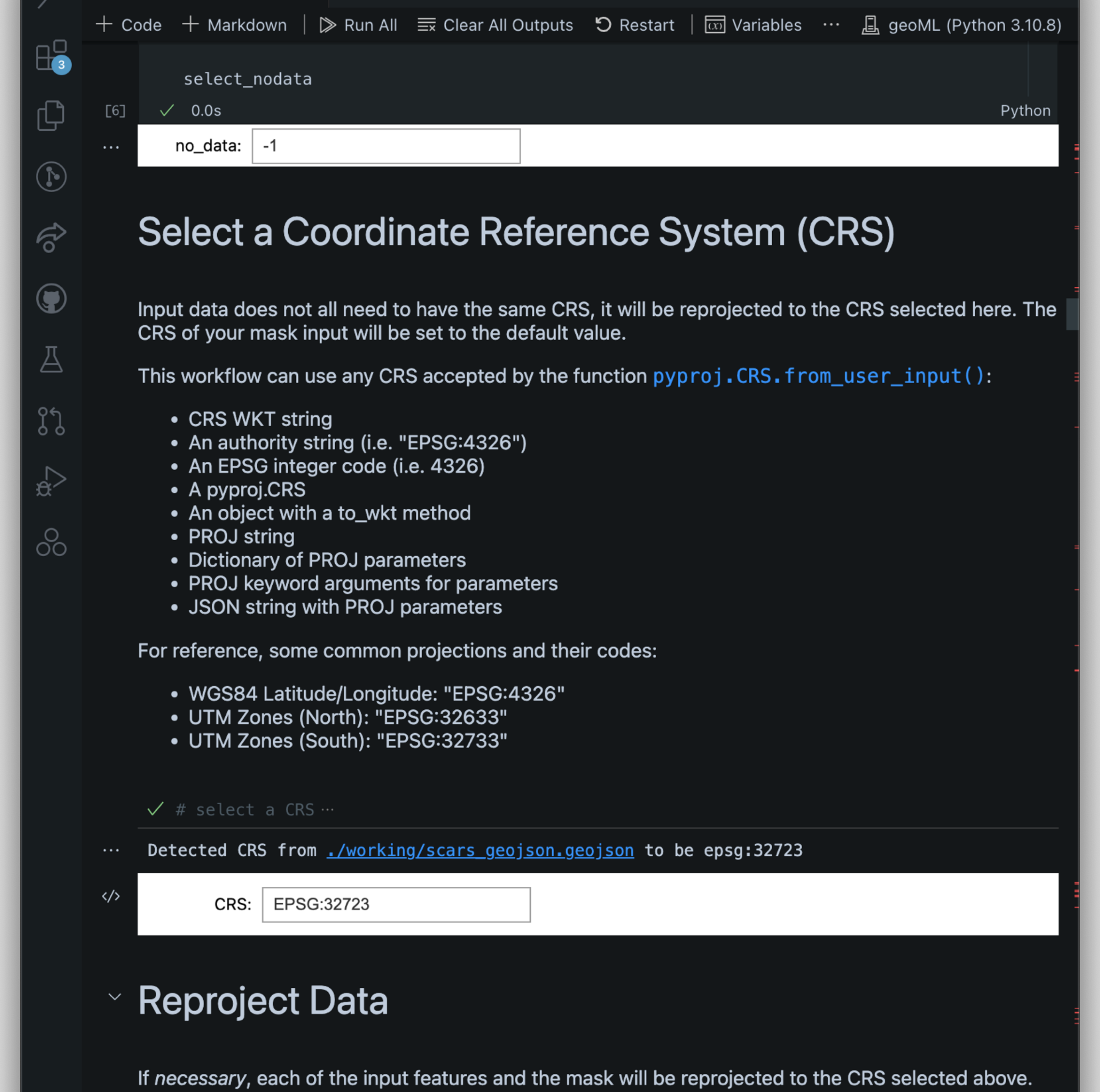
- Machine learning (ML) is currently under-utilised in geospatial analysis, but is currently under-utilised in geospatial analysis
- ML can support research by speeding up complex calculations, providing new insights, automating tedious tasks, etc.
- Geospatial analysis relies on high resolution and high dimensional data that is time- and resource- intensive to process



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

BARRIERS TO ML FOR GEOSCIENCE

- Objective: design a ML workflow that walks users through data input, data preprocessing, model training, interpretation, and deployment
- Provide custom functionality for geospatial contexts
- Provide the source code as a freely accessible Google Colab notebook for use & modification
- Most mainstream ML libraries lack functionality for working with geospatial data
- Applications such as ArcGIS have some geospatial ML capabilities, but the methodology is unclear and the software is not open source



```
select_nodata
```

[6] ✓ 0.0s Python

no_data: -1

Select a Coordinate Reference System (CRS)

Input data does not all need to have the same CRS, it will be reprojected to the CRS selected here. The CRS of your mask input will be set to the default value.

