70 Geospatial Python Libraries

Neha Sharma   July 7, 2023

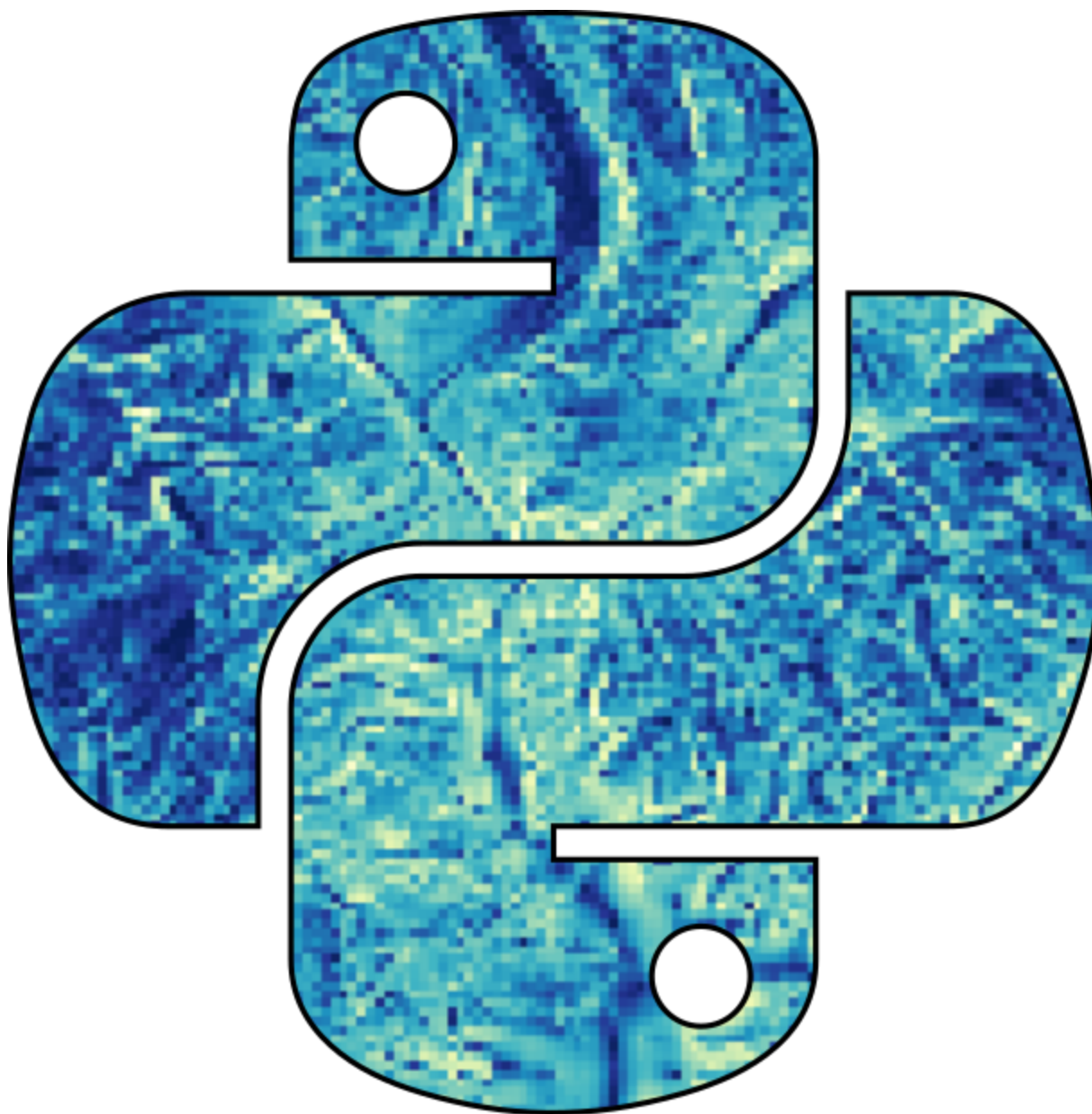Looking to hide highlights? You can now hide them from the "•••" menu.

Image Source —

Python has emerged as a dominant language in the field of Geographic Information Systems (GIS) and remote sensing due to its versatility, extensive library ecosystem, and user-friendly syntax. This compilation of 70 geospatial Python libraries showcases the rich toolkit available for GIS and remote sensing data processing and analysis.

