Autumn Jacob
Professor Colleen Van Lent
SI 206 - 001

Final Project Report: Part 2

Reflection

Before starting the final project, one of my main goals was to create something over the top and highly complex. Even though this may have been possible, after starting the project my main concern was achieving a very thorough understanding of what I was coding. Throughout the course I would often times write code without fully understanding the how and why of what I was typing. For this project, my main goal was not based upon the outcome but more so on the creation processes. I wanted to create something that was simple and useful at the same time. Moreover, I wanted to make sure I incorporated apis that were relevant to me as well as others running the program. Another goal of mine was to create a more interactive experience for the user running my program and to provide a visual experience.

In the end, I achieved almost all of my goals with 3 apis, 3 SQL tables, and one visual representation. Even though I didn't attempt any extravagant api data collections, I accomplished my goal of thoroughly understanding what I was coding. Moreover, I was able to generate data tables with useful information. Each of my programs collects data in a different way and gets more advanced building upon my previous knowledge. For example, my use of the iTunes api collects 100 albums of artist, Frank Zappa, the release date and the number of tracks within each album and returns an SQL table of the organized. Using the Flickr api, I created a program that searches photos and outputs an SQL table with information from 100 photos such as the username, photo title, and tags. For my third program, I used Facebook to access my own posts. For this program I scraped my feed to produce an SQL table with 100 of my posts, the week day I posted on and the exact time of the day for each post. I then took my program a step further to create a visual report of my data. I used plot.ly to make a bar graph representing the 100 posts to figure out which day of the week I posted the most. I am very proud of the goals I achieved and hope to use these skills in the future to create even more exciting api projects.

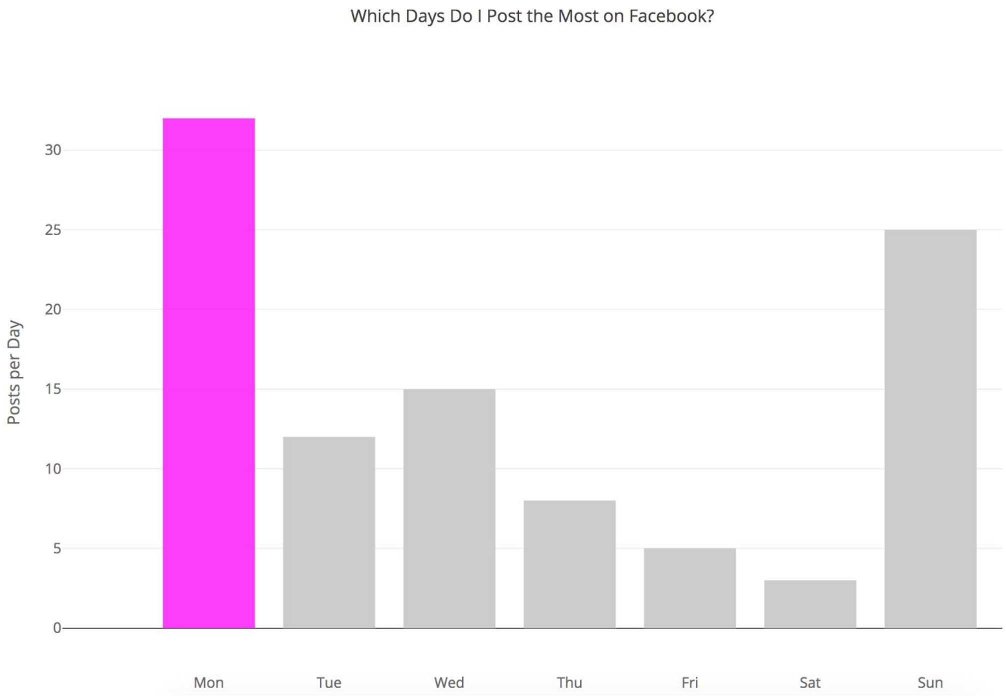
Creating these programs were a lot harder than I anticipated. I originally wanted to use the Ebay and Google api but getting access to the data was very difficult for me. Ebay's documentation was really hard to understand and didn't provide a clear direction for web scraping. A lot of the methods were for sellers trying to manipulate their selling pages rather than for people trying to collect data. Another problem that I struggled with was the Google Maps api. I could only find clarity for latitude and longitude and that is not the information that I wanted to pull. Without the pressure of submitting my databases, I think I would have been able to figure them out but this concern of time and deadlines was something I had to take note of. Moreover, I encountered many minor issues while coding such as indexing through dictionaries and being out of range when inserting data into the SQL tables.