This workflow can use any CRS accepted by the function `pyproj.CRS.from_user_input()`:

- CRS WKT string
- An authority string (i.e. "EPSG:4326")
- An EPSG integer code (i.e. 4326)
- A pyproj.CRS
- An object with a to_wkt method
- PROJ string
- Dictionary of PROJ parameters
- PROJ keyword arguments for parameters
- JSON string with PROJ parameters

For reference, some common projections and their codes:

- WGS84 Latitude/Longitude: "EPSG:4326"
- UTM Zones (North): "EPSG:32633"
- UTM Zones (South): "EPSG:32733"

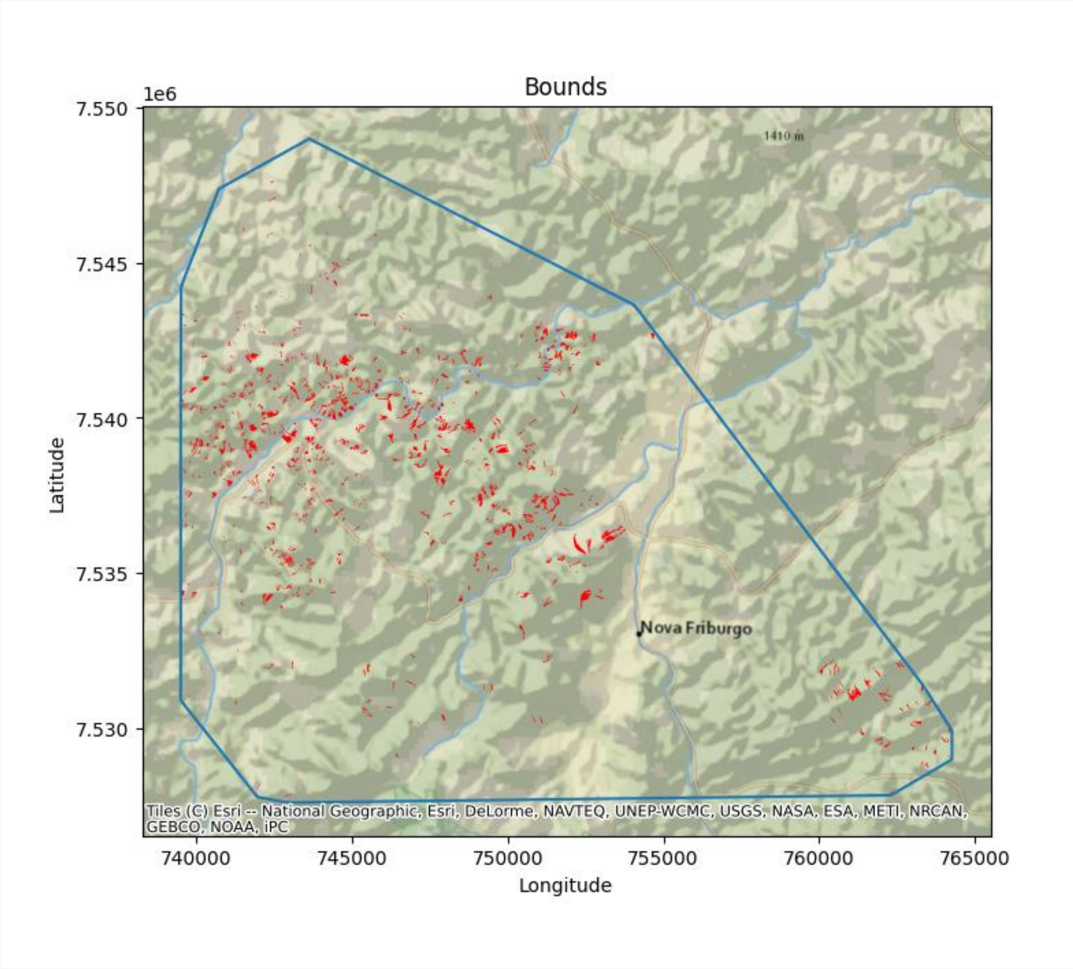
```
# select a CRS ···
```

```
Detected CRS from ./working/scars_geojson.geojson to be epsg:32723
```

