

How to Get 5000 Instagram Followers

Sloan Woodberry

OVERVIEW/HYPOTHESIS

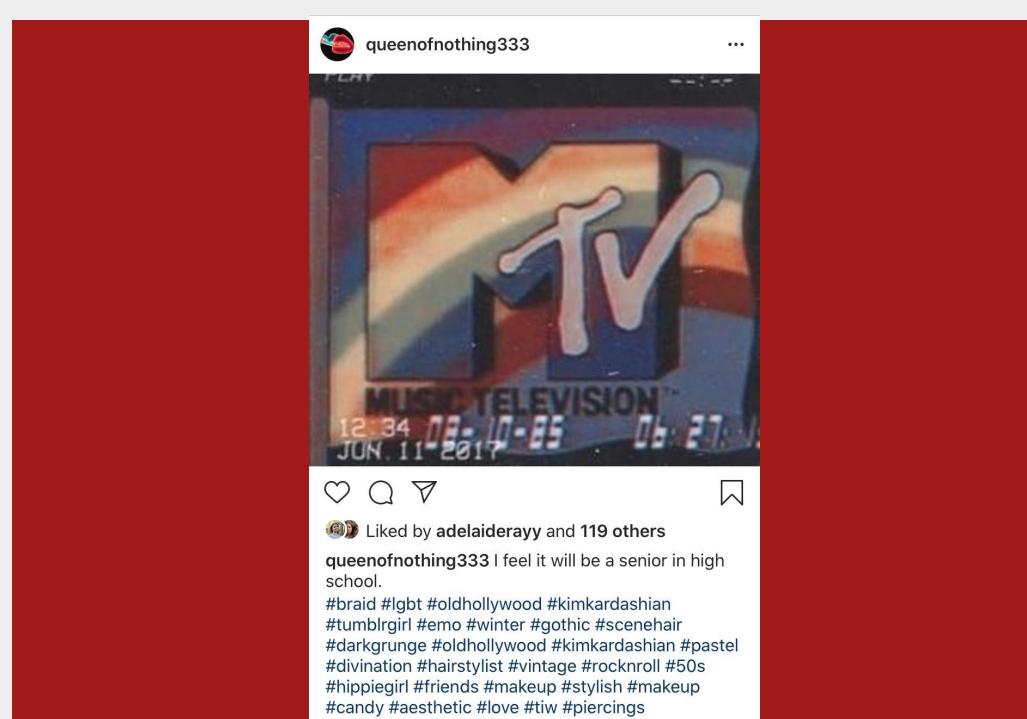
Whether it's influencing elections or testing AI text generation, we have all read about social media robots in the news. They range from scary to impressive to a mixture of both. All of this, along with natural curiosity, prompted me to build my own instagram robot with the goal of gaining 5000 followers in less than 2 months. This project touched on a lot of different places such as coding and data analysis just to name a few.

RESEARCH AND GOALS

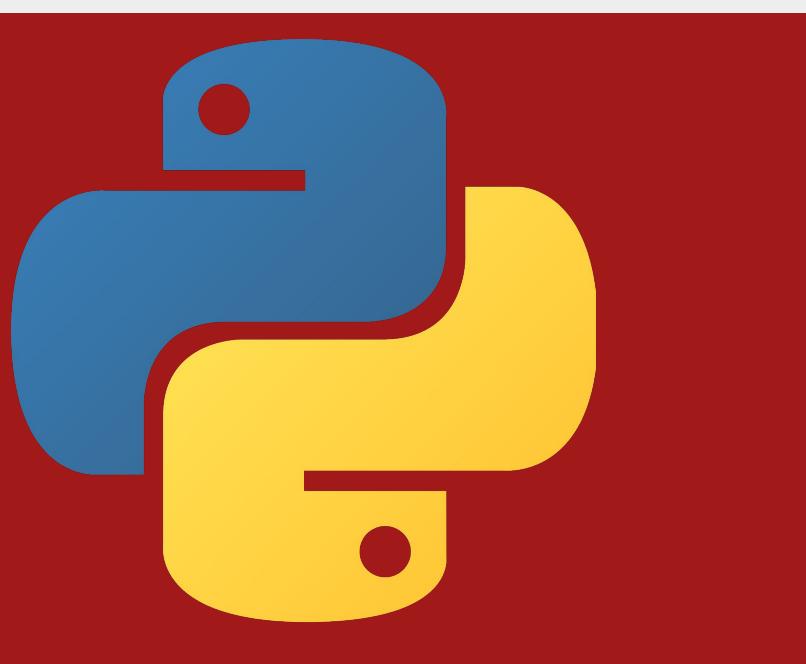
My hope for this project was to get more comfortable using python as well as learn how to navigate websites using code. Another thing I had to do in order to gain followers was program my robot to successfully interact with instagram. This meant that I had to code it to like posts, follow users, comment, and generate posts as frequently as possible without instagram realizing that I was a bot and suspending further actions. Each of these interactions alone was difficult to code which is why I decided to research code libraries to help me. While I used a series of different libraries the most prominent ones were selenium and instapy. Selenium allows you to code specific actions, for example clicking a like button, and then execute them in google chrome. Instapy is similar to selenium but is more specifically geared towards instagram and is able to help run some basic analytics on a given profile.

