



Python Geospatial Analysis with Qlik Sense AAI

REQUIREMENTS

- **Mandatory prerequisite: *'Python with Qlik Sense AAI – Environment Setup'***
- Qlik Sense June 2017+
- Python 3.5.3 64 bit
- Python Libraries: grpcio, scipy


LAYOUT

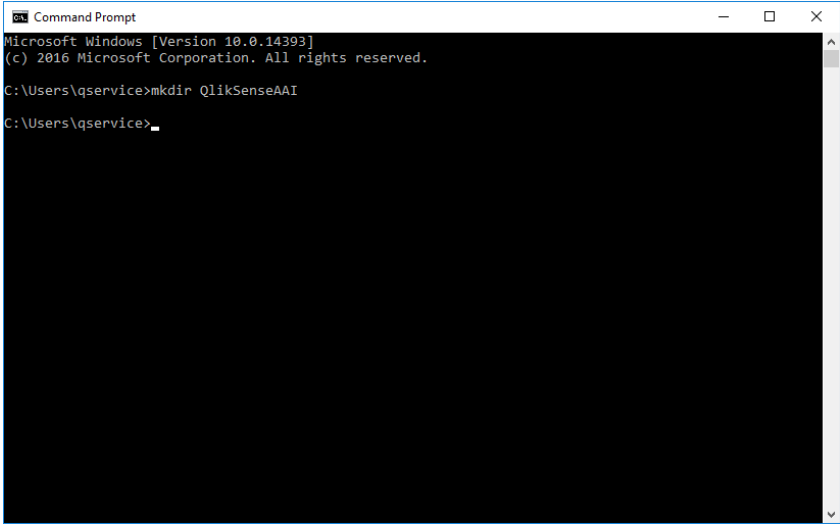
- [Prepare your Project Directory](#)
- [Install Python Libraries and Required Software](#)
- [Setup an AAI Connection in the QMC](#)
- [Copy the Package Contents and Import Examples](#)
- [Prepare/Start Services](#)
- [Leverage Python Sentiment Analysis from within Qlik Sense](#)

PREPARE YOUR PROJECT DIRECTORY

ALERT

Virtual environments are not necessary, but are frequently considered a best practice when handling multiple Python projects.

1. Open a command prompt ( +'cmd')
2. Let's now make a new project folder called QlikSenseAAI, where all of our projects will live that leverage the QlikSenseAAI virtual environment that we've created. Let's place it under 'C:\Users\{Your Username}'. If you have already created this folder in another guide, simply skip this step.

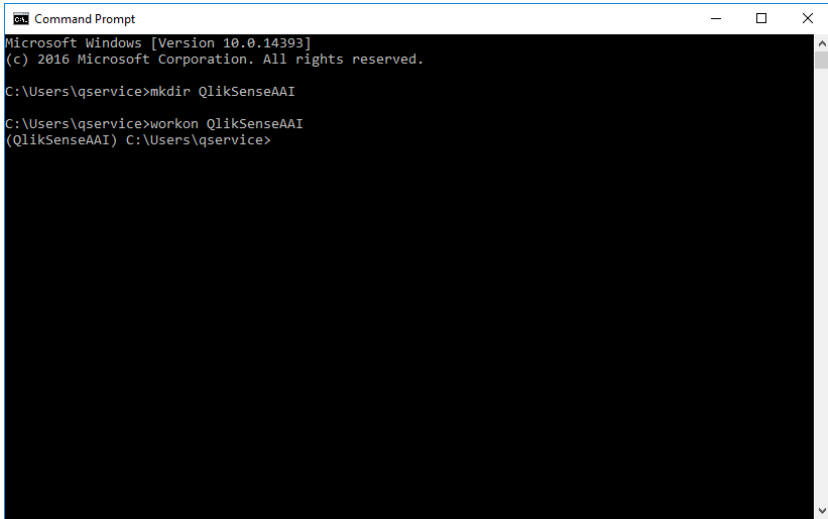


```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\qservice>mkdir QlikSenseAAI

C:\Users\qservice>
```

3. We now want to leverage our virtual environment. If you are not already in your environment, enter it by executing:
 - o workon QlikSenseAAI

