

Using the Force of Python and SAS Viya™ on Star Wars Fan Posts

Grace Heyne, Zencos Consulting, LLC

ABSTRACT

The wealth of information available on the Internet includes useful and interesting data, and it is easy to gather using the connection between SAS Viya™ and Python. Since the announcement at SAS Global Forum 2016 that SAS Viya™ supports running CAS actions within Python, SAS programmers have been exploring the many ways that SAS works well with Python. With easy transitions between the Python language and SAS, this connection allows programmers to use the best of both languages to build more powerful tools for data analysis. This paper includes specific examples of ways to use both SAS and Python together in Jupyter Notebooks to leverage Python's data collection, cleaning, and management tools, along with ways to gather data from the web and perform necessary transformations to prepare and analyze the data.

INTRODUCTION

This paper uses SAS Viya™ and Python in Jupyter Notebook to illustrate the benefits of pairing SAS Viya with open source tools. We found four advantages: access to Python's Pandas indexing and data collection capabilities, and SAS Cloud Analytics Services' (CAS) engine at the center of our project, SAS functionality in Python and SAS Visual Analytics 8.1's streamlined visual data exploration. We then walk through the data collection, data manipulation, and analysis in both tools while connected to CAS with the goal of empowering you to begin leveraging Python SWAT with SAS Viya, and to inform the tool selection decision process.

To uncover the realities of using these tools together, we went to work on collecting interesting and timely data. Star Wars: The Last Jedi comes out December 15, 2017, so we couldn't resist collecting and aggregating the necessary data to understand how fans are anticipating the movie's release. At the time of publishing, roughly 2.5 months before the release, the most popular topic of discussion is canon books. Movie discussions were a close second, and there are some theories involving Jar Jar Binks and Snoke being discussed. When strong emotions are present among posts, they are most often negative, perhaps due to the long wait between movie releases.

To gain these insights, we gather Reddit data using APIs in Python and tweets from SAS Viya's social media connections to compare the text analytics process using each platform to gather data, manipulate data, and mine text. We present the realities of each tool and aim to equip the reader with the knowledge to leverage the flexibility to connect SAS Viya to Python to make the jump from data collection to insights more efficient.

PYTHON WITH SAS

Fierce preferences for specific tools are prevalent in the analytics community. Equally prevalent is the idea that using the right tool for the job is the most efficient way to create results. Employing the Python SWAT package to connect to SAS Viya means you can easily extend your analytics toolset to efficiently achieve results and more quickly provide your stakeholders with the necessary data insights to help them move forward.

PYTHON AND JUPYTER NOTEBOOK INTRODUCTION

Python is a high-level programming language for general-purpose programming designed to be productive and readable. Python is gaining increasing popularity in the data science and analytics community because of its flexibility that allows for original work and the attributes that streamline the process for putting algorithms into production. In this paper, we write and execute Python code in Jupyter Notebook, an open-source web application that supports many languages and allows for the creation of documents that contain live code in-line with output and visualizations.

CONNECTING TO SAS VIYA FROM JUPYTER NOTEBOOK

When you connect to SAS from Python you are really establishing a connection to SAS CAS. If you can log into SAS Viya then you can connect to it from Jupyter Notebook. The server, port number (default=5570), username, and password are required to connect to CAS from Jupyter Notebook. The Python SWAT package must also be installed and imported. To learn more about installing and using the Python SWAT package please visit the SAS Software Git Hub, linked [here](#).

```
In [38]: conn = swat.CAS('ec2-52-86-133-89.compute-1.amazonaws.com', '5570', 'username', 'password')
```

Figure 1: Python SWAT Connection to CAS

Once the connection is established you can begin using both Python and SAS Viya capabilities. SAS Viya capabilities are delivered as CAS actions, which are the smallest unit of work for a CAS server. They are designed to do things like load data, transform data, compute statistics, perform analytics, and create output. CAS actions are organized into action sets such as Data Mining and Machine Learning, Statistics, Analytics, and System. All action sets licensed in SAS Viya are accessible through Python SWAT. Python SWAT also delivers many SAS specific Python capabilities within Python SWAT, e.g. a SAS Dataframe.

