CONCEPT

Create a program that allows users to input a city and find the most "instagrammable" spots in the area. The program will scrape Instagram for the most liked posts and most recent posts in a city and filter out external factors, in order to return the best specific locations to take an Instagram photo and get the most likes per post.

DATA COLLECTION

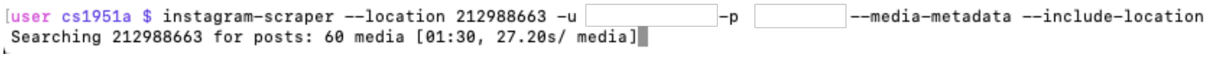
https://github.com/rarcega/instagram-scraper

We plan to use the above Instagram scraper and its --location flag to find the top posts in a city, as well as its --latest tag to find the most recent posts in a city. We'll then use the --user tag on the posts we've found to seed more post locations in the city to analyze.

When searching by --location, the scraper takes in a location ID. This we have to manually find through searching for the location on Instagram Web. We found a tutorial on how to do this here:
https://havecamerawilltravel.com/photographer/instagram-location-search/

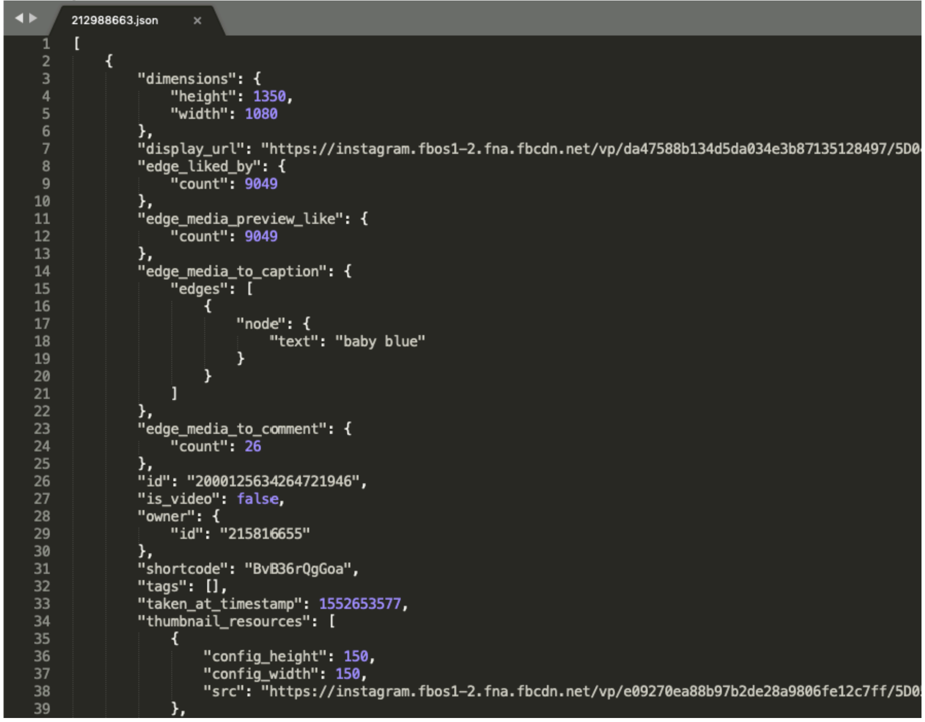
The scraper we use require a username and password. We have been using a team member’s personal Instagram account, although we might create a dummy account for research purposes for this class.

Command line orders:



```
[user cs1951a $ instagram-scraper --location 212988663 -u [redacted] -p [redacted] --media-metadata --include-location
 Searching 212988663 for posts: 60 media [01:30, 27.20s/ media]
```

Sample JSON Result for New York, New York



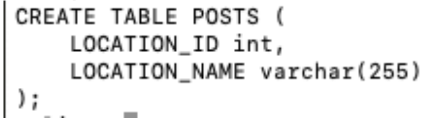
```
[
    {
        "dimensions": {
            "height": 1350,
            "width": 1080
        },
        "display_url": "https://instagram.fbos1-2.fna.fbcdn.net/vp/da47588b134d5da034e3b87135128497/5D0
        "edge_liked_by": {
            "count": 9049
        },
        "edge_media_preview_like": {
            "count": 9049
        },
        "edge_media_to_caption": {
            "edges": [
                {
                    "node": {
                        "text": "baby blue"
                    }
                }
            ]
        },
        "edge_media_to_comment": {
            "count": 26
        },
        "id": "2000125634264721946",
        "is_video": false,
        "owner": {
            "id": "215816655"
        },
        "shortcode": "BvB36rQgGoa",
        "tags": [],
        "taken_at_timestamp": 1552653577,
        "thumbnail_resources": [
            {
                "config_height": 150,
                "config_width": 150,
                "src": "https://instagram.fbos1-2.fna.fbcdn.net/vp/e09270ea88b97b2de28a9806fe12c7ff/5D0
            },
```

DATA CLEANING

We plan to store the post data we collect in a SQL database. We want two tables - users and posts. The user table will store user data (handle, number of followers, average number of likes per post, post ids, if they're an influencer). The post table will store post data (link to post, number of likes, location name, location id, user id, time).

Posts will only include posts that have been created 2 or more days ago – we only want posts that have had enough time to accumulate a representative number of likes.

