ArcMap Computatioins in Python

Kommineni Tirupathi Rayudu
MT21170
kommineni21170@iiitd.ac.in
Arnab Chhatait
MT21165
arnab21165@iiitd.ac.in

December 23, 2022

1 ArcPy

ArcPy is a Python site package that provides a useful and productive way to perform geographic data analysis, data conversion, data management, and map automation with Python. Python is a free, cross-platform, open source programming language. It's widely used and supported and is an integral part of ArcGIS. ArcPy and ArcGIS API for Python are complementary libraries; ArcPy allows you to use, automate, and extend desktop GIS, and ArcGIS API for Python supports the same for Web GIS.

Leverage specialized machine learning techniques that are inherently spatial via ArcPy—such as Spatially Constrained Multivariate Clustering, Forest-Based Classification, and Empirical Bayesian Kriging—to ensure the statistical significance of your models.

An example is using ArcPy to manage local data, adding it as layers to a map, and using geoprocessing tools to create outputs and service definition files. ArcGIS API for Python can then be used to publish the definition files to the Web GIS, compose a web map, or share those layers with others.

Python can be used to automate the execution of geoprocessing tools as well as provide the ability to create your own geoprocessing tools, either as a script tool or as a Python toolbox tool.

ArcGIS applications and scripts written using ArcPy benefit from being able to access and work with the numerous Python modules developed by GIS professionals and programmers from many different disciplines. The additional power of using ArcPy within Python is the fact that Python is a general-purpose programming language that is easy to learn and use. It is interpreted and dynamically typed, which provides you with the ability to quickly prototype and test scripts in an interactive environment while still being powerful enough to support the writing of large applications.

A function is a defined bit of functionality that does a specific task and can be incorporated into a larger program. In addition to tools, ArcPy exposes a number of functions to better support geoprocessing workflows. Functions can be used to list certain datasets, retrieve a dataset's properties, check for existence of data, validate a table name before adding it to a geodatabase, or perform many other useful scripting tasks.

2 Displaying Shapefiles in Python

- arcpy.MakeFeatureLayer_management('path_name/shape_file_name.shp','states') : This command is used to take the shapefile as input and project the file as a layer with the given name.

- arcpy.MakeFeatureLayer_management('path_name/shape_file_name.shp','districts') : This command is used to take the shapefile as input and project the file as a layer with the given name.

- arcpy.management.SelectLayerByAttribute("districts", "ADD_TO_SELECTION", "NAME_1='name_of_district'") : This command is used to select a particular district from the shapefile having column name district.

- arcpy.SelectLayerByAttribute_management("districts", "ADD_TO_SELECTION", "NAME_1='name_of_state'") : This command is used to select a particular district and add it to the existing selection formed.

- arcpy.CopyFeatures_management

  ("segregated_name","target_shapefile_name.shp") : This command is used to segregate out the new disticts selected though the previous commands to a new shapefile with the path where the file is to be saved.

- mxd.save() : This command is used to save the
- outCS = arcpy.SpatialReference('WGS 1984 UTM Zone 47N') The spatial reference properties available depend on the coordinate system used.
- arcpy.CopyFeatures_management("existing_layer","creating a new shapefile from the features")

3 ArcMap Computations in Python

3.1 Calculation of Field Area

arcpy.management.AddField('path_of_in_table', 'field_name', 'field_type', field_precision, field_scale, field_length, field_alias, field_is_nullable, field_is_required, field_domain)

- field_type : Type of output after computation - float, double, short, long, raster
- field_precision : The number of digits that can be stored in the field. All digits are counted regardless of which side of the decimal they are on.
- field_scale : The number of decimal places stored in a field.
- field_length : The length of the field. This sets the maximum number of allowable characters for each record of the field.
- field_is_nullable : Whether the field can be null or not null.
- field_domain : Constrains the values allowed in any particular attribute for a table, feature class, or subtype in a geodatabase. You must specify the name of an existing domain for it to be applied to the field.