CRS: EPSG:32723

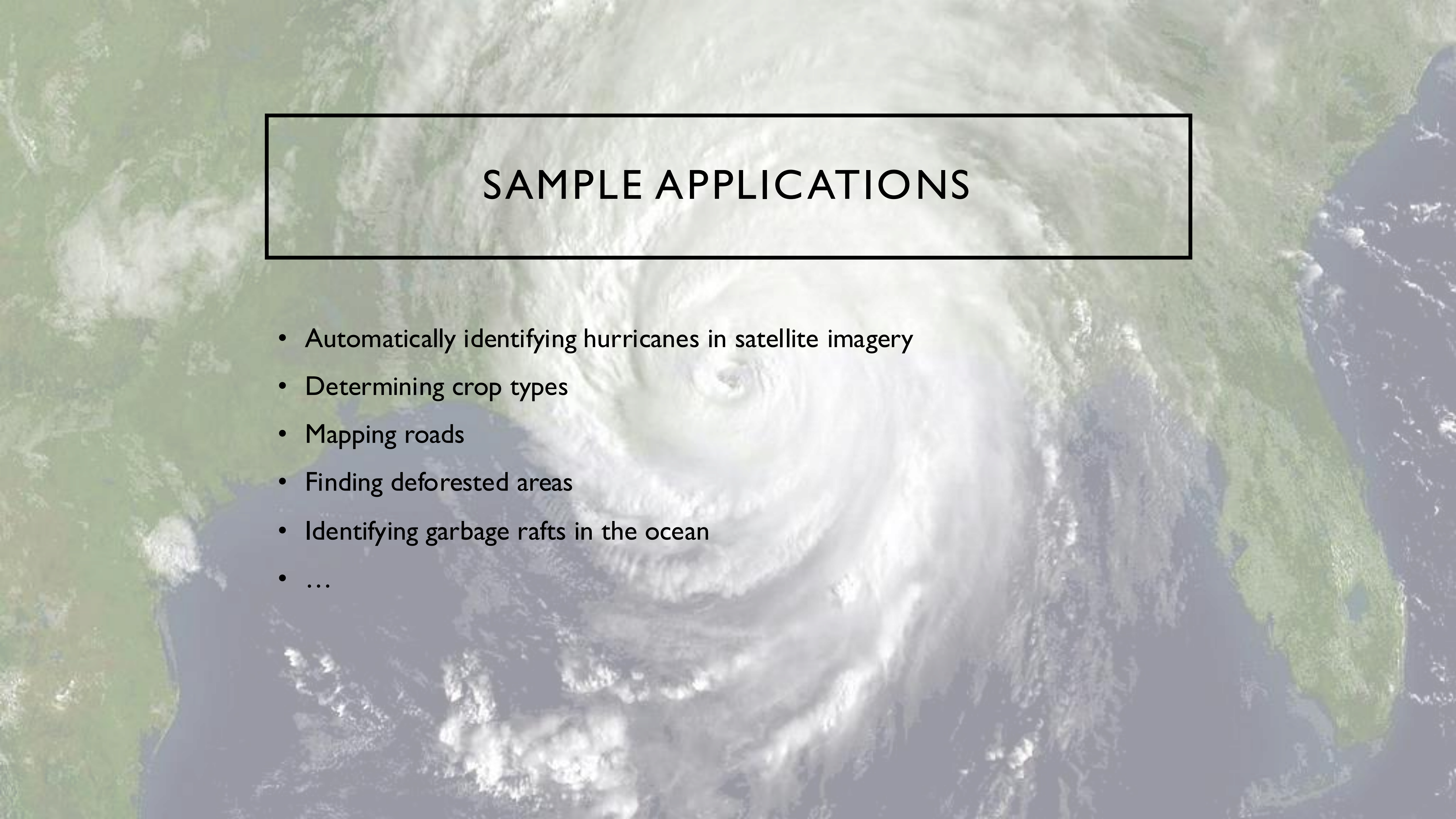
Reproject Data

If *necessary*, each of the input features and the mask will be reprojected to the CRS selected above.


Bounds
1e6
7.550
7.545
7.540
7.535
7.530
Latitude
740000
745000
750000
755000
760000
765000
Longitude
1410 m
Nova Friburgo
Tiles (C) Esri -- National Geographic, Esri, DeLorme, NAVTEQ, UNEP-WCMC, USGS, NASA, ESA, METI, NRCAN, GEBCO, NOAA, iPC

OPEN & ACCESSIBLE

- The code is written in Python and can be worked with a local set up as a Jupyter notebook or in a Google Colab notebook, *provided via Github*
- Text guidance and instructions are provided alongside the code
- Interactive widgets allow the user to directly interact with the data and code
- Graphs & plots provided along the way so the user can visually check that everything is running smoothly



SAMPLE APPLICATIONS

- Automatically identifying hurricanes in satellite imagery
- Determining crop types
- Mapping roads
- Finding deforested areas
- Identifying garbage rafts in the ocean
- …

THANK YOU!

Special thanks to Matthew Tarling and James Kirkpatrick for supervising this research.

We hope our workbook can help introduce open, standardized methods for processing geospatial data and applying machine learning in Python.