Once the connection is established, we can benefit from the flexible connection between Python SWAT and SAS Viya. We've found four major advantages.

1. THE PYTHON WAY OF DOING THINGS

Depending on the transformation needed, a SQL statement or a data step could be the best choice. In many situations, leveraging the indexing power of Pandas is a major advantage that allows for the creation of transformed data sets and new variables in the most efficient way. Once the transformations are completed, the new data set can be loaded to SAS CAS.

2. CREDIBILITY OF SAS IN PYTHON

SAS has a world class reputation in the analytics world. Python is also well known for its capabilities, but its methodologies and functions do not have the historic credibility that come with each SAS Action. Leveraging SAS's reliable procedures within Python can allow a typical Python coder to add credibility to their results and enable a SAS programmer to apply their SAS knowledge while learning to incorporate Python into their analytical skillset.

3. SPEEDY DATA EXPLORATION IN SAS VISUAL ANALYTICS 8.1

SAS Visual Analytics 8.1 is an industry-leading data visualization tool that you can load your newly transformed data into and quickly uncover a variety of insights. Easy access to data and results created in Python saves a huge amount of time and allows for the easy creation of beautiful visualizations and robust analytical insights, leading to more effective communication of results.

4. STORING DATA IN CAS INCREASES EFFICIENCY

CAS uses distributed processing, something you would need a Hadoop cluster to replicate in Python, making data processing exceedingly efficient. Further, bouncing between technologies often comes with high switching costs. These costs arise when data must be exported from one tool, imported into another, and exported yet again when work is complete. Connecting Python with SAS CAS allows you to load data once and access it from the tool of choice without incurring the time and effort costs of switching from one tool to another.

We now display these four advantages as we walk through the analytical process of gathering open-ended data from two separate sources, aggregating the data, and finally analyzing it in both technologies with CAS engine at the center.

DATA COLLECTION

We are interested in tapping into what Star Wars fans are discussing in the build up to the release of the new movie as we display the benefits of the connection between SAS Viya and Python. To get started we used SAS Viya to gather tweets related to the term 'Star Wars' and collect Reddit posts from a variety of Star Wars themed subreddits.

COLLECTING DATA FROM AN API

Application Programming Interfaces (APIs) are predefined methods of communication and tools for interacting with and building software applications. You can connect to APIs from Python to collect publicly available data. APIs often present their data in JavaScript Object Notation (JSON). JSON is built on a collection of name/value pairs or an object containing a set of values. This data-interchange format is easy to parse with Python.

Most major websites have APIs, and often there are Python wrappers, delivered as packages, specifically developed to make interacting with them increasingly easy. We used the Reddit API, which allows users to collect posts and comments from Reddit and specific subreddits. Python PRAW is the API wrapper for Reddit's API. Python PRAW provides us with functions built specifically for parsing the JSON that Reddit's API outputs, further easing the data collection process.

A connection must be established to the API and later to CAS to load data. Each API has specific connection requirement; in this case we had to register with Reddit.

Establish Connections to Reddit API and CAs

```
In [2]: reddit = praw.Reddit(client_id='bxxQjvkrt4tBJw',
                             client_secret='JmAZCiaUAqg27crpenBgkG0ABTc',
                             user_agent='Python',
                             username='username',
                             password='password')
```

Figure 2: API Connection to Reddit from Python

Once the API connection is established the next step is to parse and prepare the data to be loaded into CAS. This task is eased by the APIs wrapper and the use of Python's Pandas and NumPy packages.

```
post= list()
source=list()
for sub in subreddits:
    for submission in sub.hot(limit=50):
        post.append(submission.title)
        source.append(sub)

Reddit = pd.DataFrame(np.column_stack((source,post)),columns=['source','posts'])
```

