Wk4_SLP3_Scraping data from a webpage using Selenium and handling output

April 1, 2021

[1]:
```
student_name = "Nikki Fitzherbert"
student_id = "13848336"
```

Step 1

[21]:
```
from selenium import webdriver
import pandas as pd

# create a new instance of Firefox
driver = webdriver.Firefox()

# access Firefox and open Edmunds.com
driver.get("https://forums.edmunds.com/discussion/2864/general/x/
 ↪entry-level-luxury-performance-sedans/")
```

Steps 2-4 Create the scaffold of the class and basic methods, and complete the 'run' and 'extract_data' methods.

[22]:
```
class CarForumCrawler():
    def __init__(self, start_link):
        self.link_to_explore = start_link
        self.comments = pd.DataFrame(columns = ['Date','user_id','comments'])
        self.driver = webdriver.Firefox()
        self.pagecount = 1
        self.next = True

    def run(self):
        while self.next:
            if self.pagecount >=5:
                self.save_data_to_file()
                self.next = False
            try:
                self.driver.get(self.link_to_explore+"p"+str(self.pagecount))
                self.driver.implicitly_wait(15)
                self.extract_data()
                self.pagecount = self.pagecount + 1
            except:
```

```
                print ("Cannot get the page " + self.link_to_explore)
                self.next = False
                raise

    def extract_data(self):
        ids = self.driver.find_elements_by_xpath("//
↪*[contains(@id,'Comment_')]")
        comment_ids = []

        for i in ids:
            comment_ids.append(i.get_attribute('id'))

        for x in comment_ids:
            #Extract dates from for each user on a page
            user_date = self.driver.find_elements_by_xpath('//*[@id="' + x +'"]/
↪div/div[2]/div[2]/span[1]/a/time')[0]
            date = user_date.get_attribute('title')

            #Extract user ids from each user on a page
            userid_element = self.driver.find_elements_by_xpath('//*[@id="' + x␣
↪+'"]/div/div[2]/div[1]/span[1]/a[2]')[0]
            userid = userid_element.text

            #Extract Message for each user on a page
            user_message = self.driver.find_elements_by_xpath('//*[@id="' + x␣
↪+'"]/div/div[3]/div/div[1]')[0]
            comment = user_message.text

            #Adding date, userid and comment for each user in a dataframe
            self.comments.loc[len(self.comments)] = [date,userid,comment]

    def save_data_to_file(self):
    #we save the dataframe content to a CSV file
        self.comments.to_csv ('comments.csv', index = None, header=True)
    def close_spider(self):
    #end the session
        self.driver.quit()
```

Step 5 Run the crawler.

[23]:
```
if __name__ == '__main__':
    url = 'https://forums.edmunds.com/discussion/2864/general/x/
↪entry-level-luxury-performance-sedans/'
    try:
        mycrawler = CarForumCrawler(url)
        mycrawler.run()
        mycrawler.close_spider()
```

```
    except:
        raise
```

[ ]: