

TASK - 4

GIS IN Python

Types of Data:

Vector Data includes points , lines , polygons.

Raster Data includes images, digital elevation models, 2-D fields

GeoJson : Represents simple features in a Json format

```
{"type": "Feature",  
  "geometry": {  
    "type": "point",  
    "coordinate": [125.6, 10.1]},  
  "properties": "Dringar Islands"}}
```

GDAL

GDAL (Geospatial Data Abstraction Library) is the open source Swiss Army Knife of raster formats. It also includes the OGR simple features library for vector formats.

Command for importing : `from osgeo import gdal`

2) `from osgeo import ogr` , GDAL are not very pythonic , Most of its wrappers are in C++ counterparts.

Reading File : `geo = gdal.open(raster_file)`

Then, `drv = geo.GetDriver()`

Result `print(drv.GetMetadataItem(''))`

FIONA : Tools for importing and exporting vector data from various formats.(dictionary for each record) like shapefile.

`import fiona`

`cc = fiona.open('.shp') , rec = cc.next() , rec.keys()`

Fiona command line interface:

Commands: cat Concatenate and print the features of datasets

collect Collect a sequence of features

dump Dump a dataset to GeoJSON

info Print information about a dataset

insp Open a dataset and start an interpreter

load Load GeoJSON to a dataset in another format

Shapely : Shapely is a python library for geometric operations using the GEOS library.

Shapely can perform:

- Geometry validation
- Geometry creation(eg collections)
- Geometry operations
- # Note that this is basically a tool for analyzing 2-dimensional cartesian shapes – it has no facilities for managing projections. That you have to do with PyProj before you start manipulations with shapely.

PyProj : Tools for defining and transforming the datum and projections of spatial data.

Examples :

- EPSG:4326 latitude , longitude in WGS-84 coordinate system
- EPSG:900913 and EPSG:3857 Google spherical Mercator
- EPSG:102718 NAD 1983 StatePlane New York Long Island FIPS 3104 Feet

Create an SRS with pyproj:

```
>>> from pyproj import Proj , transform
>>> p = Proj(init='epsg:32168')
>>> lat , lon = 40.78 , - 73.97
>>> x , y = p(lon, lat)
>>> print(x,y)
(586912.6635 , 4514845.7241347)
```

Rasterio : Tools for importing and exporting raster data from various formats.

```
>>> import rasterio  
>>> with rasterio.open(".tif") as f:  
>>>     img = f.read(1)  
>>> imshow(img, cmap='gray')
```

GeoPandas : Make working with geographic data like working with other kinds of data in python

Work with existing tools

- Desktop GIS(ArchGIS, QGIS)
- Geospatial databases(PostGIS)
- Web maps(Leaflet ,D3,etc)
- Python data tools(pandas , numpy etc)

Geopandas can do

- Geometry operations
- Data alignment
- Coordinate transformation(pyproj)
- Read/Write GIS File formats(fiona)
- Create a GeoDataFrame from PostGIS table
- Output any object as geoJSON
- Plotting

- **Loading Data from shapefiles**

```
>>>boros = GeoDataFrame.from_file('.shp')  
>>>boros.sort()
```

Notebook is the link

<https://www.kaggle.com/raj10gupta/gis-all-in-one>