Figure 3: Parsing through JSON using Python PRAW Wrapper

After data aggregation is complete you can upload the data to CAS.

```
In [46]: conn.upload_frame(Reddit, casout=conn.CASTable('Reddit', promote=True))

NOTE: Cloud Analytic Services made the uploaded file available as table REDDIT in caslib CASUSER(username).
NOTE: The table REDDIT has been created in caslib CASUSER(username) from binary data uploaded to Cloud Analytic Services.

Out[46]: CASTable('REDDIT', caslib='CASUSER(username)')
```

Figure 4: Uploading DataFrame to CAS

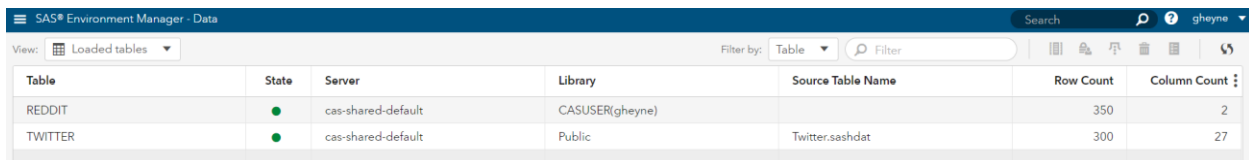
We see above that the data we collected and formatted into a DataFrame is successfully loaded to CAS as a CAS table.

UTILIZING DIRECT SOCIAL MEDIA CONNECTION

SAS Viya has a direct Social Media data import option for Twitter, Facebook, Google Analytics, and YouTube. We used the direct Social Media data import option within SAS Viya to collect tweets containing the keyword 'Star Wars'. We saved and loaded them into a CAS Table.

DATA PREPARATION

Having CAS at the center of our project provides a lot of flexibility and to easily switch between tools. The data that we collected through the two different means is now available in CAS. You can access the exact same tables from SAS Viya and Python without having to perform any data type conversions.



The screenshot shows the SAS Environment Manager - Data interface. At the top, there's a search bar and a user profile 'gheyne'. Below the header, there's a 'View:' dropdown set to 'Loaded tables' and a 'Filter by:' dropdown set to 'Table'. A table lists the loaded tables:

Table	State	Server	Library	Source Table Name	Row Count	Column Count
REDDIT	●	cas-shared-default	CASUSER(gheyne)		350	2
TWITTER	●	cas-shared-default	Public	Twitter.sashdat	300	27

Figure 5: Data from Python and SAS Viya both loaded in CAS

We need to combine the text data from Reddit and Twitter into one Star Wars dataset. You can perform this data aggregation task in both SAS Viya and Python. Both tools are using the CAS engine, so they both benefit from the distributed processing. Therefore, there is not one best way or tool to complete data manipulation and preparation.

PYTHON: DATA PREPARATION

Once you have loaded both CAS tables into our Python session you can perform transformations using pythonic methods.

```
In [89]: source = 'Twitter'
         twitter = pd.DataFrame({'source': source, 'posts':raw_twitter['body']})

In [90]: StarWars = Reddit.append(twitter)
```

Figure 6: Data aggregation in Python

SAS: DATA PREPARATION

You can also accomplish the exact same task using familiar PROCs and Data Steps within SAS Studio on SAS Viya.

```
proc sql;
  create table twitter as
  select 'twitter' as source, body as posts
  from public.twitter;
quit;

proc append base=reddit data=twitter FORCE;
run;
```

Figure 7: Data aggregation in SAS

Neither technology has the advantage for all tasks. Often data transformations are easier to accomplish by leveraging index selection within Python's Pandas or NumPy, but other times PROC SQL or a Data Step are better for the job. Due to the flexibility to use either tool, programmers are able to select the best tool for the job.

ANALYSIS

Coupling SAS Viya and Python provides you the opportunity to expand your problem-solving methods and analytical creativity through an almost endless variety of techniques available at our finger tips. Python has many useful text mining packages, such as gensim and NLTK, but as mentioned, calling SAS CAS Actions from Python is a great advantage. CAS actions come with the backing of SAS and are often simpler to implement and put into production than their Python counterparts, especially when SAS will be the implementation tool.

