**Study Method**

# Data Collection

# Social Media platform

# Twitter is the main social media platform that I have used here because the twitter tweets are some kind of more accurate and reliable than other social media platforms according to the current findings on the Internet. Twitter is mainly used for share valuable information with different parties and it has become the quickest way to share news nowadays.

# Therefore, here I will be using twitter tweets as the main data collection.

# Dataset Collecting Procedure

# Due to COVID-19, Twitter API developers permission application review times might take longer than usual therefore the best available option had was to use selenium to collect tweets out of twitter regarding covid-19 lockdown.

# *Therefore, the current tweets collecting selenium algorithm is mentioned below:*

**import** selenium  
**from** selenium **import** webdriver  
**from** selenium.webdriver.common.keys **import** Keys  
**from** selenium.webdriver.common.by **import** By  
**from** selenium.webdriver.support.ui **import** WebDriverWait  
**from** selenium.webdriver.support **import** expected\_conditions **as** EC  
**from** selenium.common.exceptions **import** TimeoutException, StaleElementReferenceException  
**from** bs4 **import** BeautifulSoup **as** bs  
**import** time  
  
  
**def** init\_driver():  
 *# initiate the driver:* driver = webdriver.Chrome()  
  
 *# set a default wait time for the browser [5 seconds here]:* driver.wait = WebDriverWait(driver, 5)  
  
 **return** driver  
  
  
**def** close\_driver(driver):  
 driver.close()  
  
 **return  
  
  
def** login\_twitter(driver, username, password):  
 *# open the web page in the browser:* driver.get(**"https://twitter.com/login"**)  
  
 *# find the boxes for username and password* username\_field = driver.find\_element\_by\_class\_name(**"js-username-field"**)  
 password\_field = driver.find\_element\_by\_class\_name(**"js-password-field"**)  
  
 *# enter your username:* username\_field.send\_keys(username)  
 driver.implicitly\_wait(1)  
  
 *# enter your password:* password\_field.send\_keys(password)  
 driver.implicitly\_wait(1)  
  
 *# click the "Log In" button:* driver.find\_element\_by\_class\_name(**"EdgeButtom--medium"**).click()  
  
 **return  
  
  
class** wait\_for\_more\_than\_n\_elements\_to\_be\_present(object):  
 **def** \_\_init\_\_(self, locator, count):  
 self.locator = locator  
 self.count = count  
  
 **def** \_\_call\_\_(self, driver):  
 **try**:  
 elements = EC.\_find\_elements(driver, self.locator)  
 **return** len(elements) > self.count  
 **except** StaleElementReferenceException:  
 **return False  
  
  
def** search\_twitter(driver, query):  
 *# wait until the search box has loaded:* box = driver.wait.until(EC.presence\_of\_element\_located((By.NAME, **"q"**)))  
  
 *# find the search box in the html:* driver.find\_element\_by\_name(**"q"**).clear()  
  
 *# enter your search string in the search box:* box.send\_keys(query)  
  
 *# submit the query (like hitting return):* box.submit()  
  
 *# initial wait for the search results to load* wait = WebDriverWait(driver, 10)  
  
 **try**:  
 *# wait until the first search result is found. Search results will be tweets, which are html list items and have the class='data-item-id':* wait.until(EC.visibility\_of\_element\_located((By.CSS\_SELECTOR, **"li[data-item-id]"**)))  
  
 *# scroll down to the last tweet until there are no more tweets:* **while True**:  
  
 *# extract all the tweets:* tweets = driver.find\_elements\_by\_css\_selector(**"li[data-item-id]"**)  
  
 *# find number of visible tweets:* number\_of\_tweets = len(tweets)  
  
 *# keep scrolling:* driver.execute\_script(**"arguments[0].scrollIntoView();"**, tweets[-1])  
  
 **try**:  
 *# wait for more tweets to be visible:* wait.until(wait\_for\_more\_than\_n\_elements\_to\_be\_present(  
 (By.CSS\_SELECTOR, **"li[data-item-id]"**), number\_of\_tweets))  
  
 **except** TimeoutException:  
 *# if no more are visible the "wait.until" call will timeout. Catch the exception and exit the while loop:* **break** *# extract the html for the whole lot:* page\_source = driver.page\_source  
  
 **except** TimeoutException:  
  
 *# if there are no search results then the "wait.until" call in the first "try" statement will never happen and it will time out. So we catch that exception and return no html.* page\_source = **None  
  
 return** page\_source  
  
  
**def** extract\_tweets(page\_source):  
 soup = bs(page\_source, **'lxml'**)  
  
 tweets = []  
 **for** li **in** soup.find\_all(**"li"**, class\_=**'js-stream-item'**):  
  
 *# If our li doesn't have a tweet-id, we skip it as it's not going to be a tweet.* **if 'data-item-id' not in** li.attrs:  
 **continue  
  
 else**:  
 tweet = {  
 **'tweet\_id'**: li[**'data-item-id'**],  
 **'text'**: **None**,  
 **'user\_id'**: **None**,  
 **'user\_screen\_name'**: **None**,  
 **'user\_name'**: **None**,  
 **'created\_at'**: **None**,  
 **'retweets'**: 0,  
 **'likes'**: 0,  
 **'replies'**: 0  
 }  
  
 *# Tweet Text* text\_p = li.find(**"p"**, class\_=**"tweet-text"**)  
 **if** text\_p **is not None**:  
 tweet[**'text'**] = text\_p.get\_text()  
  
 *# Tweet User ID, User Screen Name, User Name* user\_details\_div = li.find(**"div"**, class\_=**"tweet"**)  
 **if** user\_details\_div **is not None**:  
 tweet[**'user\_id'**] = user\_details\_div[**'data-user-id'**]  
 tweet[**'user\_screen\_name'**] = user\_details\_div[**'data-screen-name'**]  
 tweet[**'user\_name'**] = user\_details\_div[**'data-name'**]  
  
 *# Tweet date* date\_span = li.find(**"span"**, class\_=**"\_timestamp"**)  
 **if** date\_span **is not None**:  
 tweet[**'created\_at'**] = float(date\_span[**'data-time-ms'**])  
  
 *# Tweet Retweets* retweet\_span = li.select(**"span.ProfileTweet-action--retweet > span.ProfileTweet-actionCount"**)  
 **if** retweet\_span **is not None and** len(retweet\_span) > 0:  
 tweet[**'retweets'**] = int(retweet\_span[0][**'data-tweet-stat-count'**])  
  
 *# Tweet Likes* like\_span = li.select(**"span.ProfileTweet-action--favorite > span.ProfileTweet-actionCount"**)  
 **if** like\_span **is not None and** len(like\_span) > 0:  
 tweet[**'likes'**] = int(like\_span[0][**'data-tweet-stat-count'**])  
  
 *# Tweet Replies* reply\_span = li.select(**"span.ProfileTweet-action--reply > span.ProfileTweet-actionCount"**)  
 **if** reply\_span **is not None and** len(reply\_span) > 0:  
 tweet[**'replies'**] = int(reply\_span[0][**'data-tweet-stat-count'**])  
  
 tweets.append(tweet)  
  
 **return** tweets  
  
  
driver = init\_driver()  
  
*# log in to twitter (replace username/password with your own):*username = **"don\_rukshan"**password = **" "**login\_twitter(driver, username, password)  
  
*# search twitter:*query = **"covid19 lockdown"**page\_source = search\_twitter(driver, query)  
  
*# extract info from the search results:*tweets = extract\_tweets(page\_source)  
 *# close the driver:*close\_driver(driver)