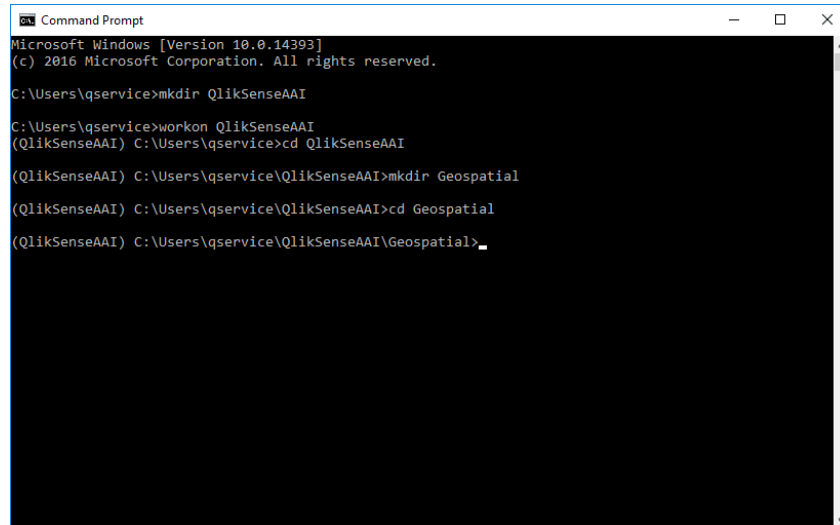


```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\qservice>mkdir QlikSenseAAI

C:\Users\qservice>workon QlikSenseAAI
(QlikSenseAAI) C:\Users\qservice>
```

4. Now, ensuring you are in the 'QlikSenseAAI' folder that you created (if you have followed another guide, it might redirect you to a prior working directory, as seen below), execute the following commands to create and navigate into your project's folder structure:
- cd QlikSenseAAI
 - mkdir Geospatial
 - cd Geospatial



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\qservice>mkdir QlikSenseAAI

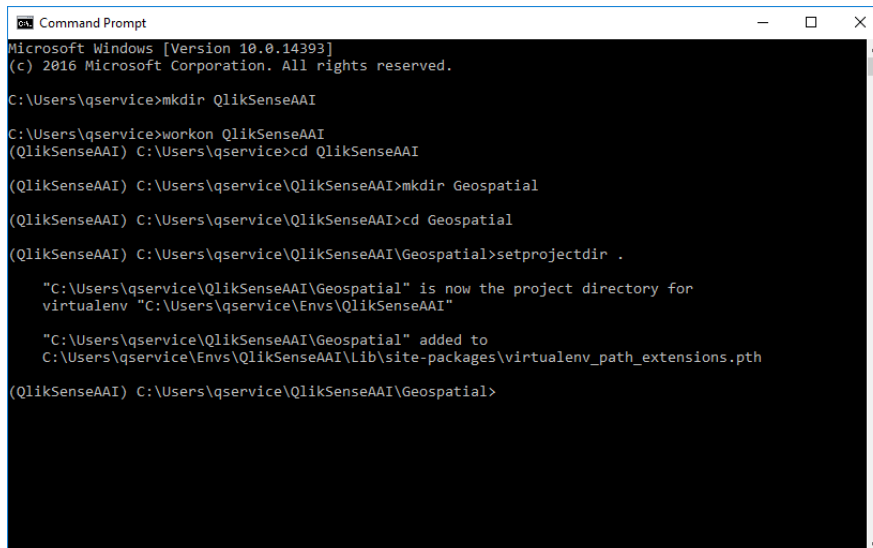
C:\Users\qservice>workon QlikSenseAAI
(QlikSenseAAI) C:\Users\qservice>cd QlikSenseAAI