We demonstrate the use of CAS Actions in Python or SAS Viya using the Sentiment Analysis action, but it is important to remember that any licensed action in SAS Viya can be loaded and called in Python. In this example, the applySent action uses a domain-independent Sentiment Analysis Model (SAM) to score the sentiment of input text. You are free to provide a custom SAM when necessary.

PYTHON: CAS ACTIONS

The ability to load CAS actions is another feature provided within the Python SWAT package. To load you just employ the load action function for the action you wish to load.

```
In [86]: conn.loadactionset('sentimentanalysis')
NOTE: Added action set 'sentimentanalysis'.
Out[86]: $ actionset
          sentimentanalysis
elapsed 0.000251s · mem 0.0287MB
```

Figure 8: Loading CAS Action in Python

Once the action is loaded you can proceed with analysis. First, create the output table within CAS that will be loaded with the output from the Sentiment Analysis action.

```
In [84]: output = conn.CASTable('sentiment', replace=True)
          sentiment_object = starwars_df_long.applySent(
              casOut="output",
              docId='doc_id',
              text="posts"
          )
```

Figure 9: Calling CAS Action in Python

SAS: CAS ACTIONS

As in Python, you must also load CAS Actions when utilizing them in SAS Viya. PROC CAS is the SAS Viya procedure that enables you to interact with SAS Cloud Analytical Services from your SAS Client. Within the PROC CAS procedure you identify the session you are connecting to, load the needed action set, and call the action. Intuitively, you call the action with the action keyword, specifying the action set and then the specific action from that set.

```
proc cas;
  session mySession;
  loadactionset "sentimentAnalysis";

  action sentimentAnalysis.applySent;
  param
    docID="doc_id"
    text="posts"
    table={caslib="public", name="StarWars_Final"}
    casOut={caslib="casuser", name="outsent", replace=TRUE}
  ;
run;
```

Figure 10: Loading and Calling CAS Action in SAS Studio

Printing the first few rows in our results table confirms that whether calling a CAS Action from SAS Viya or from Python, the results are equivalent and equally reliable.

Obs	doc_id	_sentiment_	_probability_
1	1	Neutral	0.5
2	2	Negative	0.400000006
3	3	Neutral	0.5
4	4	Neutral	0.5
5	5	Neutral	0.5
6	6	Neutral	0.5
7	7	Positive	0.6000000238
8	8	Neutral	0.5
9	9	Positive	0.6000000238
10	10	Neutral	0.5
11	11	Neutral	0.5

In [91]: sentiment = conn.fetch(table=dict(name=sentiment))			
	doc_id	_sentiment_	_probability_
0	1.0	Neutral	0.5
1	2.0	Negative	0.4
2	3.0	Neutral	0.5
3	4.0	Neutral	0.5
4	5.0	Neutral	0.5
5	6.0	Neutral	0.5
6	7.0	Positive	0.6
7	8.0	Neutral	0.5
8	9.0	Positive	0.6
9	10.0	Neutral	0.5
10	11.0	Neutral	0.5
11	12.0	Neutral	0.5

Figure 11: Sentiment Action Output Comparison

Leveraging SAS Viya from within Python gives you the advantage of using of CAS Actions within Python and gives those who prefer Python to code in their preferred language but still produce results that can easily be accessed and used in collaboration with SAS Programming counterparts.

RESULTS

Presenting results brings us to the final advantage of leveraging SAS Viya with Python: Using SAS Visual Analytics on SAS Viya to quickly gain insights and create alluring visualizations. Whether you choose to perform your analysis in SAS Viya or Python, your results are loaded to CAS and are ready to be visualized or even analyzed using the capability provided within SAS Visual Analytics through SAS Visual Data Mining and Machine Learning (VDMML) or SAS Visual Statistics.

