

### === LinkedIn Employee Scraper ===

- Creator: Ashwin Madhavan
- Tags: linkedin, scraper, tool, employee, python
- Best ran on Jupyter notebook
- Python 3.8.8 or greater
- Requires selenium, beautifulsoup4 package (should automatically download when code is executed)

### === Description ===

The purpose of this code is to help the sales team speed up the process of finding potential leads by automatically sifting through the employees on specific company LinkedIn pages and scrapping relevant data. The output is an excel file with the following information:

Organization Name	Organization Name URL	Website	Company LinkedIn	Employee Title	Employee LinkedIn	Employee Name	Strength Index
Outreach	<a href="https://www.crunchbase.com/organization/outreach">https://www.crunchbase.com/organization/outreach</a>	<a href="https://www.outreach.io/">https://www.outreach.io/</a>	<a href="https://www.linkedin.com/company/outreach-sea/">https://www.linkedin.com/company/outreach-sea/</a>	Principal Machine Learning Engineer at Outreach	<a href="https://www.linkedin.com/in/robovse/">https://www.linkedin.com/in/robovse/</a>	Robert (Yan) Xue	3
				Staff Machine Learning Engineer at Outreach	<a href="https://www.linkedin.com/in/yufan-xue-37112b79/">https://www.linkedin.com/in/yufan-xue-37112b79/</a>	Yufan Xue	3
				Machine Learning Engineer at Outreach	<a href="https://www.linkedin.com/in/jianchen/">https://www.linkedin.com/in/jianchen/</a>	Uang-Hu Chen	3
				Machine Learning Engineer at Outreach	<a href="https://www.linkedin.com/in/ygcag/">https://www.linkedin.com/in/ygcag/</a>	Yong-Gang Cao	3
				Senior Staff Software Engineer at Outreach	<a href="https://www.linkedin.com/in/stefan-lopez-34b7914b/">https://www.linkedin.com/in/stefan-lopez-34b7914b/</a>	Stefan Lopez	2
DataRobot	<a href="https://www.crunchbase.com/organization/datarobot">https://www.crunchbase.com/organization/datarobot</a>	<a href="https://datarobot.com">https://datarobot.com</a>	<a href="https://www.linkedin.com/company/datarobot">https://www.linkedin.com/company/datarobot</a>	Senior Machine Learning Engineer at DataRobot	<a href="https://www.linkedin.com/in/anatoli-stehni-8936355b/">https://www.linkedin.com/in/anatoli-stehni-8936355b/</a>	Anatoli Stehni	4
				DataRobot, Senior Machine Learning Engineer & Data Scientist, Founder of AI Consulting.	<a href="https://www.linkedin.com/in/yuryguts/">https://www.linkedin.com/in/yuryguts/</a>	Yury Guts	4
				Senior Machine Learning Engineer at DataRobot	<a href="https://www.linkedin.com/in/magshakov/">https://www.linkedin.com/in/magshakov/</a>	Mikhailo Poliakov	4
				Senior Machine Learning Engineer at DataRobot, MSc in Mathematics	<a href="https://www.linkedin.com/in/lukas-traub/">https://www.linkedin.com/in/lukas-traub/</a>	Lukas Traub	4
				Senior Machine Learning Engineer @ DataRobot	<a href="https://www.linkedin.com/in/jmbannon/">https://www.linkedin.com/in/jmbannon/</a>	Jesse Bannon	4

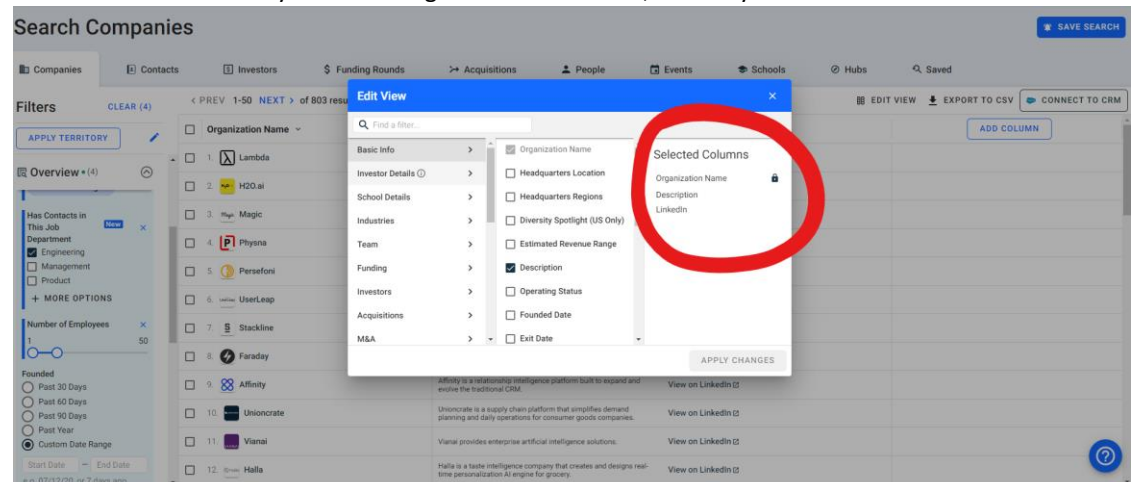
The top 5 matches in the above example are found using key search words ["machine", "learning", "senior", "director", "engineering"]. And the number of matches correlate to the strength index.