(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI>mkdir Geospatial

(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI>cd Geospatial

(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI\Geospatial>
```

5. Lets now bind our current working directory as the virtual environment's default. Execute (Note the period!):
- setprojectdir .



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\qservice>mkdir QlikSenseAAI

C:\Users\qservice>workon QlikSenseAAI
(QlikSenseAAI) C:\Users\qservice>cd QlikSenseAAI

(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI>mkdir Geospatial

(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI>cd Geospatial

(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI\Geospatial>setprojectdir .

"C:\Users\qservice\QlikSenseAAI\Geospatial" is now the project directory for
virtualenv "C:\Users\qservice\Envs\QlikSenseAAI"


"C:\Users\qservice\QlikSenseAAI\Geospatial" added to
C:\Users\qservice\Envs\QlikSenseAAI\Lib\site-packages\virtualenv_path_extensions.pth

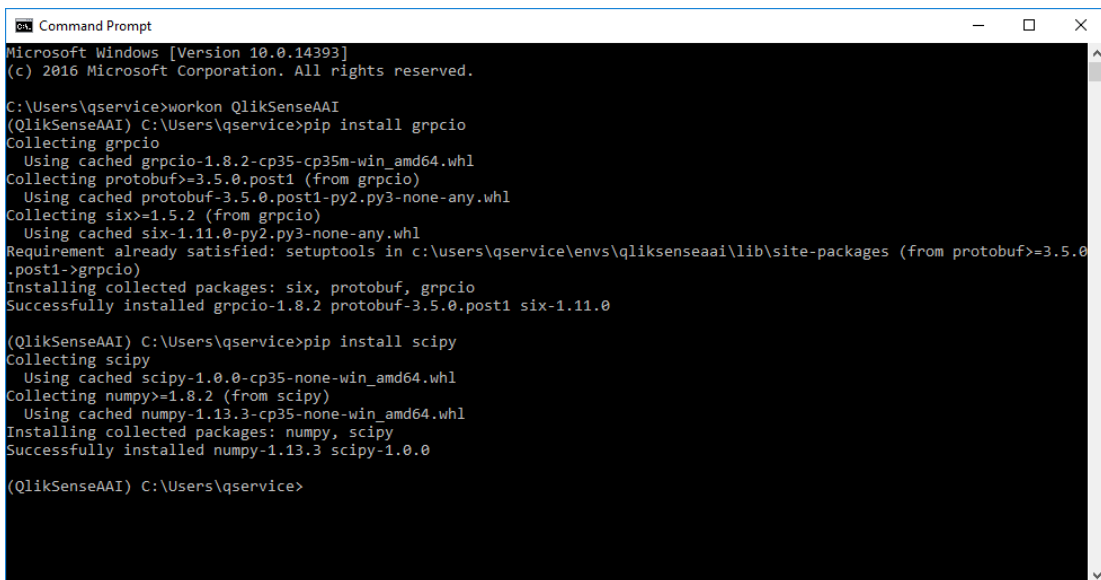
(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI\Geospatial>
```

6. We have now set the stage for our Geospatial Python environment. To navigate back into this project in the future, simply execute:
- workon QlikSenseAAI

This will take you back into the environment with the default directory that we set above. To change the directory for future projects within the same environment, change your directory to the desired path and reset the working directory with 'setprojectdir .'

INSTALL PYTHON LIBRARIES AND REQUIRED SOFTWARE

1. Open a command prompt ( + 'cmd') or continue in your current command prompt, ensuring that you are currently within the virtual environment—you will see (QlikSenseAAI) preceding the directory if so. If you are not, execute:
 - a. workon QlikSenseAAI
2. Execute the following commands (in my case, both requirements have already been met. If you have followed a previous guide, you have more than likely already installed grpcio):
 - a. pip install grpcio
 - b. pip install scipy
 - i. Note you can always download and install the wheel from <https://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy> and searching for ['scipy-1.0.0-cp35-cp35m-win_amd64.whl'](#)
 - ii. then you can execute:
 1. pip install c:\path\to\directory\scipy-1.0.0-cp35-cp35m-win_amd64.whl



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

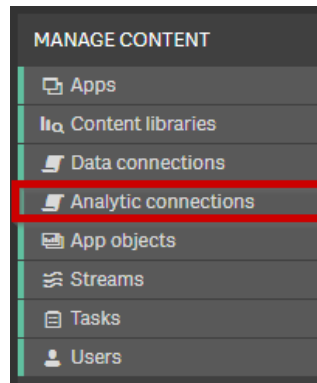
C:\Users\qservice>workon QlikSenseAAI
(QlikSenseAAI) C:\Users\qservice>pip install grpcio
Collecting grpcio
  Using cached grpcio-1.8.2-cp35-cp35m-win_amd64.whl
Collecting protobuf>=3.5.0.post1 (from grpcio)
  Using cached protobuf-3.5.0.post1-py2.py3-none-any.whl
Collecting six>=1.5.2 (from grpcio)
  Using cached six-1.11.0-py2.py3-none-any.whl
Requirement already satisfied: setuptools in c:\users\qservice\envs\qliksenseaa1\lib\site-packages (from protobuf>=3.5.0.post1->grpcio)
Installing collected packages: six, protobuf, grpcio
Successfully installed grpcio-1.8.2 protobuf-3.5.0.post1 six-1.11.0

(QlikSenseAAI) C:\Users\qservice>pip install scipy
Collecting scipy
  Using cached scipy-1.0.0-cp35-none-win_amd64.whl
Collecting numpy>=1.8.2 (from scipy)
  Using cached numpy-1.13.3-cp35-none-win_amd64.whl
Installing collected packages: numpy, scipy
Successfully installed numpy-1.13.3 scipy-1.0.0