Python's importance in GIS stems from its ability to handle complex geospatial data, ranging from vector and raster formats to point clouds. With Python, GIS professionals can efficiently perform tasks such as data manipulation, spatial analysis, visualization, and automation.

Over the years, Python has evolved to seamlessly work with GIS data through specialized libraries and tools. Python's integration with powerful geospatial libraries like GDAL, Fiona, and Shapely has provided a foundation for reading, writing, and processing spatial data in various formats. Moreover, the emergence of GeoPandas has brought the capabilities of pandas to geospatial data, enabling seamless integration of attribute and spatial operations.

Python's open-source nature and extensive community support have fostered the development of numerous geospatial libraries. These libraries, such as Cartopy, Basemap, and Pyproj, offer specialized functionalities for tasks like map projection, coordinate transformation, spatial analysis, and interactive visualizations. They provide intuitive interfaces and efficient algorithms, empowering GIS professionals to solve complex geospatial problems effectively.

The future of Python in GIS appears promising. With advancements in machine learning and artificial intelligence, Python libraries like ArcGIS API for Python, PyTorch, TensorFlow, and scikit-learn are being applied to geospatial data analysis, enabling tasks such as land cover classification and object detection. Additionally, integrating Python with emerging technologies like remote sensing, SAR, and hyperspectral data processing is expanding the capabilities of remote sensing applications.

Python's flexibility, ease of use, and a vast collection of geospatial libraries make it a preferred choice for GIS professionals, spatial data scientists, researchers, and developers. As the geospatial field continues to evolve, Python's adaptability and extensive library ecosystem will play a pivotal role in driving innovation, enabling advanced analysis techniques, and facilitating the integration of geospatial data with other domains.

This compilation of **70 geospatial Python libraries** demonstrates Python's importance in GIS. Python's evolution to seamlessly work with GIS data, coupled with its extensive library ecosystem, positions it as a powerful tool for geospatial data processing and analysis. With its promising future in emerging technologies, Python continues to shape the future of GIS and remote sensing, enabling advanced geospatial applications and insights.

1. Developed by , this library provides GeoAI capabilities for ArcGIS users.
2. The Earth Engine API allows you to access Google Earth Engine's vast collection of geospatial data and perform analysis tasks using Python.
3. (fast.ai): While not specifically designed for GeoAI, fastai.vision is a popular library for deep learning and computer vision tasks, that can be used for analyzing geospatial imagery.

4. (Esri): arcpy is a Python library provided by Esri for working with geospatial data on the ArcGIS platform. It allows you to automate geoprocessing tasks and perform spatial analysis.
5. (Riverbank Computing): PyQt is a set of Python bindings for the Qt application framework. It can be used to create graphical user interfaces (GUIs) for geospatial applications.
6. : GeoPandas extends the functionality of Pandas, a popular data analysis library, to handle geospatial data. It provides a high-level interface for working with vector data, such as points, lines, and polygons.
7. : Rasterio is a library for reading and writing geospatial raster datasets. It provides efficient access to raster data and allows for various operations, such as reprojecting and resampling.
8. (Open-Source Geospatial Foundation): GDAL (Geospatial Data Abstraction Library) is a powerful library for reading, writing, and manipulating geospatial raster and vector data formats.
9. Fiona is a Python API for reading and writing geospatial data in different formats, including Shapefiles and GeoJSON. It integrates well with other libraries like GeoPandas.
10. Shapely is a library for geometric operations in Python. It allows you to create, manipulate, and analyze geometric objects like points, lines, and polygons.
11. GeoPy is a library for geocoding and reverse geocoding, which allows you to convert addresses to geographic coordinates and vice versa. A library for geocoding and distance calculations. Geopy also provides functionalities for calculating distances between points based on different distance metrics.
12. PySAL (Python Spatial Analysis Library) provides a wide range of spatial analysis methods, including spatial autocorrelation, clustering, and spatial regression.
13. : Pyproj is a Python interface to the PROJ library, which provides coordinate transformation capabilities. It allows you to convert coordinates between different coordinate reference systems (CRS).
14. Rasterstats provides tools for zonal statistical analysis on raster datasets. It allows you to calculate statistics, such as the mean or maximum, within defined zones.
15. Geos is a C++ library that provides low-level geometric operations, and the Python Geos library offers a Python interface to Geos. It is often used in conjunction with Shapely.
16. : RSGISLib is a library for remote sensing and GIS analysis. It provides a range of functionalities, including image classification, change detection, and spectral index computation. It also includes capabilities for working with thermal remote sensing data. It offers functions for thermal image processing, including radiometric correction, land surface temperature estimation, and emissivity retrieval.
17. : WhiteboxTools is a library for geospatial analysis and data processing. It offers a comprehensive set of tools for tasks like terrain analysis, hydrological modeling, and LiDAR data processing.

