Python 腳本操作參考

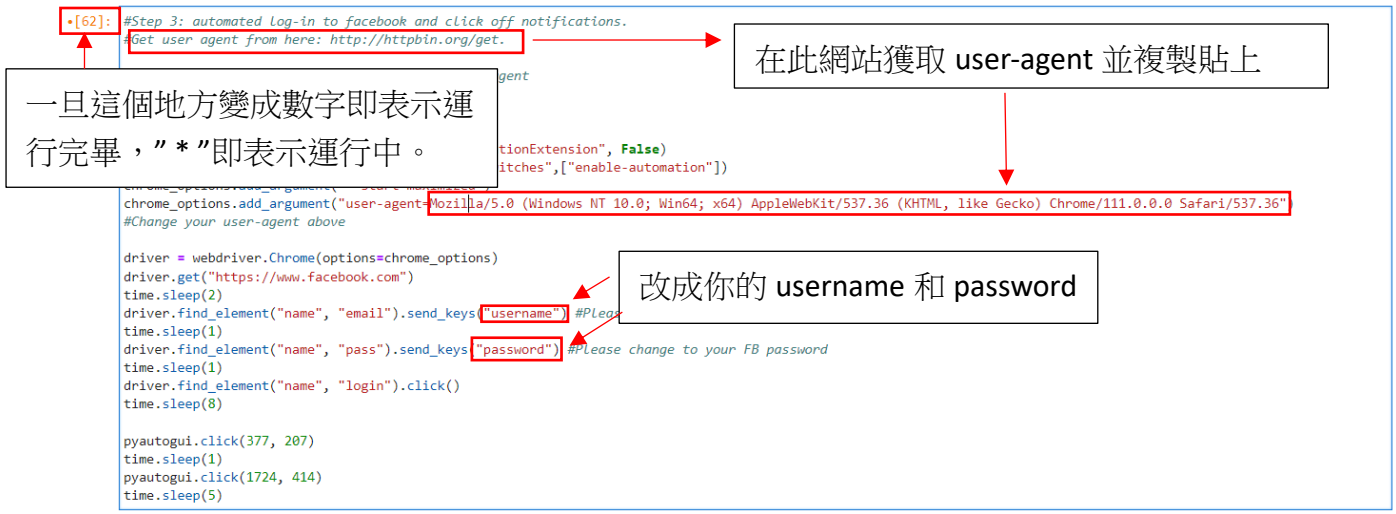
1. 我們將使用 Jupyter notebook 來打開 python 腳本，所以必須先安裝 Anaconda，請下載 64-Bit Graphical Installer (599 MB) (https://www.anaconda.com/products/distribution#Downloads)。
2. 安裝完成後，在電腦裡搜索 Anaconda Prompt 並打開，輸入 jupyter lab 之後按 enter，這會在瀏覽器上開啟 jupyter IDE。

3. 請在 C 槽:/user/username 底下創建一個資料夾，接著把所有爬蟲 ipynb 檔以及透過 Bardeen 生成的 ID excel 檔放入資料夾內，爬蟲生成的 csv 檔到時候也會出現在資料夾裡。E.g., C:\Users\mib67\CAAB420\Facebook_Crawler
4. 接著在 jupyter lab 裡雙擊開啟 ipynb 檔，分別是 scraper_for_oneID.ipynb 和 scraper_for_multipleID.ipynb。
5. 兩個 ipynb 檔的 code 非常相似，差別在於 multipleID 能夠一次把所有 ID 的 po 文擷取下來，oneID 則只能一次擷取一個 ID 的 po 文。
6. 這裡以 scraper_for_multipleID.ipynb 做示範，首先打開腳本。選取第一個 cell 之後使用快捷鍵 Shift+Enter 即可運行，第一個 cell 會 import 很多需要用到的 library。如果遇到報錯就是你的 python 還沒安裝特定 library，可以根據報錯判斷缺少哪個 library，然後新增一個 cell，在 cell 裡打上 pip install libraryName 運行即可。都沒問題即可運行下一個 cell 來讀取 ID excel 檔(必須把 excel 檔放在跟腳本同一資料夾) 。