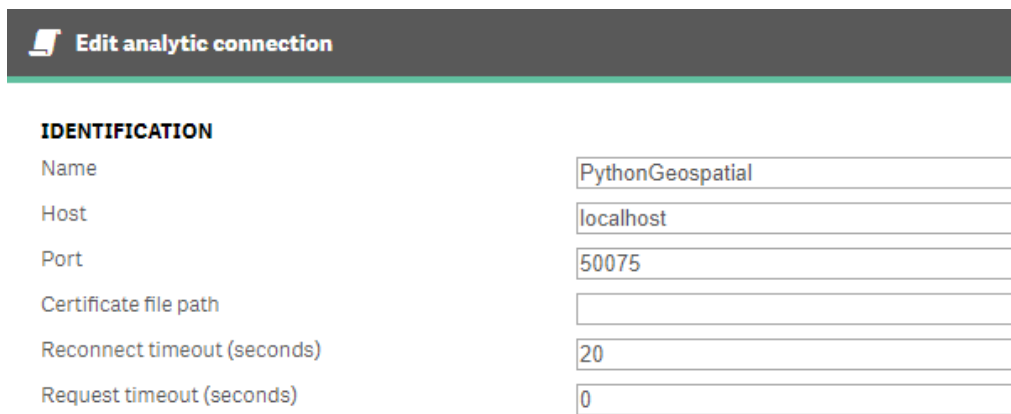
(QlikSenseAAI) C:\Users\qservice>
```

SET UP AN AAI CONNECTION IN THE QMC

1. Navigate to the QMC and select 'Analytic connections'



2. Fill in the parameters as follows – note that the 'Name' field will be how you will call your desired connection within the application editors in Sense (e.g. PythonGeospatial.ConvexHull()). The default port is 50051, but each analytics connection needs to have its own port. This module's port is set to 50075, but if you'd like to change that, simply search for '-port' in the 'extensionService_geospatial.py' file and change the 'default' parameter to an available port.

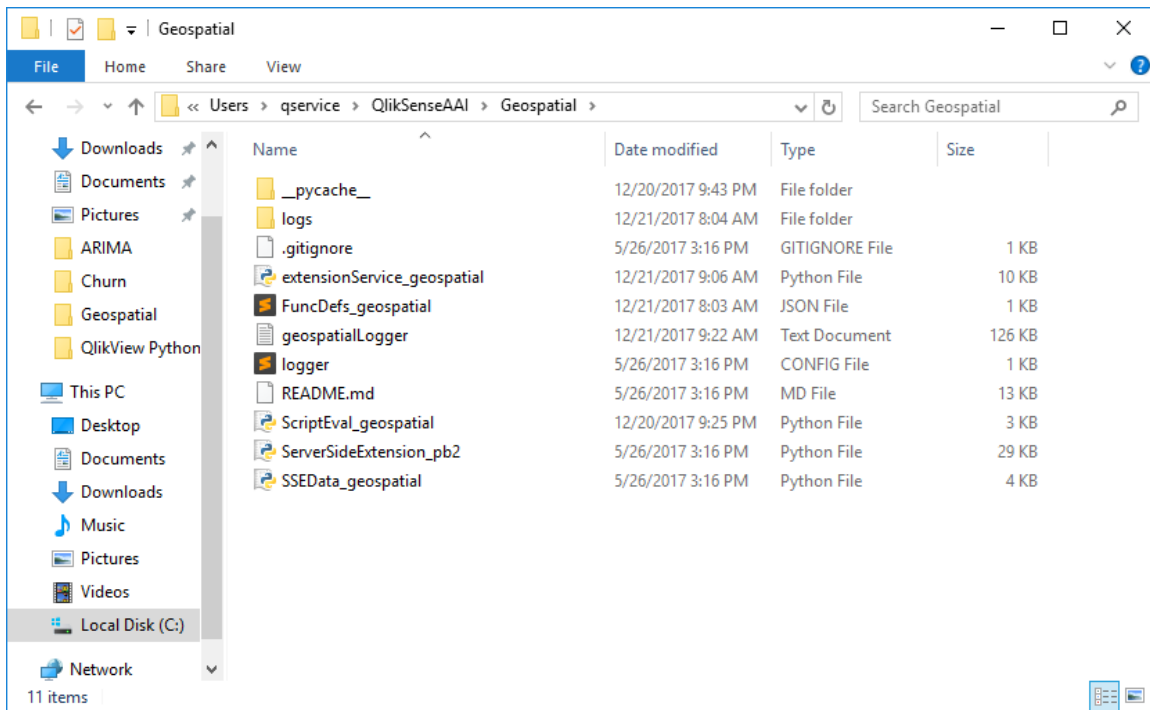
A screenshot of the 'Edit analytic connection' form. The form has a title bar 'Edit analytic connection' and a section titled 'IDENTIFICATION'. The fields and their values are: Name (PythonGeospatial), Host (localhost), Port (50075), Certificate file path (empty), Reconnect timeout (seconds) (20), and Request timeout (seconds) (0).

IDENTIFICATION	
Name	PythonGeospatial
Host	localhost
Port	50075
Certificate file path	
Reconnect timeout (seconds)	20
Request timeout (seconds)	0

3. Click 'Apply', and you've now created a new analytics connection.

COPY THE PACKAGE CONTENTS AND IMPORT EXAMPLES

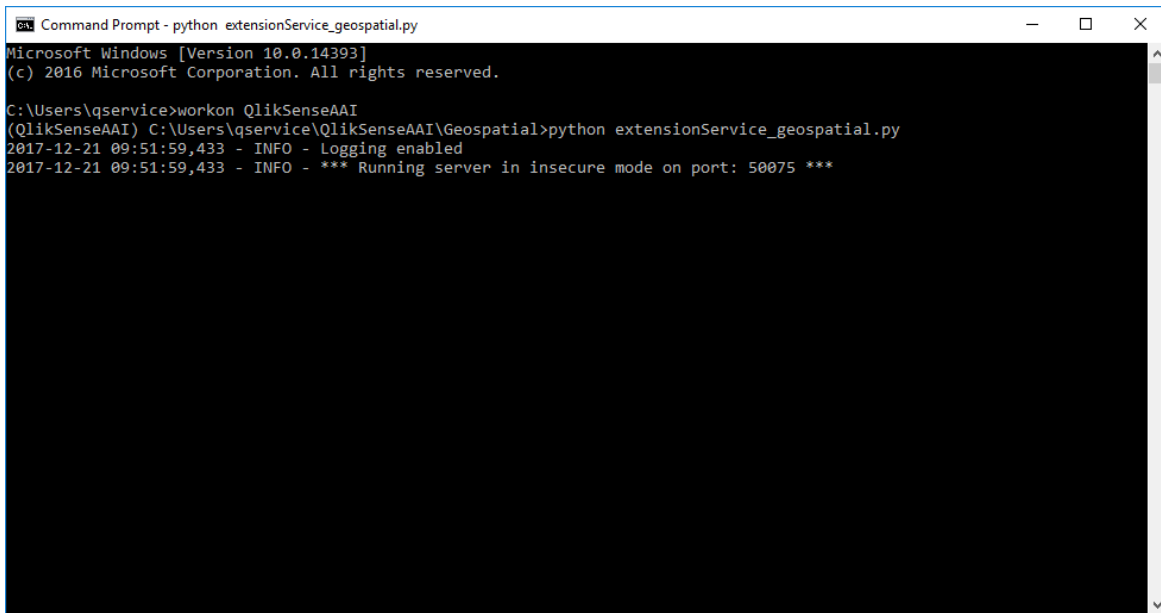
1. Now we want to setup our example geocoding service and app. Let's start by copying over the contents of the geocoding example from this packages zip file to the '..\QlikSenseAAI\Geospatial\' location, as so:



2. After copying over the contents, go ahead and import the example qvf using the QMC.

PREPARE/START SERVICES

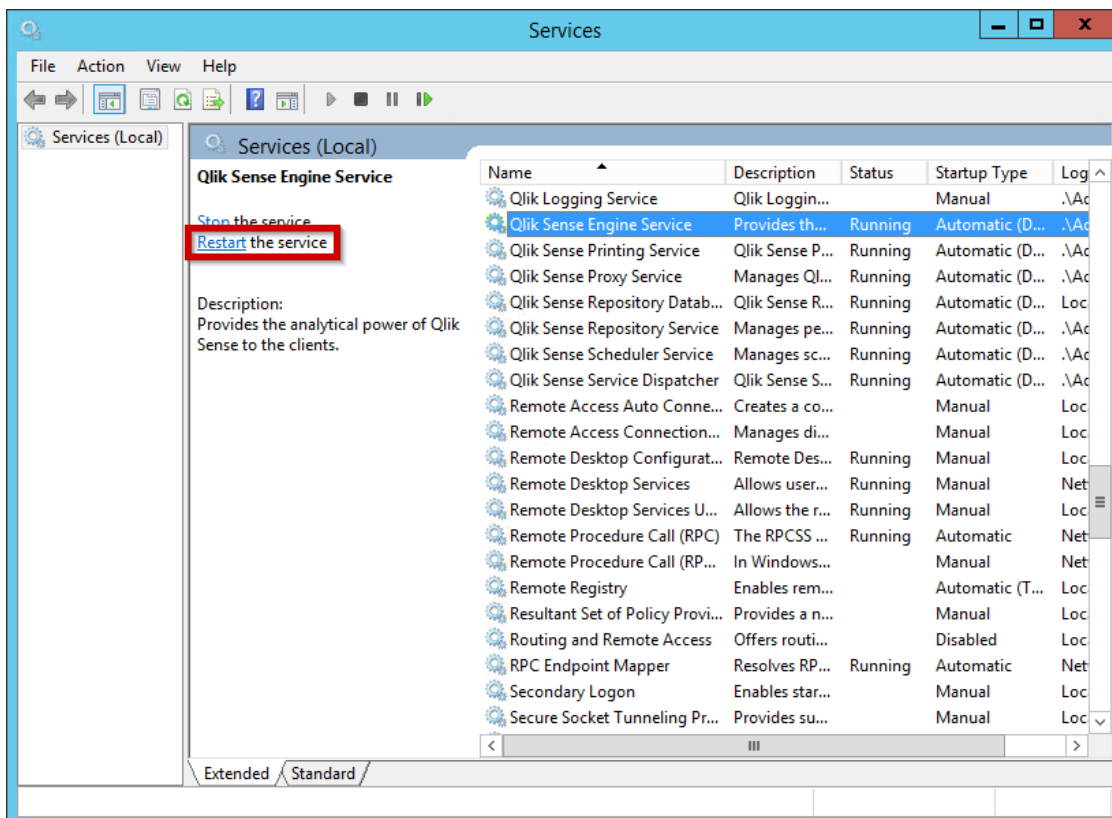
1. At this point the setup is complete, and we now need to start the extension service. To do so, navigate back to the command prompt. Please make sure that you are inside of the virtual environment. (*note that the command is 'workon QlikSenseAAI')
2. Once at the command prompt and within your environment, execute:
 - `python extensionService_geospatial.py`



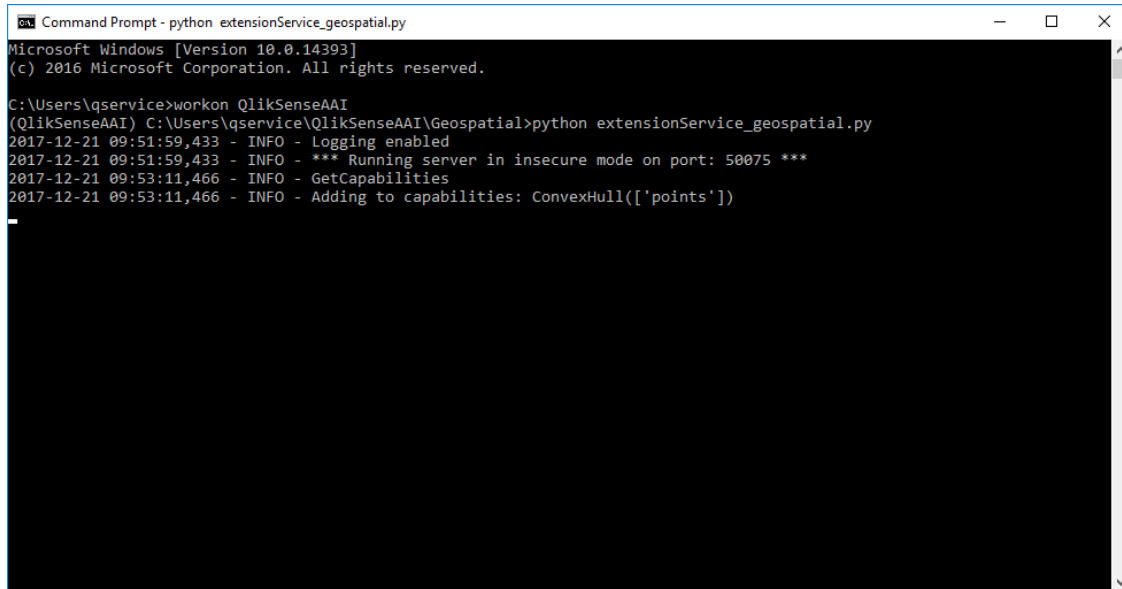
```
Command Prompt - python extensionService_geospatial.py
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\qservice>workon QlikSenseAAI
(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI\Geospatial>python extensionService_geospatial.py
2017-12-21 09:51:59,433 - INFO - Logging enabled
2017-12-21 09:51:59,433 - INFO - *** Running server in insecure mode on port: 50075 ***
```

3. We now need to restart the Qlik Sense engine service so that it can register the new AAI service. To do so, navigate to windows Services and restart the 'Qlik Sense Engine Service'



4. You should now see in the command prompt that the Qlik Sense Engine has registered the function ConvexHull() from the extension service over port 50075, or whichever port you've chosen to leverage.



```
Command Prompt - python extensionService_geospatial.py
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\qservice>workon QlikSenseAAI
(QlikSenseAAI) C:\Users\qservice\QlikSenseAAI\Geospatial>python extensionService_geospatial.py
2017-12-21 09:51:59,433 - INFO - Logging enabled
2017-12-21 09:51:59,433 - INFO - *** Running server in insecure mode on port: 50075 ***
2017-12-21 09:53:11,466 - INFO - GetCapabilities
2017-12-21 09:53:11,466 - INFO - Adding to capabilities: ConvexHull(['points'])
```


LEVERAGE PYTHON GEOSPATIAL ANALYSIS FROM WITHIN SENSE

1. This Python Geospatial package at this point in time contains a single function called `ConvexHull()`. This algorithm calculates the containing boundary points of a series of points. This expression is scalar, and can be used on the backend to pre-aggregate these boundaries, or on the front end to create them dynamically. The expression is setup to accept a single parameter:
 - Points
 - i. A list of coordinates in the format of `'[1,2],[3,4],[5,6],...'`
2. The function will return a list of points in the format of: `'[[1,2],[3,4],[5,6],...]'`
 - Note, to draw in GeoAnalytics, you will need to wrap the returned object with an additional `'[]'`
3. Example function call from the front end:
 - `PythonGeospatial.ConvexHull(Concat(DISTINCT GeoPoint,','))` – where `GeoPoint` is a Qlik `GeoMakePoint()` object
4. Example function call from the load script:

```
PreAggregatedRegions:
```

```
LOAD
```

```
    '[' & PythonGeospatial.ConvexHull(Concat(GeoPoint,',')) & ']' as ConvexHullRegions,
```

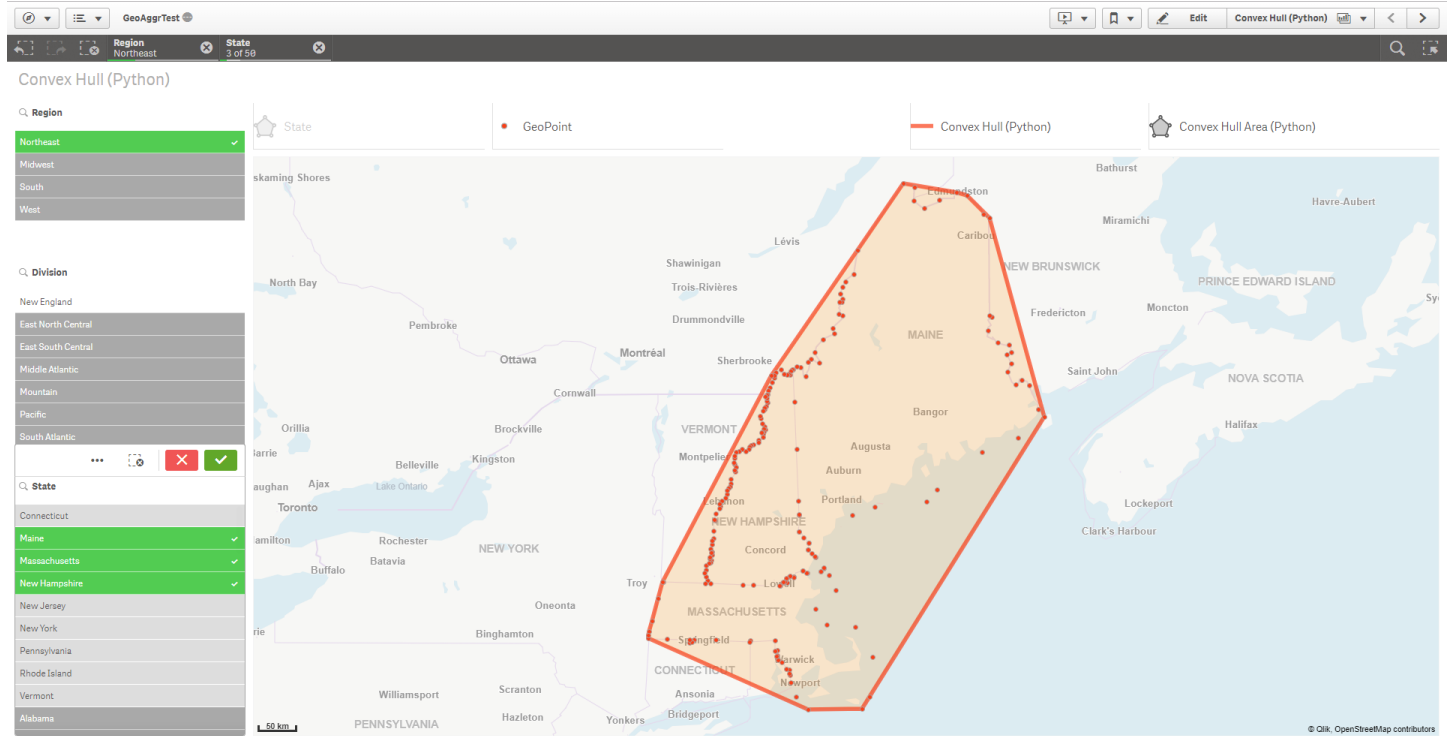
```
    Region
```