One instance of use of the above command :
arcpy.AddField_management('path_name_file_name.shp', "Area", "Double", "20", "5", "", "", "NON_NULLABLE", "NON_REQUIRED", "")

3.2 Intersection Computation

In GIS, an Intersect is an analytical operation that can be used to select any part of a feature that intersects with one or more other features. The areas of the map where all the input features intersect will create a feature as the intersect output.

arcpy.analysis.Intersect(in_features, out_feature_class, join_attributes, cluster_tolerance, output_type)

- in_features : A list of the input feature classes or layers. When the distance between features is less than the cluster tolerance, the features with the lower rank will snap to the feature with the higher rank. The highest rank is 1.

- join_attributes : Specifies the attributes from the input features that will be transferred to the output feature class.

    - ALL — All the attributes from the input features will be transferred to the output feature class. This is the default.
    - NO_FID — All the attributes except the FID from the input features will be transferred to the output feature class.
    - ONLY_FID — Only the FID field from the input features will be transferred to the output feature class.

- cluster$_tolerance : Theminimumdistanceseparatingallfeaturecoordinates(nodesandverti$
  $Specifiesthetypeofintersectionstobereturned.$

- - INPUT—The intersections returned will be the same geometry type as the input features with the lowest dimension geometry. If all inputs are polygons, the output feature class will contain polygons. If one or more of the inputs are lines and none of the inputs are points, the output will be line. If one or more of the inputs are points, the output feature class will contain points. This is the default.
  - LINE—The intersections returned will be line. This is only valid if none of the inputs are points.
  - POINT—The intersections returned will be point. If the inputs are line or polygon, the output will be a multipoint feature class.

One instance of use of the above command :
arcpy.Intersect_analysis(r'shape_file_path_of_road' ;'shape_file_path_of_district' ',r'path_of_output_shape_file', 'ALL', '', 'INPUT')

3.3 Statistical Analysis of ArcMap in Python

arcpy.analysis.Statistics(in_table, out_table, statistics_fields, case_field)

- statistics_fields [[field, {statistic_type}] ,...]

One instance of use of the above command :
arcpy.Statistics_analysis("ap_road_ins_u_utm",out_file,[["Road_len","SUM"]] ,"ID_2")
The result will be stored in the form of a text file in this location : 'path_of_file_road_file.txt'

4 Adding New Field to the Shapefile in ArcPy

Adds a new field to a table or the table of a feature class or feature layer, as well as to rasters with attribute tables.

Using the Add Field Commands in ArcPy:

- For shapefiles and dBase tables, if field type defines a character, blanks are inserted for each record. If field type defines a numeric item, zeros are inserted for each record.
- The Field Length parameter is only applicable to fields of type text. If a length is not specified, the default length of 255 will be used.
- For geodatabases, if field type defines a character or number, ¡null¿ is inserted into each record if the Field supports null values parameter is checked.
- A nonnullable field cannot be added to a nonempty geodatabase feature class or table.

- A shapefile does not support aliases for fields, so you cannot add a field alias to a shapefile.

- The Field Domain parameter can use an existing domain from a feature class in a geodatabase. The name of an existing domain must be provided. Providing an invalid domain name or value will not cause the tool to fail, but the invalid name or value will be ignored and no domain will be set for the field.

- The precision and scale of a field describe the maximum precision and size of data that can be stored in the field. The precision describes the number of digits that can be stored in the field, and the scale describes the number of decimal places for float and double fields. For example, for a value of 54.234, the scale is 3 and the precision is 5.

arcpy.management.AddField(in_table, field_name, field_type, {field_precision}, {field_scale}, {field_length}, {field_alias}, {field_is_nullable}, {field_is_required}, {field_domain})

- field_type
  - TEXT The field type will be text. Text fields support a string of characters.
  - FLOAT The field type will be float. Float fields support fractional numbers between -3.4E38 and 1.2E38.
  - DOUBLE The field type will be double. Double fields support fractional numbers between -2.2E308 and 1.8E308.
  - SHORT The field type will be short. Short fields support whole numbers between -32,768 and 32,767.
  - LONG The field type will be long. Long fields support whole numbers between -2,147,483,648 and 2,147,483,647.
  - DATE The field type will be date. Date fields support date and time values.
  - BLOB The field type will be BLOB. BLOB fields support data stored as a long sequence of binary numbers. You need a custom loader or viewer or a third-party application to load items into a BLOB field or view the contents of a BLOB field.