To gain further insight into what fans are talking about you can leverage the Text Topics object within SAS Visual Analytics on Viya. It performs topic discovery, analyzes sentiment, creates a word cloud, and displays a table with a direct view into the text you are analyzing. The full output is seen below, and we immediately see that 'star' and 'wars' are the most mentioned words and that there are five general topics being discussed within the data we collected.

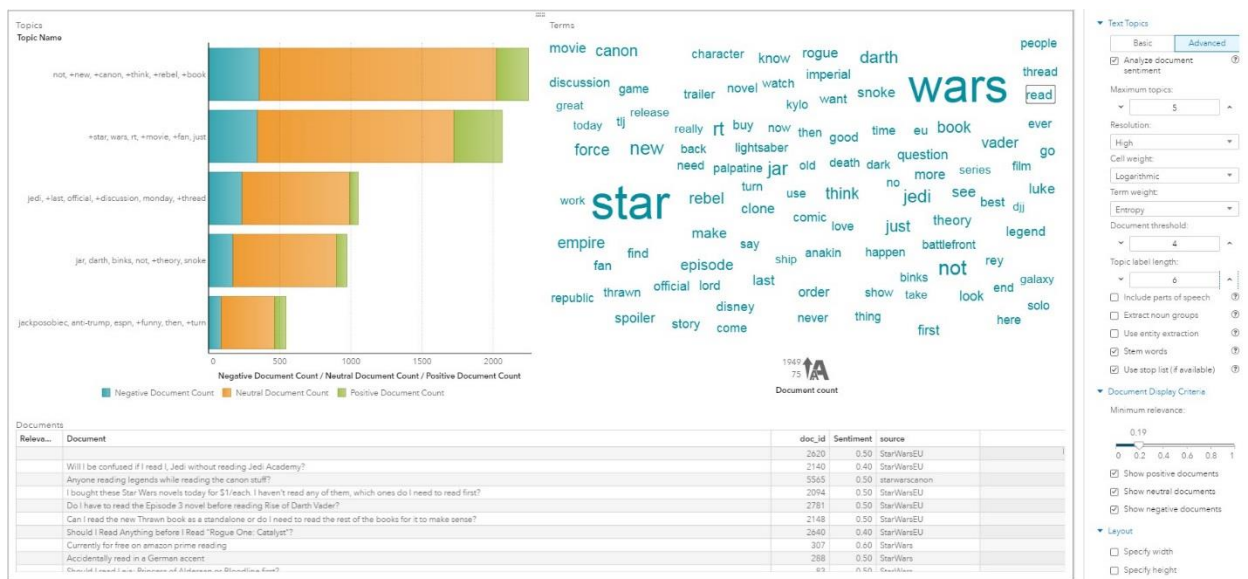


Figure 12: Text Topics Object in SAS Visual Analytics 8.1

Zooming in on the topics we can see that the highest portion of fan discussions are about the new canon, with other fans tweeting and retweeting about the movie, creating theories about Jar Jar Binks and Darth Vader, and discussing some current events seemingly unrelated to Star Wars.