18. : GeoDjango is a Python framework for developing web applications with geospatial functionality. It integrates with Django, a popular web framework, and provides tools for managing geospatial data and performing spatial queries.
19. : Designed for working with multidimensional geospatial datasets, it provides tools for data manipulation, analysis, and visualization.
20. Integrates geospatial data into SQLAlchemy, a Python SQL toolkit, allowing spatial queries and operations with geodatabase.
21. : scikit-mobility is a library for analyzing and modeling mobility patterns and trajectories.
22. (EarthPy contributors): EarthPy is a library that focuses on earth science data analysis, specifically for remote sensing and geospatial imagery.
23. (HoloViz): GeoViews is a library for creating interactive visualizations of geospatial data using the HoloViews ecosystem.
24. : Geocoder is a simple and easy-to-use library for geocoding and reverse geocoding tasks.
25. : GeoPandas-Bokeh is an extension of GeoPandas that enables interactive geospatial visualizations using Bokeh.
26. : Cartopy is a Python library for geospatial data processing and visualization. It provides a simple and intuitive interface for creating maps and working with geospatial datasets.
27. : Folium is a library for interactive mapping visualizations, allowing users to create and manipulate interactive maps using Python and Leaflet.js.
28. : geoplot is a library for geospatial data visualization, offering a high-level interface for creating various types of maps, including choropleth maps, point maps, and line maps. It's an extension to cartopy and matplotlib which makes mapping easy: like seaborn for geospatial.
29. : PyCRS is a library for working with Coordinate Reference Systems (CRS) in Python. It provides utilities for converting between different CRS formats and performing CRS transformations.
30. : Designed for deep learning applications in remote sensing, it provides tools for preprocessing and analyzing satellite imagery.
31. : OSMnx is a Python library for retrieving, analyzing, and visualizing OpenStreetMap (OSM) data. It allows users to download street networks, buildings, and other OSM features and perform network analysis on them.
32. : Overpy is a Python wrapper for the Overpass API, which is a powerful tool for querying and downloading OpenStreetMap (OSM) data. It allows users to extract specific OSM features based on custom criteria and can be used for various geospatial analyses.
33. : NetworkX is a Python library for the creation, manipulation, and study of complex networks. It can be used to model and analyze spatial networks, such as transportation networks, social networks, or infrastructure networks. NetworkX provides algorithms for network analysis, community detection, and centrality measures, which can be applied to geospatial networks for various purposes.

34. : geospatial-learn is a Python module for using scikit-learn and xgb models with geospatial data, chiefly raster and vector formats.
35. : OGR is a library that provides support for reading and writing geospatial vector data formats. It is part of the GDAL project and allows users to work with various vector data formats such as Shapefile, GeoJSON, and KML.
36. : ipyleaflet is a Python library for interactive mapping visualizations in Jupyter Notebooks and JupyterLab. ipyleaflet is useful for creating interactive geospatial visualizations and web mapping applications.
37. : Pydeck is a high-level library for creating interactive data visualizations on 2D and 3D maps. It is built on top of Deck.gl and provides a simple and intuitive API for creating stunning visualizations that can be easily integrated into Jupyter Notebooks or web applications.
38. : GeoMesa is a geospatial data management and analysis library built on top of distributed computing frameworks like Apache Hadoop, Apache Accumulo, Apache HBase, and Apache Kafka. It allows efficient storage, indexing, and querying of large-scale geospatial data. GeoMesa is designed for handling spatiotemporal data and enables advanced analytics for GeoAI applications, including spatiotemporal indexing and geospatial query optimizations.
39. : RasterFrames is a library for working with geospatial raster data using the DataFrame API of Apache Spark. It enables the integration of raster data with Spark's distributed processing capabilities, making it scalable for large-scale geospatial analysis tasks. RasterFrames allows users to perform distributed raster processing and analysis, leveraging the power of Spark's parallel processing capabilities.
40. : laspy is a Python library for reading, writing, and modifying LIDAR data in the ASPRS LAS format. LIDAR (Light Detection and Ranging) data is commonly used for capturing detailed 3D information about the Earth's surface. With laspy, you can easily work with LAS files, access point cloud data, and extract features such as elevation, intensity, and classification information from LIDAR data. This library is particularly useful for terrain modeling, forestry, urban planning, and other applications involving high-resolution 3D data.
41. (Point Data Abstraction Library): PDAL is a powerful open-source library for processing and analyzing point cloud data. PDAL supports numerous data formats and integrates seamlessly with other geospatial tools, including GDAL and LASzip. It is often used in geospatial pipelines to preprocess point cloud data before performing more advanced analysis using libraries like laspy or GeoMesa.
42. : PyVista is a Python library for 3D visualization and analysis of point clouds, meshes, and volumetric data. It provides a high-level interface for visualizing point cloud data in interactive 3D plots, generating surface meshes, and performing various geometric operations on point clouds. PyVista is actively maintained by Bane Sullivan and the OpenGeoVis team

43. : Open3D is an open-source library for 3D data processing. It includes modules for point cloud visualization, registration, filtering, and segmentation. Open3D supports various point cloud formats and provides a user-friendly interface for working with point cloud data.

44. : PCL is a C++ library for point cloud processing, but it also provides Python bindings. It offers a wide range of algorithms for point cloud filtering, feature extraction, registration, and segmentation. The Python bindings allow users to access PCL functionalities and perform advanced point cloud analysis tasks.

45. : PyntCloud is a Python library designed specifically for point cloud processing. It provides tools for reading, manipulating, and analyzing point clouds. PyntCloud supports multiple point cloud formats and offers functions for point cloud visualization, downsampling, and surface reconstruction.

46. : geemap is a Python library built on top of Google Earth Engine (GEE) and provides a user-friendly interface for interactive mapping and geospatial analysis. It allows users to visualize and analyze Earth observation data from GEE using interactive maps and widgets.

47. : reportlab is a Python library for creating PDF documents programmatically. In the context of geospatial data analysis, reportlab can be used to generate customized reports, maps, and visualizations from geospatial data analysis results. It allows you to create dynamic PDF documents with tables, charts, images, and maps, making it useful for generating high-quality reports and presentations based on geospatial analysis.

48. : h3-py is a Python binding for the H3 library, which is a geospatial indexing system based on hexagons. It provides a fast and efficient way to index and analyze geospatial data, especially for tasks involving spatial aggregation and hexagonal binning.

49. : Rtree is a library for performing spatial indexing. It allows for efficient spatial searches and queries, making it useful for tasks that involve spatial data organization and retrieval.

50. (Owned by: NASA/JPL-Caltech): PySAR is a Python library developed by NASA's Jet Propulsion Laboratory (JPL) and the California Institute of Technology (Caltech). It provides tools for processing, analyzing, and visualizing SAR data, including interferometric SAR (InSAR) and time series analysis. PySAR supports various SAR data formats and offers functionalities for deformation monitoring, topographic mapping, and other SAR analysis applications.

51. : SarPy is a Python library for reading, analyzing, and processing SAR data, including complex radar data. It provides tools for working with SAR data in various formats and performing SAR-specific tasks such as image formation, focusing, and interferometric processing. SarPy is developed and maintained by the United States Air Force Research Laboratory (AFRL).

52. : snappy is a Python library developed by the European Space Agency (ESA) for working with SAR data in the Sentinel-1 Toolbox (S1TBX). It provides a Python interface to the Sentinel Application Platform (SNAP) software, allowing users to access and process Sentinel-1 SAR data using Python scripts. snappy enables tasks such as data reading, preprocessing, and SAR analysis within the SNAP environment.

53. (Owned by: NASA/GSFC): PyRAT (Python Radiative Transfer) is a Python library developed by NASA's Goddard Space Flight Center (GSFC) for hyperspectral remote sensing analysis. It provides tools for atmospheric radiative transfer modeling, spectral unmixing, and retrieval of surface parameters from hyperspectral data. PyRAT is specifically designed for hyperspectral data analysis in Earth observation applications.

