GDAL:

There are five major modules that are included with the GDAL Python bindings.:

```
>>> from osgeo import gdal

>>> from osgeo import ogr

>>> from osgeo import osr

>>> from osgeo import gdal_array

>>> from osgeo import gdalconst
```

Essential geospatial Python libraries:

Shapely and Geopandas:

When dealing with geometry data, functionality of the combined use of shapely and geopandas.

With **shapely**, you can create shapely geometry objects (e.g. Point, Polygon, Multipolygon) and manipulate them, e.g. buffer, calculate the area or an intersection etc.

Shapely itself does not provide options to read/write vector file formats (e.g. shapefiles or geojson) or handle projection conversions. This can be handled e.g. with the Fiona library. But there is an even more convenient way:

Geopandas combines the geometry objects of shapely, the read/write/ projection functions of fiona and the powerful dataframe interface of the pandas library in one awesome package. In the spreadsheet-like dataframe, the last column 'geometry' stores the shapely geometry objects, all shapely functions can be applied. The pandas mechanics offers super easy ways to manipulate, plot and analyze the data, e.g. dataframe groupby operations etc.

Rasterio

Rasterio is the go-to library for raster data handling. It lets you read/write raster files to/from numpy arrays (the de-facto standard for Python array operations), offers many convenient ways to manipulate these array (e.g. masking, vectorizing etc.) and can handle transformations of coordinate reference systems. Just like any other numpy array, the data can also be easily plotted, e.g. using the matplotlib library.

rasterstats: For zonal statistics. Extracts statistics from rasters files or numpy arrays based on geometries.

scikit-image: Library for image manipulation, e.g. histogram adjustments, filter, segmentation/edge detection operations, texture feature extraction etc.

scikit-learn: The best and at the same time easy-to-use Python machine learning library. Regression, classification, dimensionality reductions etc.

folium: Lets you visualize spatial data on interactive leaflet maps.

descartes: Enables plotting of shapely geometries as matplotlib paths/ patches. Also a dependency for the geometry plotting functions of geopandas.

pyproj: For transformation of projections. Mostly unnecessary when using the more conveniant geopandas coordinate reference system (crs) functions.

PySAL: The Python Spatial Analysis Library contains a multitude of functions for spatial analysis, statistical modeling and plotting.

xarray: Great for handling extensive image time series stacks, imagine 5 vegetation indices x 24 dates x 256 pixel x 256 pixel. xarray lets you label the dimensions of the multidimensional numpy array and combines this with many functions and the syntax of the pandas library (e.g. groupby, rolling window, plotting). Not essential for beginners, but it is a great addition when working with extensive time series data.

References:

https://chrieke.medium.com/essential-geospatial-python-libraries-5d82fcc38731

https://gdal.org/