INITIAL GOALS

Along with the long term goal of getting 5000 followers over the course of 2 months I had smaller goals for what specific actions I wanted for my bot. These mainly consisted of "user actions" such as liking posts, following users, unfollowing users, and making posts.



I found that with the help of the instapy library I was able to finish the like and follow functions relatively quickly. My robot followed people who posted under a specific hashtag that was a part of a list that I had previously made. My function for liking was relatively similar with my robot only liking posts that appeared under a specific hashtag that was part of the same list I just mentioned. The unfollow feature was a bit more difficult but eventually I was able to make it so my robot only unfollowed the users that were not following it back. The most difficult out of all these functions however was the post function. My goal was to create a post with a caption generated from a series of hashtags with the image being chosen at random from a premade library. The hashtags would then be put below the generated caption and the entire thing would be posted. The big issue with this was that instagram only allows people to post using the mobile app, not using a desktop browser which is what I was using. In order to fix this issue, I found another python library that emulates a mobile device in order to log in to my account and make a post.



TESTING AND ITERATION

My iteration process consisted of both debugging as well as implementing a new following algorithm that helped grow my following.

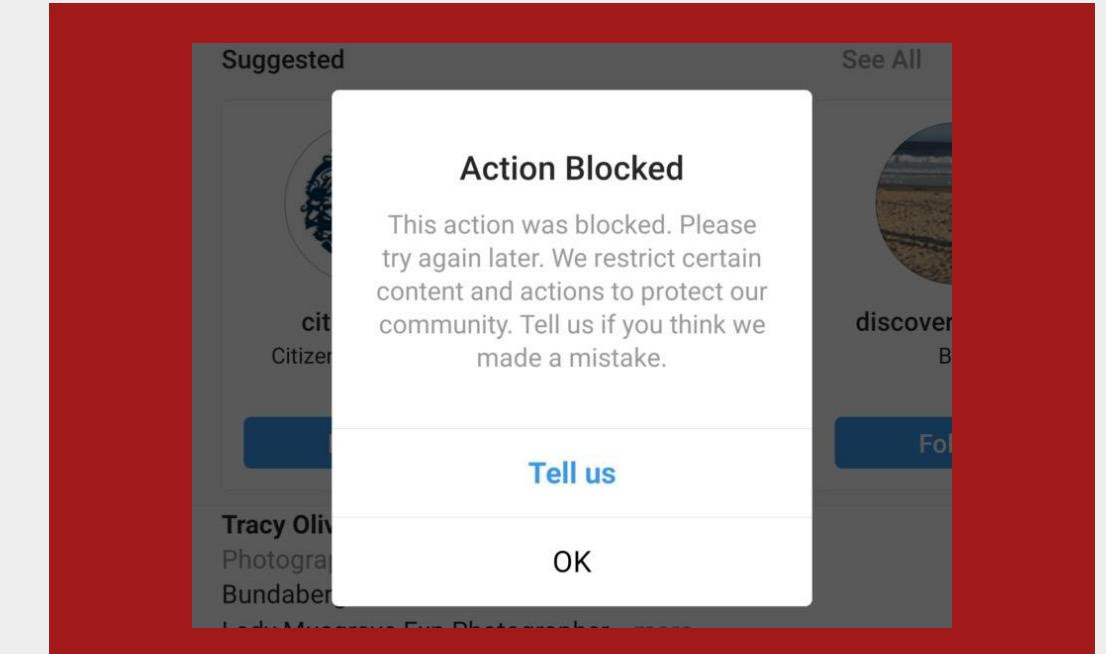
DEBUGGING

One of the biggest issues for me initially was instagram blocking my account from performing various actions for certain periods of time. There were three major causes of this problem: timing, hierarchy of actions, and source of actions. **Timing:** The first 2 times I got blocked was after I coded a function that was able to like 20 posts in roughly a minute. While this may have been efficient, it made my account appear suspicious so it was flagged. After this I added in a delay of 3 seconds after any given action. While this worked for awhile, eventually I was flagged again after running my like function for an hour. My final solution was to set random delays between 2 -30 seconds based off the type of action being performed.