Sample POSTS table: TBD column fields.



```
CREATE TABLE POSTS (
    LOCATION_ID int,
    LOCATION_NAME varchar(255)
);
```

DATA ANALYSIS

At a minimum, we'll use the database described above to get an ordered list of where to go to get the most likes. We'll look at all the stored posts for a location and find its mean number of likes, also considering median and variance and filtering out outliers. The intent is to identify locations that draw attention for the location itself and not for external factors.

We'll then sort the results based on what locations have the highest average number of likes.

RECENT DEVELOPMENTS

Since our last check-in, we've set up our Google Cloud platform and started collecting sample data.

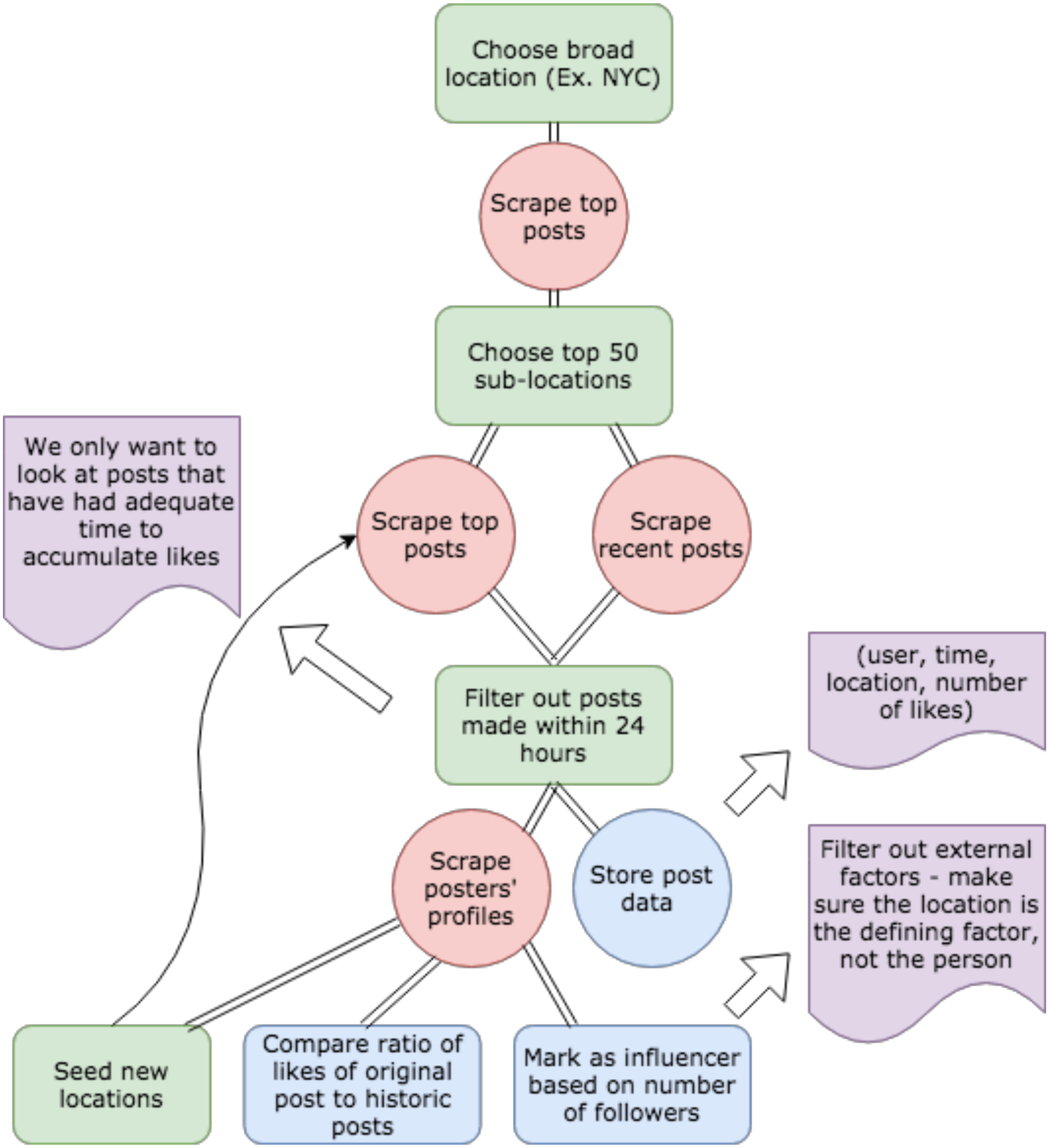
TIME SINKS

1. Working with the instagram-scraper and figuring out the location and meta data it provides. We have forked the repository so that we could customize it even further.
2. Determining the scope of this project: what are the fields that we want to make customizable to the user? By location, time of day, day of week, influencer/ non-influencer? How can we toggle these options and have that reflected in our code? Do we want to create a command-line tool, or simply the pipeline with executable code for future implementation?

NEXT STEPS

- Create sizable data-base
- Test hypotheses for best filtering pipelines

DATA COLLECTION PIPELINE


Choose broad location (Ex. NYC)
Scrape top posts
Choose top 50 sub-locations
We only want to look at posts that have had adequate time to accumulate likes
Scrape top posts
Scrape recent posts
Filter out posts made within 24 hours
(user, time, location, number of likes)
Scrape posters' profiles
Store post data
Filter out external factors - make sure the location is the defining factor, not the person
Seed new locations
Compare ratio of likes of original post to historic posts
Mark as influencer based on number of followers