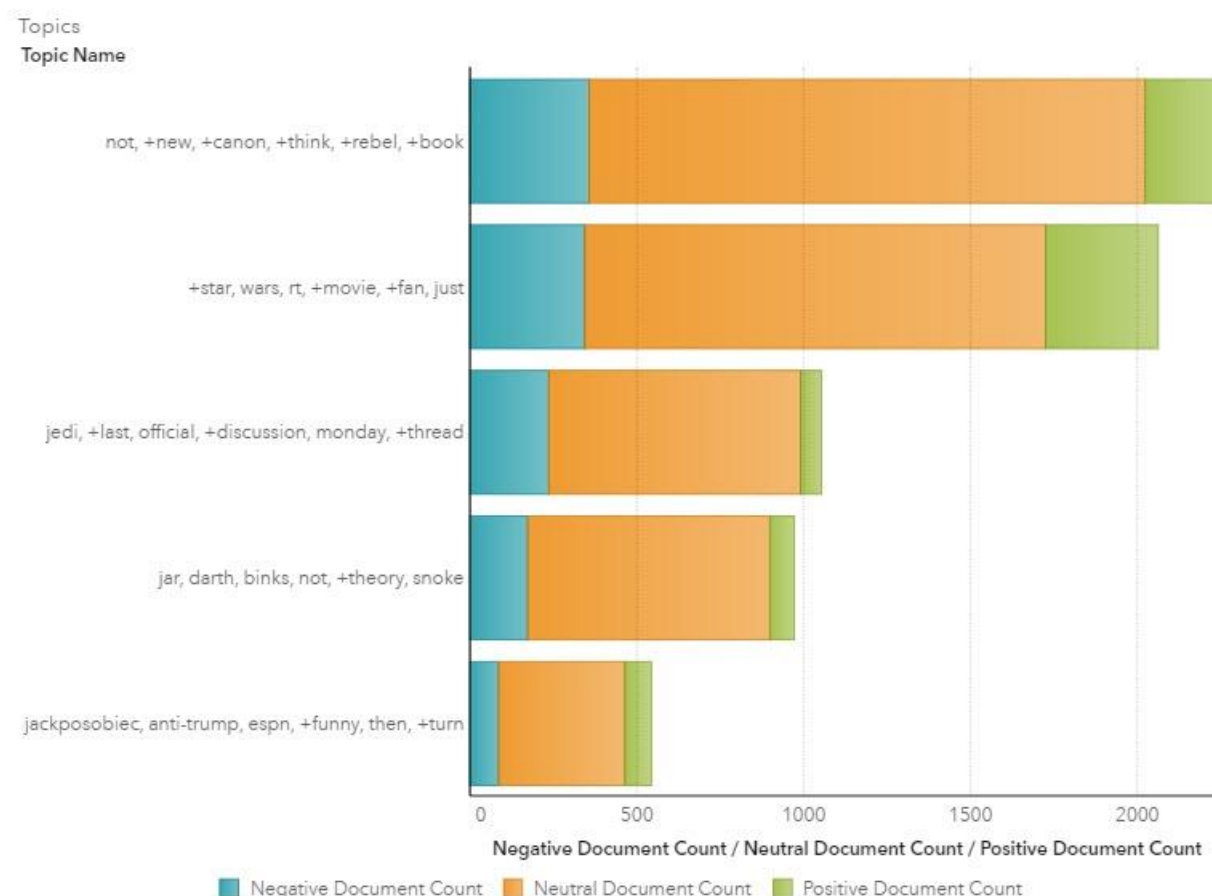


Figure 13: Text Topics Zoom

You have many options when it comes to text analytics, but keeping CAS at the center of any of your techniques allows you to always come back to SAS Visual Analytics for visualizations or robust point-and-click analysis with instant insights.

CONCLUSION

It is unlikely that analyzing Star Wars fan posts is relevant to your business. However, easily gathering social media postings about topics relevant to you and your customers likely is. Even more likely, creating efficient and robust analytical results by leveraging the flexibility to connect SAS Viya with Python SWAT is extremely relevant.

Utilizing the flexibility to connect SAS Viya with Python allows you to leverage the Python way of doing things, incorporate the credibility of SAS in Python, benefit from the lightning fast data processing with SAS CAS, and take advantage of the speedy data exploration and analytics in SAS Visual Analytics. We hope you will use the force provided in the flexibility of connecting SAS Viya to Python for good.

REFERENCES

Smith, Kevin. Meng, Xiangxiang. 2017. SAS Viya the Python Perspective. Cary, NC : SAS Institute Inc.

“SAS Scripting Wrapper for Analytics Transfer.” *SAS Software Git Hub Repository*. September 2017.
Available at <https://sassoftware.github.io/python-swat/>.

RECOMMENDED READING

- *SAS Viya the Python Perspective*
- *Python for Data Analysis*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Grace Heyne
Zencos Consulting, LLC
gheyne@zencos.com
www.zencos.com