  - RASTER The field type will be raster. Raster fields can store raster data in or alongside the geodatabase. All ArcGIS software-supported raster dataset formats can be stored, but it is recommended that only small images be used.
  - GUID The field type will be GUID. GUID fields store registry-style strings consisting of 36 characters enclosed in curly brackets.

  Specifies the field type of the new field. Although the Field object's type property values are not an exact match for the keywords used by the Add Field tool's field_type parameter, all of the Field object's type values can be used as input to this parameter. The different field types are mapped as follows:

  - Integer to LONG
  - String to TEXT
  - Small Integer to SHORT

- field_domain : Constrains the values allowed in any particular attribute for a table, feature class, or subtype in a geodatabase. You must specify the name of an existing domain for it to be applied to the field.

arcpy.management.CalculateField(in_table, field, expression, expression_type, code_block, field_type, enforce_domains)

- expression_type : Specifies the type of expression that will be used.

  - PYTHON3 : The Python expression type will be used. This is the default.
  - ARCADE : The Arcade expression type will be used.
  - SQL : The SQL expression type will be used. SQL expressions support faster calculations for feature services and enterprise geodatabases. Instead of performing calculations one feature or row at a time, a single request is sent to the server or database, resulting in significantly faster calculations. Only feature services and enterprise geodatabases support SQL expressions.

- field_type Specifies the field type of the new field. This parameter is only used when the field name does not exist in the input table. If the field is of type text, the new field will have a length of 512. For shapefiles and database files, the field will have a length of 254.

  - TEXT : The field type will be text. Text fields support a string of characters.
  - FLOAT : The field type will be float. Float fields support fractional numbers between -3.4E38 and 1.2E38.
  - DOUBLE : The field type will be double. Double fields support fractional numbers between -2.2E308 and 1.8E308.
  - SHORT : The field type will be short. Short fields support whole numbers between -32,768 and 32,767.
  - LONG : The field type will be long. Long fields support whole numbers between -2,147,483,648 and 2,147,483,647.
  - DATE : The field type will be date. Date fields support date and time values.
  - BLOB : The field type will be BLOB. BLOB fields support data stored as a long sequence of binary numbers. You need a custom loader or viewer or a third-party application to load items into a BLOB field or view the contents of a BLOB field.
  - RASTER : The field type will be raster. Raster fields can store raster data in or alongside the geodatabase. All ArcGIS software-supported raster dataset formats can be stored, but it is recommended that only small images be used.
  - GUID : The field type will be GUID. GUID fields store registry-style strings consisting of 36 characters enclosed in curly brackets. String

- enforce_domains : Specifies whether field domain rules will be enforced.
  - ENFORCE_DOMAINS : Field domain rules will be enforced.
  - NO_ENFORCE_DOMAINS : Field domain rules will not be enforced. This is the default.

The Output will be the updated table along with the newly caculated column.

One instance of use of the above command.

arcpy.CalculateField_management(”dist_ap_utm”,”dist_ap_utm.Road_den”,’!SUM_Road_l

4.1 ArcPy Joins in Python

arcpy.management.AddJoin(in_layer_or_view, in_field, join_table, join_field, join_type, index_join_fields)

- in_layer_or_view : The layer or table view to which the join table will be joined.
- in_field : The field in the input layer or table view on which the join will be based.
- join_table : The table or table view to be joined to the input layer or table view.
- join_field : The field in the join table that contains the values on which the join will be based.
- join_type : Specifies whether only records in the input that match a record in the join table will be included in the output.
    - KEEP_ALL : All records in the input layer or table view will be included in the output. This is also known as an outer join. This is the default.
    - KEEP_COMMON : Only those records in the input that match a row in the join table will be included in the output. This is also known as an inner join.
- index_join_fields : Specifies whether table attribute indexes will be added to both joining fields.
    - INDEX_JOIN_FIELDS : Both join fields will be indexed. If the table already has an index, a new index will not be added.
    - NO_INDEX_JOIN_FIELDS : Indexes will not be added. This is the default.

We want to join two file: one is a shapefile and the other is a CSV file, so that we can get the road length over the file we joined.