7. 未來如果有新的 ID，可以修改 code 來讀取新的檔案。

```
•[1]: import pandas as pd
      import numpy as np
      import re, time, requests
      from selenium import webdriver
      from selenium.webdriver.common.by import By
      from bs4 import BeautifulSoup
      pd.set_option("display.max_rows", None, "display.max_columns", None, 'display.max_colwidth', None)

      import pyautogui
      import pyperclip
      import itertools
      #from fake_useragent import UserAgent

[ ]:  #pip install selenium

[59]: #Step 1: use Bardeen to get IDs from Facebook search result
      #Step 2: get poster's ID
      ID_df = pd.read_excel("保養品ID.xlsx") #This is where you read the ID file, you can change the name to yours.
      ID_df["ID"] = ID_df["Link"].str.split("/").str[3]
      ID_df["ID"] = ID_df["ID"].str.split("?").str[0]
      ID_df = ID_df[ID_df.ID !="profile.php"]
      ID_df = ID_df.drop_duplicates(subset=['ID'], keep='first')
      ID_df = ID_df.reset_index(drop=True)

      ID_df
```

8. 這個 cell 能夠自動登入 Facebook，需要輸入你的 username 和 password。此階段會額外開啟一個 chrome 瀏覽器，整個爬蟲過程請不要關掉。直到此段 code 運行完畢為止都不要移動滑鼠，因為 code 會自動使用滑鼠去點擊通知，這個 cell 大概會花 20 秒。這裡需要先填上 username 和 password，並且獲取新的 user-agent(僅需變更一次)。




```
•[62]: #Step 3: automated log-in to facebook and click off notifications.
       #Get user agent from here: http://httpbin.org/get.

       chrome_options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36")
       #Change your user-agent above

       driver = webdriver.Chrome(options=chrome_options)
       driver.get("https://www.facebook.com")
       time.sleep(2)
       driver.find_element("name", "email").send_keys("username") #Plea
       time.sleep(1)
       driver.find_element("name", "pass").send_keys("password") #Please change to your FB password
       time.sleep(1)
       driver.find_element("name", "login").click()
       time.sleep(8)

       pyautogui.click(377, 207)
       time.sleep(1)
       pyautogui.click(1724, 414)
       time.sleep(5)
```

9. 這個 cell 會開始在 chrome 上自行打開每個網紅的粉專，然後自行向下滑動數次加載貼文來獲取每個

貼文的 url。此階段根據設置的參數大概會運行 30 分以上，請維持 chrome 打開並且不要最小化(就是這段時間你無法使用電腦)。測試時我用了 40 分鐘獲取了 1407 個 link，等於此次爬蟲會有 1407 個 post 的內容和留言。