### === Downloading From GitHub ===

1. Go to <https://github.com/ashwin-madhavan/linkedinEmployeeScraperTool>
2. Click the green download code button
3. Unzip the folder and move the unzipped folder to where you prefer
4. Open jupyter notebook and navigate to the folder that contains the code
5. Open the folder, then click the [LinkedInAutomationV2.ipynb](#) file

### === Steps to Executing the Code ===

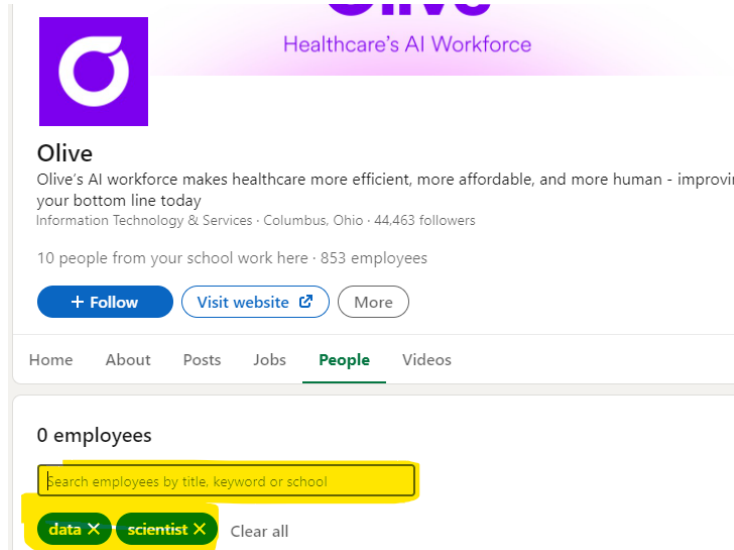
1. Go to CrunchBase and filter the companies you want to scrape for LinkedIn information.
2. Click edit view and make sure only the following are in the columns, and they must be in this order:



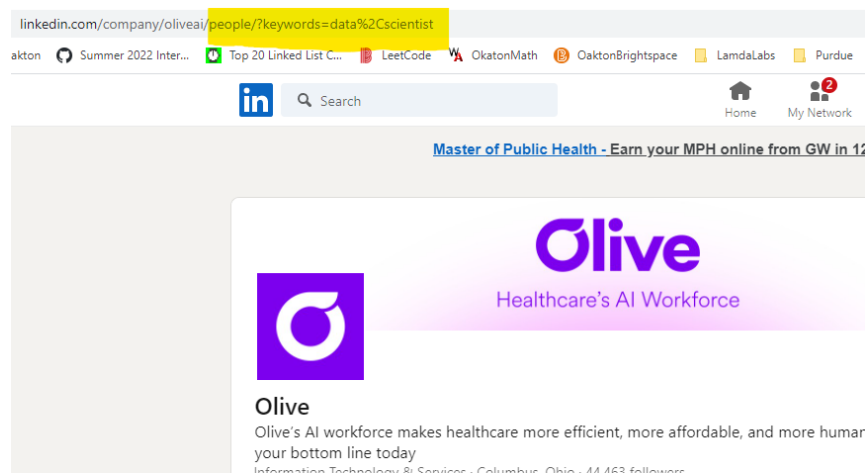
- a.
3. Export the csv file and place it in the folder the python code is located



