**Final Project Report**

**Goals**

* Use Instagram’s API to access 100 of my own posts and check their likes and comments. Then I would analyze any trends in them.
* Use Facebook’s API and assess my 100 interactions through how many posts I have liked. Through that I can see which days I have like the most number of pics and then plot them onto a graph to see how I active I am.
* Also use Facebook’s API to get friends hometown and plot their location on a map.
* Use SQLite to input the above data in tables.
* Use Tableau public as the visualization tool.

**Achieved**

I used Instagram’s API to get the details on my posts. I used the api to get information of my last 20 posts: Picture ID, date posted, caption, comments, and likes. Additionally, Instagram uses unix timestamps so I had to convert that using the python datetime library.

With the GraphAPI, I accessed information of 25 of my friends: Name, userID, gender, hometown, location.

Graph

For Instagram, I wanted to create a bar chart with my last 20 posts and compare the number of likes and comments I got on each.

For Facebook, I was able to create a pie chart showing the gender distribution.

**Problems**

With Instagram’s API, my app was not approved so it was categorized under sandbox. As a result, there were a lot of restrictions on it. I was limited to only 20 posts (not 100 like I initially stated).

I struggled figuring out the created\_time with Instagram because they used unix timestamps. However, I resolved the issue with help from StackOverflow (<https://stackoverflow.com/questions/3682748/converting-unix-timestamp-string-to-readable-date-in-python>).

The data from the facebook api is limited to friends who installed the app. Only their information is returned in API v2.0 and higher. When I tested it, I only had 26 friends who have downloaded it. Resultantly, I used 25 of my friends. Additionally, Facebook’s api token changes frequently. However, I was able to overcome this problem by caching the data. So I have added a try and except statement.

The other idea was to get the hometown of my friends and map their location on a map. However, only 12 people in total had information for either their hometown or current residing location, so I did not end up creating the map.

In my initial project proposal, I stated that I would use tableau public to plot my graphs. However, I switched it to Plotly because my GSI suggested it.

Plotly prints this output, I could not get rid of it.

/anaconda3/lib/python3.6/site-packages/plotly/tools.py:1410: UserWarning:

Looks like you used a newline character: '\n'.

Plotly uses a subset of HTML escape characters

to do things like newline (<br>), bold (<b></b>),

italics (<i></i>), etc. Your newline characters

have been converted to '<br>' so they will show

up right on your Plotly figure!

**Social media “report”**

Instagram bar graph: From the bar graph, there is a huge discrepancy between the number of likes and comments I receive on each post. I receive 15x/20x more likes than comments. In addition to that, there is an increasing trend in the number of likes I receive on my posts. This growth, in terms of social media, is a good stat.

Facebook gender pie chart: It is evident that from this sample of friends that I have more male friends than female. The number was 80% to be exact. However, I do not think this is an accurate representation of my whole friend group on Facebook.

Instagram Graph: <https://plot.ly/~suhasmaganti/2>

Facebook Graph: <https://plot.ly/~suhasmaganti/4>

**Instructions**

The zipped file should have 4 files:

1. FinalProject.py – All of my python code
2. api\_info.py – Tokens to access api’s
3. FinalProject\_cache.json – Cache file
4. FinalProject.sqlite – Database files for the information collected
5. FinalProject\_InitalCache.json – an extra file in case Facebook’s accesstoken fails

When you run my python file, it should create a cache file (json), database (sqlite) and graphs on plotly which can be accessed using the links given above.

As mentioned above, Facebook frequently updates the access\_token. However, the file should still run with the InitialCache file above.