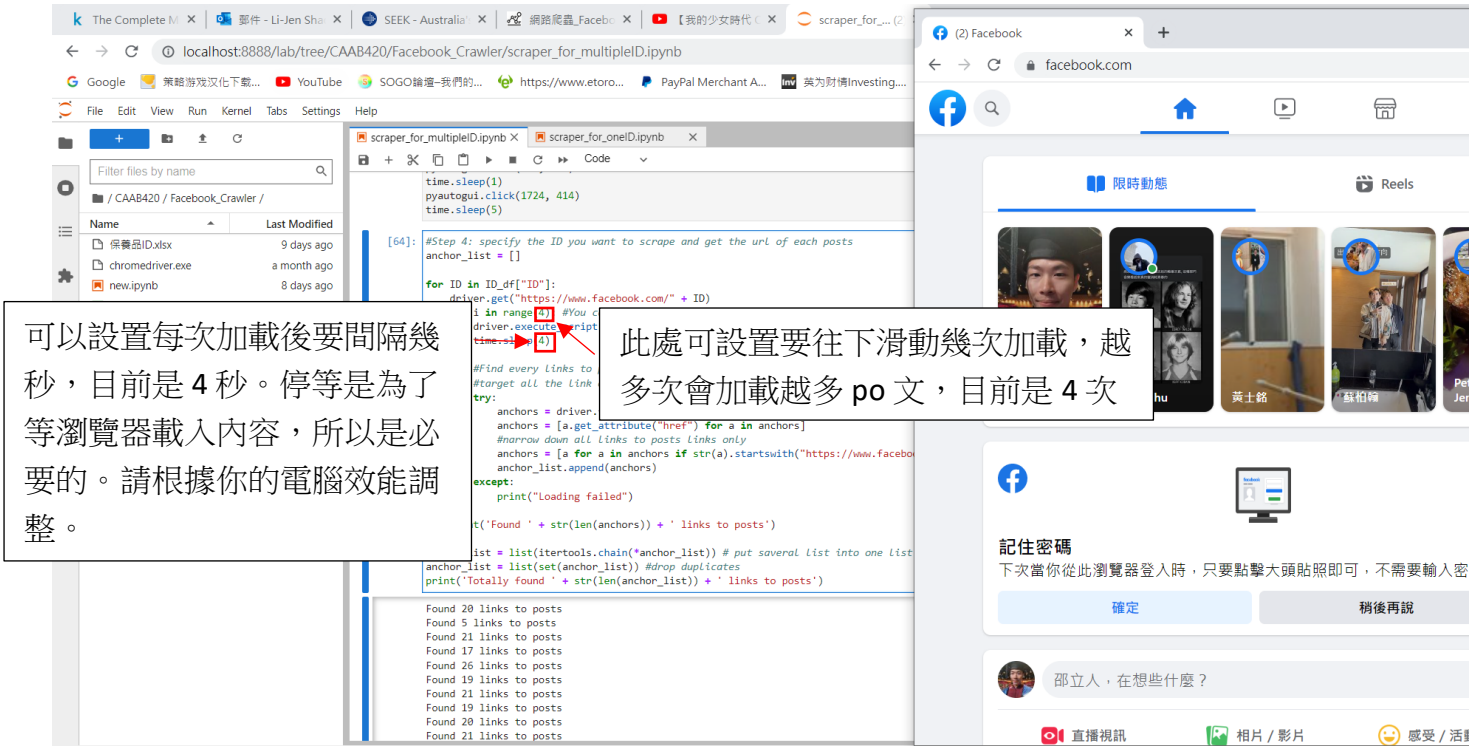


Figure 1: 此階段請將整個螢幕維持在這樣的狀態，print out 的地方會即時更新以擷取多少 link。

10. 最後一個 cell 會使用剛剛取得的 link 重新打開每個 post 並擷取所有內容和留言，過濾掉不含 Link 的 po 文然後存成 csv 檔。此階段會運行比較久，測試時 1407 個 post 花了我 4 小時左右完成運行。這裡可以將 chrome 最小化(可以使用電腦)。Print out 會即時更新擷取情況。

