

Python and SAS® Viya™ Integration

Installation Steps for the Client Modules

January, 2017

Python-SWAT (Scripting Wrapper for Analytics Transfer) package, is a Python interface to SAS Cloud Analytic Services (CAS) which is required to make a connection to a SAS Viya Platform. It allows users to execute CAS actions and process the results all from Python.

<https://github.com/sassoftware/python-swat>

Requirements

- A client machine to install Python with the required packages
- SAS Viya connection details for saasnow server
 - a. Hostname (e.g. *vdmml-sasmea.saasnow.com*)
 - b. Username/Password (e.g. *foo/bar*)
 - c. Port Number (e.g. *8443*)

A) Client Installation on LINUX - Connecting to SAS Viya using Python Open API

1. Download and install Python-Anaconda on the client machine: <https://docs.continuum.io/anaconda/install>
64-bit Python version of 2.7, 3.4, or 3.5 is required.
Anaconda also installs Jupyter notebook and the most popular Python packages.
2. Install SWAT-Python package from GitHub: <https://github.com/sassoftware/python-swat>
>> pip install <https://github.com/sassoftware/python-swat/releases/download/v1.0.0/python-swat-1.0.0-linux64.tar.gz>
3. Test connection from client machine to SAS Viya server
>> ping <replace-this-with-your-valid-SASViya-hostname>
e.g. ping vdmml-sasmea.saasnow.com
4. Start up Jupyter notebook on the client machine
>> jupyter notebook -ip='*' -port=8888 -no-browser
The option "ip" allows connection from other machines. If you would like to allow the connection only from one machine then replace '*' with the IP address.
e.g. jupyter notebook -ip='192.168.0.7' -port=8888 -no-browser
If you are running Jupyter on your localhost, you can ignore the "ip" option.
e.g. jupyter notebook -port=8888 -no-browser

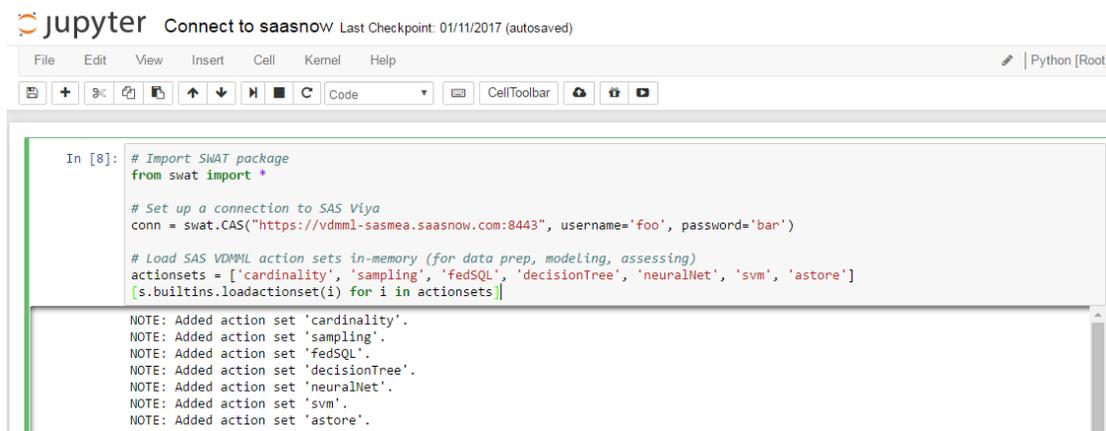
5. Connect to Jupyter notebook on Linux client from a browser
<http://<replace-this-with-your-valid-client-hostname>:8888/tree>
e.g. <http://192.168.0.7:8888/tree>

6. Open a new Python notebook on Jupyter and type the following code:

```
# Import SWAT package
from swat import *

# Set up a connection to SAS Viya
conn = swat.CAS("https://<replace-this-with-your-valid-SASViya-hostname>", username='<replace-this-with-your-username>', password='<replace-this-with-your-password>')
# conn = swat.CAS("https://vdmml-sasmea.saasnow.com:8443", username='foo', password='bar')

# Load SAS VDMML action sets in-memory (for data prep, modeling, assessing)
actionsets = ['cardinality', 'sampling', 'fedSQL', 'decisionTree', 'neuralNet', 'svm', 'astore']
[s.builtins.loadactionset(i) for i in actionsets]
```



```
jupyter Connect to saasnow Last Checkpoint: 01/11/2017 (autosaved)
File Edit View Insert Cell Kernel Help Python [Root]
In [8]: # Import SWAT package
from swat import *

# Set up a connection to SAS Viya
conn = swat.CAS("https://vdmml-sasmea.saasnow.com:8443", username='foo', password='bar')

# Load SAS VDMML action sets in-memory (for data prep, modeling, assessing)
actionsets = ['cardinality', 'sampling', 'fedSQL', 'decisionTree', 'neuralNet', 'svm', 'astore']
[s.builtins.loadactionset(i) for i in actionsets]

NOTE: Added action set 'cardinality'.
NOTE: Added action set 'sampling'.
NOTE: Added action set 'fedSQL'.
NOTE: Added action set 'decisionTree'.
NOTE: Added action set 'neuralNet'.
NOTE: Added action set 'svm'.
NOTE: Added action set 'astore'.
```

You are ready to go!

B) Client Installation on WINDOWS or Other OS - Connecting to SAS Viya using REST API

Follow the steps in A) by replacing the **pip install** link in the **2nd step** with the following:

```
>> pip install https://github.com/sassoftware/python-swat/archive/v1.0.0.tar.gz
```

Online documentation

A sample machine learning code and other examples available on GitHub

<https://github.com/sassoftware/sas-viya-programming/tree/master/python/data-mining>
<https://github.com/sassoftware/sas-viya-programming/tree/master/communities>

Other materials available on SAS Communities:

- [Your First CAS Connection from Python](#)
- [Getting CAS Action Help from Python](#)
- [Loading Data from Python into CAS](#)
- [Getting Started with Creating Charts in Python](#)
- [Running Data Step from Python](#)
- [Getting a Python CASTable Object from an Existing CAS Table](#)
- [Simple Statistics in Python](#)