- a. *\*All in same file*
4. Follow TODO comments (instructions on code)
  - a. Download Chrome driver and set file location
  - b. Enter burner LinkedIn account login information
  - c. Enter the name of the csv file being processed from CrunchBase
  - d. Name the output file followed by .xlsx
  - e. Enter new LinkedIn Search URL if needed (original code set to "machine learning engineer")
    - i. Under company page>>>people/employee search page enter search words



- ii. Copy the URL after the "/" before people



- iii. Paste as String equal to searchURLEnding Variable

- f. Add/remove search words into the strengthIndexWordCheck array (must be lowercase and comma separated)
- g. Set the number of employees you want displayed from the search to the output csv file

5. Hit RUN to see code execute
  - a. Make sure output file is closed if it happens to be open already
  - b. Code should open LinkedIn on chrome driver and interact with the page automatically
  - c. Code Purposely Runs slowly to bypass LinkedIn automation detection software
  - d. Once script outputs “finished!” open the output file(should be located in same folder as python script) and view the results

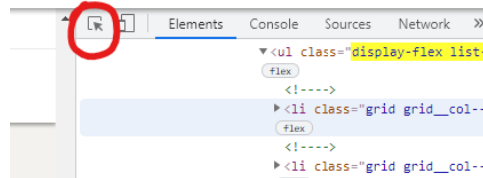
### === Additional ===

\*Search for “ADDITIONAL TODO” comments to find points needed to be updated\*

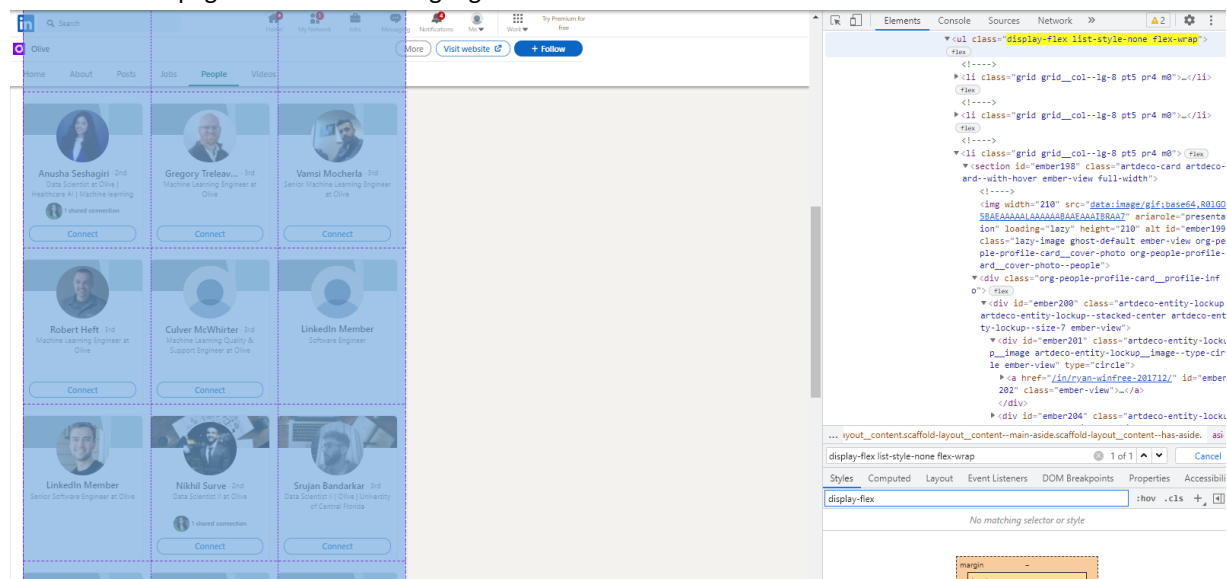
To bypass LinkedIn automation detection software the code waits 15 seconds before moving to the next profile. If you wish to change the delay adjust the value in the time.sleep() function. Keep in mind the value is in seconds

The script works by searching through HTML to find information and therefore if LinkedIn happens to update HTML the code will need to update the following to work.

1. Update “getNewProfileIDs” function
  - a. Go to a company page and the people/company explore page
  - b. Inspect Element the page
  - c. Click the box with the arrow

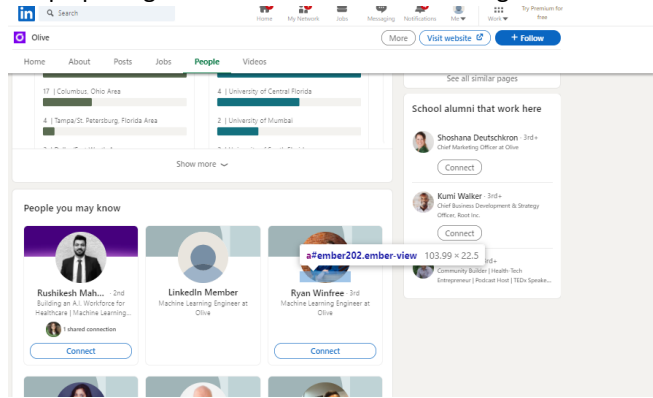


- d. Hover over the page to find the blue highlight that matches below



- e. Update the soup.find() parameters with the respective tag type and class title

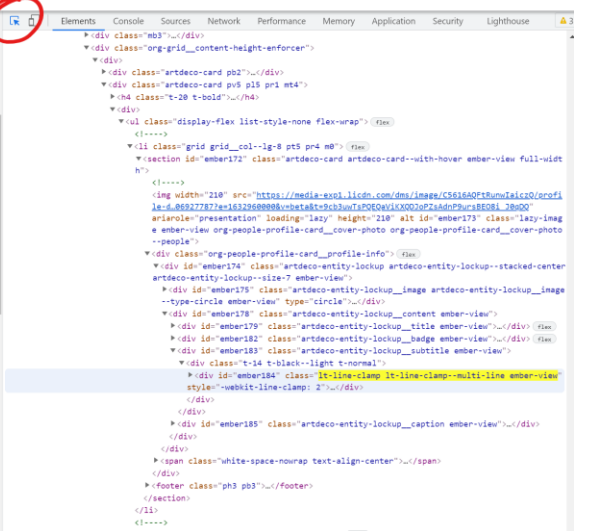
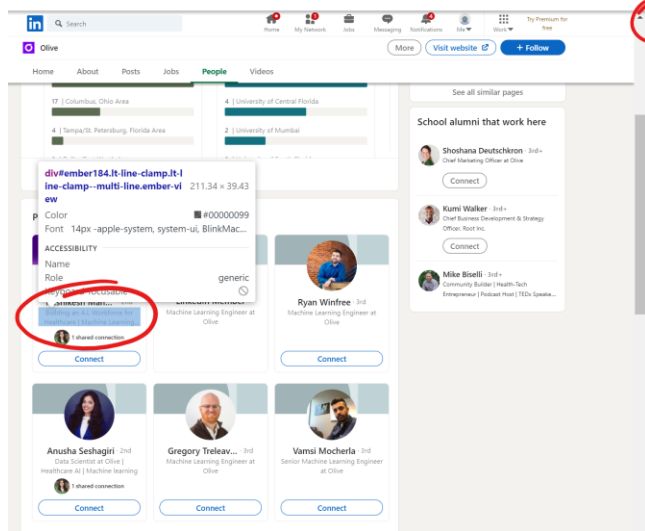
- f. Keep opening containers until an “<a href” tag is found



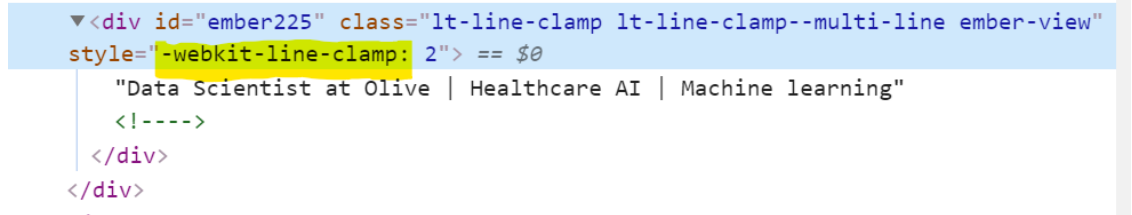
- g. Update the pav.findAll() parameters with the respective tag and class title

2. Update “getNewProfileDescriptions” function

- a. Follow steps a-e from #1
- b. Click on the box with the Arrow. Then hover over the profile description



- c. Update the pav.findAll() parameters with the respective tag and class title
- d. Update description.find() parameters with the style tag name



3. Update “getNewProfileNames” function

- a. Follow steps a-e from #1
- b. Click on the box with the Arrow. Then hover over the profile name



- d. Update the `pav.findAll()` parameters with the respective tag and class title