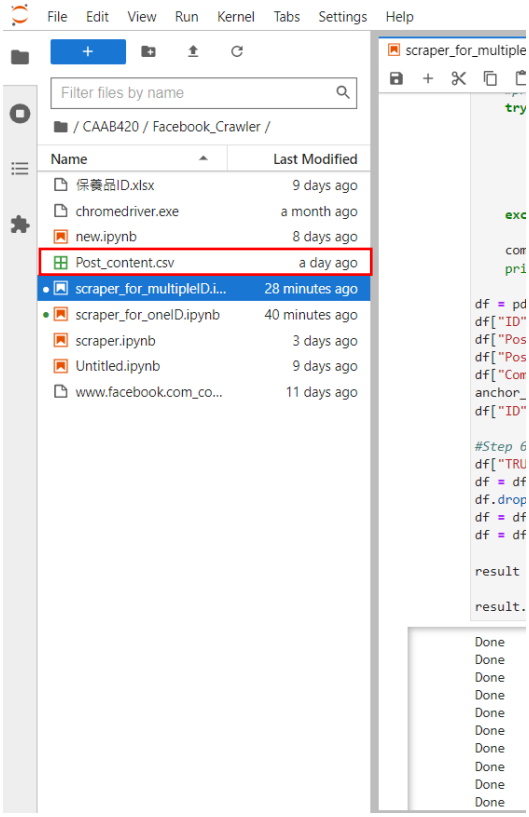
```
[65]: #Step 5: Scrape all the posts and comments which has been loaded and save it as csv file.
post_list = []
comment_list = []

for link in anchor_list:
    driver.get(link)
    time.sleep(4)
    soup = BeautifulSoup(dr
    content = soup.find("di
    comment = soup.find("di

    #process the post content
    try:
        text_post = str(content).replace('href=', 'start;"')
        text_post = str(text_post).replace('style="text-align:start">', 'style="text-align: start;">')
        text_post = str(text_post).split('start;"')
        text_post = [i for i in text_post if str(i).startswith((">", '"'))]
        text_post = [i.split("<")[0] for i in text_post]
        text_post = [i.replace('"', '>') for i in text_post]
        text_post = [i.split(">")[1] for i in text_post]
        text_post = "\n".join(str(i) for i in text_post)
    except:
        print("Failed")
    post_list.append(text_post)

    #process the comment content
    try:
        text_comment = str(comment).split(' ')
        text_comment = [i for i in text_comment if str(i).startswith("start")]
        text_comment = [i.split("<")[0] for i in text_comment]
        text_comment = [i.split('"')[1] for i in text_comment]
        text_comment = "\n".join(str(i) for i in text_comment)
    except:
        print("Failed")
    comment_list.append(text_comment)
    print("Done")

df = pd.DataFrame()
df["ID"] = " "
df["Post link"] = anchor_list
```