Hierarchy of Actions: Through trial and error I discovered that not all actions are created equal and that certain actions are more likely to result in you getting flagged than others. I tested this by repeatedly performing certain actions and timing how long it took for my account to be flagged. From most likely to least likely to result in a flag we have posting, unfollowing, following, commenting, and liking. This means that I needed to adjust the delay period between these actions accordingly, for example taking longer in between unfollows than between likes. I also had to adjust the delay in between groups of actions, for example if I was performing a series of follows I would have to wait longer to resume work than for a series of likes.

Source of Actions: After awhile I noticed that when I ran my robot through my computer, as opposed to someone else's, I was much more likely to be flagged. I figured it was a result of the amount of requests I was sending so I decided to implement a rotating proxy in order to help that issue.

```
12 # @file Me_mainTest_.json
13
14 options = webdriver.ChromeOptions()
15 options.add_argument("user-data-dir=C:/Users/sloanwoodberry/Library/Application Support/Google/Chrome/Pr
16
17 driver = webdriver.Chrome(executable_path="bin/chromedriver",chrome_options=options)
18
19 #driver.get('https://www.instagram.com/')
20 carotIsBlocked=False
21
22 # for cookie in pickle.load(open("cookies.pkl","rb")):
23 #     if "expiry" in cookie:
24 #         del cookie["expiry"]
25 #         pickle.dump(cookie,open("cookies.pkl","wb"))
26
27 def sleep1Pause():
28     sleep(random.randrange(1,5)*random.random())
29
30 driver.get('https://www.instagram.com/accounts/login/')
31 sleep(random.randrange(2,12)*random.random())
32 # If hasn't logged in yet, print("Carol needs to login")
33 #     print("Carol needs to login")
34 #     username = driver.find_element_by_name('username')
35 #     username.send_keys('queenofnothing333')
36 #     smallPause()
37 #     password = driver.find_element_by_xpath("//input[@aria-label='Password']")
38 #     password.send_keys("shephethisworks")
39 #     smallPause()
40 #     login = driver.find_element_by_xpath('//*[@type="submit"]')
41 #     login.click()
42 #     print("Carol is in")
43 #     sleep(0.3)
44 else:
45     print("Carol doesn't need to login")
46 def isLocked():
47     try:
48         driver.find_element_by_xpath("//div[contains(text()), 'This action was blocked. Please try again lat
49         carotIsBlocked=True
50     sleep(random.randrange(1800, 60400, 4))
51 except:
52     pass
```



NEW AND IMPROVED FOLLOWING

My initial method of gaining followers was to comb through popular hashtags and follow the most recent accounts that came up. While this method worked I felt that it could be improved upon. I had noticed a few common trends in the people that had been following me back. To explain clearly I am going to define some terms

non-follower- someone who a given person follows but that follow is not reciprocated

full follower- someone who a given person follows and that follow is reciprocated

unfollower- someone who follows a given person but that give person does not reciprocate the follow

follow ratio - the ratio of the number of people a user is following divided by the number of followers they have

I found that users with a high amount of non-followers had a higher tendency to follow me back than those who had a large amount of unfollowers. Additionally, those with a high amount of non-followers also tended to have full followers who also had a large amount of non-followers. Lastly, as one might assume, the users with a lot of non-followers had a larger follow ratio when compared to the average user. Responding to these trends, I created a program that looked through my list of full followers and randomly picked one with a follow ratio of 1.4 or above. From there, I would go through that user's full followers and took users that had follow ratios of 1.4 or above/a large amount of non-followers and added them to a list. I would repeat this process until my list had reached a given length, then I shuffled the order of followers in the list to make my actions appear more random and followed the users on the list.

After implementing this new following program the number of followers I had grew much faster which helped me get closer to my goal.

RESULTS/CONCLUSION

Unfortunately, after my robot was able to surpass 8,000 followers the instapy library stopped working. In the future, I would want to go back and rewrite all of the important functions myself (like, follow, post, etc) using selenium so I would not have to rely so much on instapy. I would also like to adapt my posting function to include characters like emojis so I can add them to my posts and potentially hashtags as well. Lastly, I would like to run more of my own programs to track my robots analytics instead of relying on instapy. This data would include some of the more effective hashtags i've used in posts, the profiles of people who like/follow me, and the overall change in my following.