54. (Owned by: Python Spectral Development Team): Spectral is a Python library maintained by the Python Spectral Development Team. It focuses on hyperspectral remote sensing data analysis and provides functionalities for reading, processing, and visualizing hyperspectral data. Spectral offers numerous algorithms for spectral analysis, including classification, endmember extraction, and vegetation indices calculation. It can also be used for processing and analyzing thermal remote-sensing data. Spectral provides functions for loading, preprocessing, and analyzing thermal imagery. It includes algorithms for temperature retrieval and can handle thermal data in conjunction with hyperspectral data if required.

55. PySptools is a Python module that implements spectral and hyperspectral algorithms. The specializations of the library are endmember extraction, unmixing processes, supervised classification, target detection, noise reduction, convex hull removal, and feature extraction at the spectrum level. The library is designed to be easy to use and almost all functionality has a plot function to save you time with the data analysis process. The actual sources of the algorithms are the Matlab Hyperspectral Toolbox of Isaac Gerg, the pwctools of M. A. Little, the Endmember Induction Algorithms Toolbox (EIA), the HySime Matlab module of José Bioucas-Dias and José Nascimento and research papers.

56. : Hylite is a Python library for hyperspectral image analysis and visualization. It focuses on providing efficient algorithms for hyperspectral data processing, including spectral analysis, feature extraction, and classification. Hylite also offers interactive visualization capabilities for exploring hyperspectral datasets.

57. : Hyperspy is a Python library for multidimensional data analysis, including hyperspectral data. While not specifically focused on remote sensing, Hyperspy offers functionalities for handling and analyzing large and complex datasets commonly encountered in hyperspectral imaging. Hyperspy is an open-source project with contributions from various developers.

58. : Py6S is a Python interface for the 6S (Second Simulation of the Satellite Signal in the Solar Spectrum) radiative transfer model. It allows users to simulate and correct the atmospheric effects in thermal remote sensing data. By providing inputs such as sensor specifications, atmospheric conditions, and surface properties, Py6S can estimate the atmospheric correction and provide calibrated thermal imagery.

59. : ENVI Py is a Python interface to the ENVI (Environment for Visualizing Images) remote sensing software. ENVI is widely used in the remote sensing community, and ENVI Py allows users to access ENVI functionalities within Python. It provides tools for processing and analyzing thermal remote sensing data, including radiometric calibration, atmospheric correction, and temperature retrieval.
60. : PotreeConverter is a command-line tool (which can be used with Python) for converting large point cloud datasets into the Potree format, which is designed for efficient 3D web visualization. PotreeConverter is commonly used to create interactive web-based point cloud visualizations.
61. : pptk is a Python library for visualizing and interacting with large 3D point clouds in real time. It is particularly useful for the visual exploration and analysis of massive point cloud datasets.
62. : lasio is a Python library for reading and writing Log ASCII Standard (LAS) files, which are commonly used for storing well log data in the oil and gas industry. While not specifically for point cloud data, it can be useful when dealing with LAS-formatted point clouds.
63. : Pypotree is a Python library for reading, writing, and manipulating point cloud data in the Potree format. It allows users to convert Potree point clouds to LAS or PLY formats and perform various operations on Potree data.
64. : A Python library for accessing GPSD, a service daemon that retrieves and parses GPS data from various sources. It provides a simple interface to interact with GPS devices and access location, velocity, and time information.
65. : A library for parsing GPX (GPS Exchange Format) files. It allows easy reading, writing, and manipulation of GPS track, route, and waypoint data. The library provides a high-level API for working with GPX files and extracting information from them.
66. : A library for parsing NMEA (National Marine Electronics Association) data from GPS devices. It enables the extraction of various pieces of information, such as latitude, longitude, speed, and time, from NMEA sentences. pynmea2 simplifies the parsing process and provides a convenient interface for working with GPS data.
67. : geograpy3 is a library for extracting location names from the text, useful for geolocating GPS data based on textual information or reverse geocoding.
68. : A library for geodetic calculations and coordinate conversions, providing functions for distance, bearing, and destination calculations based on different models.
69. : A library for generating KML (Keyhole Markup Language) files, allowing the creation of KML documents for visualizing GPS and geospatial data in Google Earth or other KML-compatible software.

Note: The "owners" mentioned for some of the libraries refer to the organizations or developers who maintain and contribute to the libraries.

Follow me to stay updated on upcoming geospatial articles! 🔔

Thank you for reading this article. :)