此處秒數也是等待瀏覽器加載專頁內容的秒數，請根據電腦效能設置，目前是 4 秒

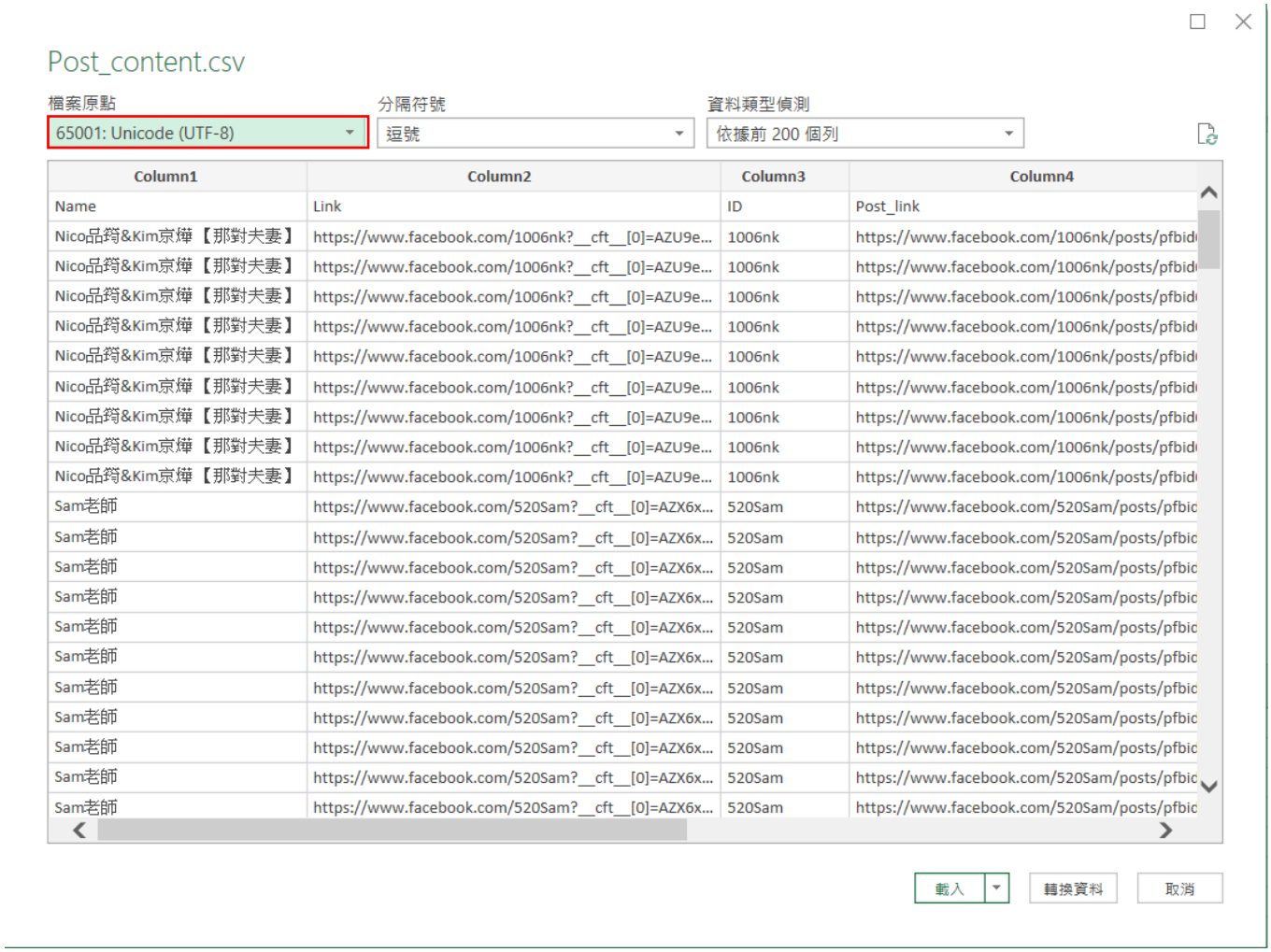
```
df["Post"] = post_list
df["Comment"] = comment_list
anchor_name = [i.split("/")[3] for i in anchor_list]
df["ID"] = [i.split("?")[0] for i in anchor_name]

#Step 6:Filter out the posts which doesn't contain https.
df["TRUE"] = df["Post"].str.contains("https")
df = df[df.TRUE == True]
df.drop("TRUE", inplace=True, axis=1)
df = df.sort_values(by=['ID'])
df = df.reset_index(drop=True)

result = pd.merge(ID_df, df, how='right')

result.to_csv("Post_content.csv", index=False) #Save as csv file
```

```
Done
Done
Done
Done
Done
Done
Done
Done
Done
Done
```

11. 沒有任何問題的話，爬蟲完畢會在左側出現新的 Post_content.csv 檔，單擊右鍵或去本資料夾即可下載。



12. 打開 csv 檔後可能會是亂碼，請去上方資料 > 從文字/csv > 選取 Post_content.csv > 在檔案原點中點選 65001:Unicode(UTF-8) > 載入。測試時總共擷取了 1028 篇關於保養品的 po 文，來自 88 個粉專。



檔案原點: 65001: Unicode (UTF-8)

分隔符號: 逗號

資料類型偵測: 依據前 200 個列

Column1	Column2	Column3	Column4
Name	Link	ID	Post_link
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Nico品荺&Kim京燁【那對夫妻】	https://www.facebook.com/1006nk?__cft__[0]=AZU9e...	1006nk	https://www.facebook.com/1006nk/posts/pfbid
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi
Sam老師	https://www.facebook.com/520Sam?__cft__[0]=AZX6x...	520Sam	https://www.facebook.com/520Sam/posts/pfbi

13. *Voilà.*