Social Media Report

1. Using the iTunes api, I was able to create an SQL table with 100 of Frank Zappa's albums, the release date and the number of tracks within each album. With this data, I was easily able to find the albums with the most tracks and the dates in which they were released. This was interesting to see because I was able to compare albums and find out if later albums were more likely to have more or less songs. The album with the most songs included from this data is Halloween 77 (Live at Palladium, New York City, NY) with 159 songs.

2. Using the Flickr api, results may change based on the search term you provide. In my example, I used the search term "puppies". This produces an SQL output of data with the user id, title of the image and the tags for 100 images. This SQL table is a neat way to organize information surrounding photos. Flickr puts an emphasis on visuals so this SQL table provides a simple way to navigate information about a posted photo.

3. Using the Facebook api I gathered my own data to figure out the day of the week I post most often for a sample of 100 posts. I first made an SQL table of my data including the post id, the day of the week, and the time of the day that I posted. From there I created a function to sum up the posts based on the day of the week and then converted the data into a bar graph. From this sample, I found that I post the most on Monday's.



Instructions

1. My iTunes program is very simple to run. There is no need to provide an access code. Simply make sure to run “pip install python-itunes” in terminal and you should be good to go!
2. For my Flickr program, I have already provided an app key and secret within my program. Make sure you run “pip install flickr_api” in terminal. When you run the program make sure to interact with the code and provide a search term when prompted. If you have internet access feel free to search anything, if not please use the cache data and search “puppies”.
3. Within my Facebook program I have provided a proper access key and user id. Facebook keys do not last for more than an hour, hence you will be using the data from my cache. If problems arise you can get a key and user id from https://developers.facebook.com/tools/explorer though this should not be necessary. Lastly, make sure you have Facebook and Plot.ly installed in terminal. You can make sure you have the packages installed by running “ virtualenv facebookenv
   source facebookenv/bin/activate
   pip install -e git+https://github.com/mobolic/facebook-sdk.git#egg=facebook-sdk”

Documentation

Date	Issue Description	Location of Resource	Result
12/4/17 12/5/17 12/8/17	Accessing Facebook API and locating correct parameters	https://developers.facebook.com/tools/explorer/	Successful
12/4/17	Converting timestamp	https://docs.python.org/2/library/datetime.html	Successful
12/4/17	Selecting data from SQL	http://www.dofactory.com/sql/select	Successful
12/4/17	Issues with elif syntax	https://www.tutorialspoint.com/python/python_if_else.htm	Successful
12/4/17 12/5/17 12/8/17 12/9/17	Viewing dictionaries	http://jsoneditoronline.org/	Very helpful resource
12/8/17	Accessing iTunes API and locating	https://affiliate.itunes.apple.co	Successful

	correct parameters	m/resources/documentation/itunes-store-web-service-search-api/#searchexamples	
12/8/17	Confused with pip iTunes installation	https://github.com/ocelma/python-itunes	Successful
12/8/17	Using class methods - forgetting init	http://sthurlow.com/python/lesson08/	Successful
12/8/17	Locating correct parameters	https://www.flickr.com/services/api/flickr.photos.search.html	Successful
12/9/17	Applying for key and secret	https://www.flickr.com/services/apps/create/apply	Successful
12/9/17	Error "binding parameter 0 - probably unsupported type". Problems with SQL	https://stackoverflow.com/questions/21981709/error-binding-parameter-0-probably-unsupported-type	Not very helpful - ending up using tuple and worked fine
12/12/17	Understanding plot.ly usage for Python	https://plot.ly/python/bar-charts/#basic-bar-chart	Successful
12/12/17	Looking up color code	http://www.december.com/html/spec/colorrgbadec.html	Successful