```
RESIDENT Master
```

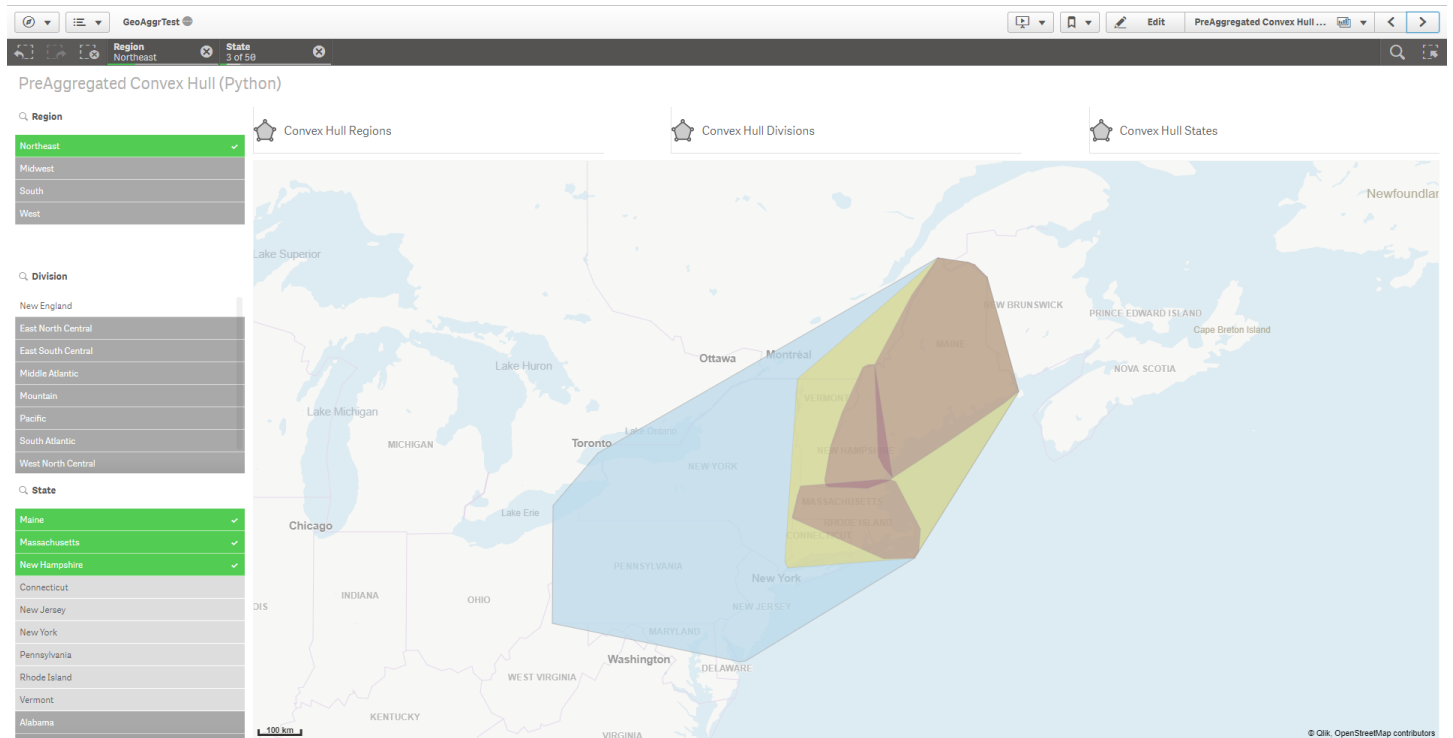
```
GROUP BY Region;
```

The attached demo app contains lat/long coordinates of the boundaries of all of the US States. The ConvexHull() function takes the outermost boundaries of all available points.

Dynamic Convex Hull:



Pre-aggregated Convex Hull by Region, Division, State:



Pre-aggregated Convex Hull Drilldowns:

