一、專題摘要 (解釋實作與說明需要解決的問題，限 300~500 字。)

1. 期末專題主題

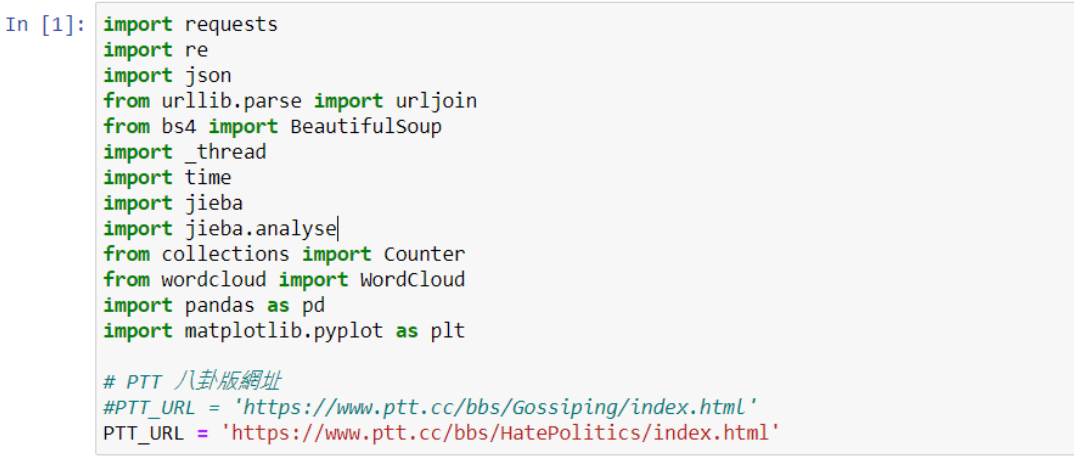
- 爬取 PTT 政黑板約前 500 篇文章及每篇文章推文，並做簡單整理

2. 期末專題基本目標

- 爬取 PTT 政黑板文章
- 整理全部的文章，了解最近趨勢、熱門關鍵字
- 整理發/推文者其全部的發/推文，進一步用 jieba 抓取關鍵字
- 整理有使用相同 ip 的推/發文者，猜測推/發文者是否有關聯

二、實作方法介紹 (介紹使用的程式碼、模組，並附上實作過程與結果的截圖，需圖文並茂。)

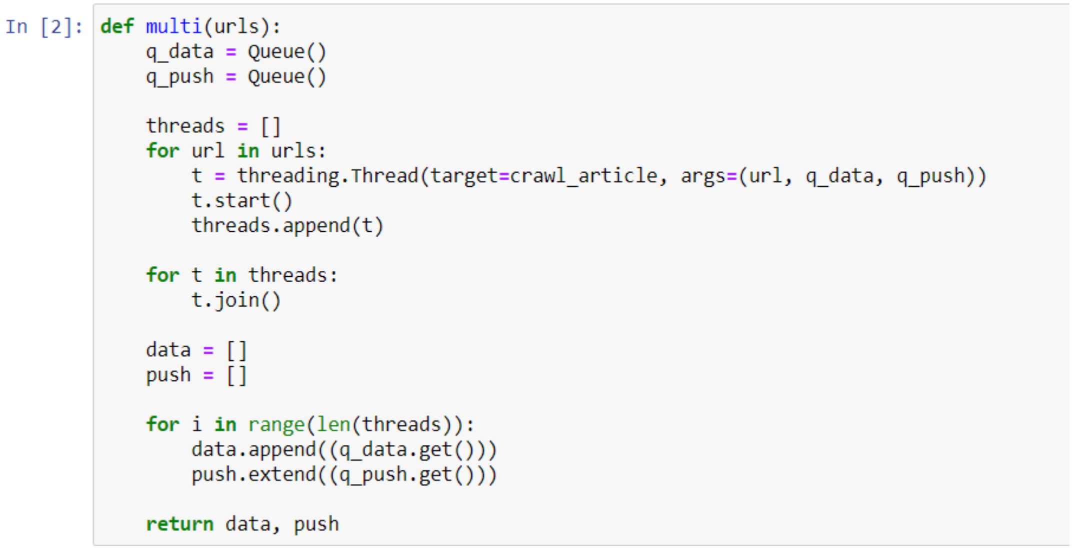
1. 使用的模組介紹



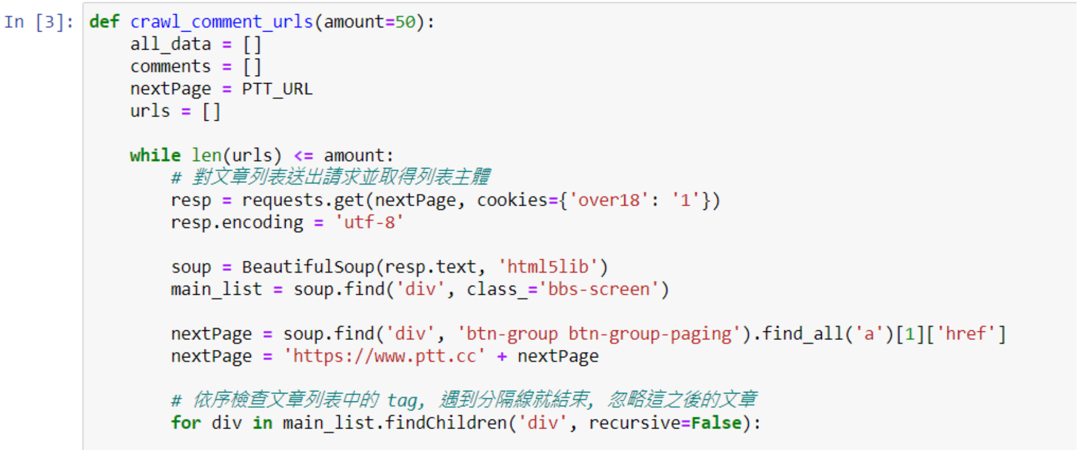
```
In [1]: import requests
        import re
        import json
        from urllib.parse import urljoin
        from bs4 import BeautifulSoup
        import _thread
        import time
        import jieba
        import jieba.analyse
        from collections import Counter
        from wordcloud import WordCloud
        import pandas as pd
        import matplotlib.pyplot as plt

        # PTT 八卦版網址
        #PTT_URL = 'https://www.ptt.cc/bbs/Gossiping/index.html'
        PTT_URL = 'https://www.ptt.cc/bbs/HatePolitics/index.html'
```

2. 介紹使用的程式碼(完整程式碼請參閱最後 github 連結)

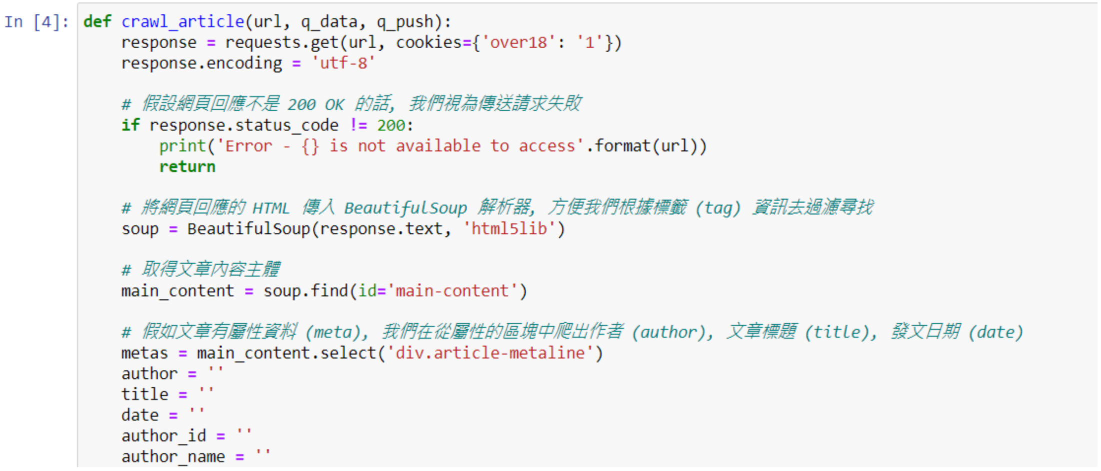
- 定義爬 PTT
- 使用 Crawl_article 以及 crawl_comment
- 並且使用 MultiThread 加速



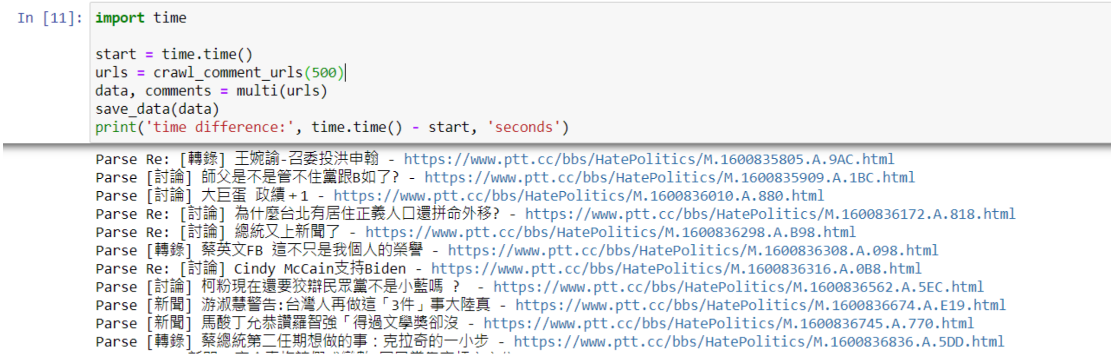
```
In [2]: def multi(urls):
            q_data = Queue()
            q_push = Queue()

            threads = []
            for url in urls:
                t = threading.Thread(target=crawl_article, args=(url, q_data, q_push))
                t.start()
                threads.append(t)

            for t in threads:
                t.join()

            data = []
            push = []

            for i in range(len(threads)):
                data.append((q_data.get()))
                push.extend((q_push.get()))

            return data, push
```



```
In [3]: def crawl_comment_urls(amount=50):
            all_data = []
            comments = []
            nextPage = PTT_URL
            urls = []

            while len(urls) <= amount:
                # 對文章列表送出請求並取得列表主體
                resp = requests.get(nextPage, cookies={'over18': '1'})
                resp.encoding = 'utf-8'

                soup = BeautifulSoup(resp.text, 'html5lib')
                main_list = soup.find('div', class_='bbs-screen')

                nextPage = soup.find('div', 'btn-group btn-group-paging').find_all('a')[1]['href']
                nextPage = 'https://www.ptt.cc' + nextPage

                # 依序檢查文章列表中的 tag, 遇到分隔線就結束, 忽略這之後的文章
                for div in main_list.findChildren('div', recursive=False):
```

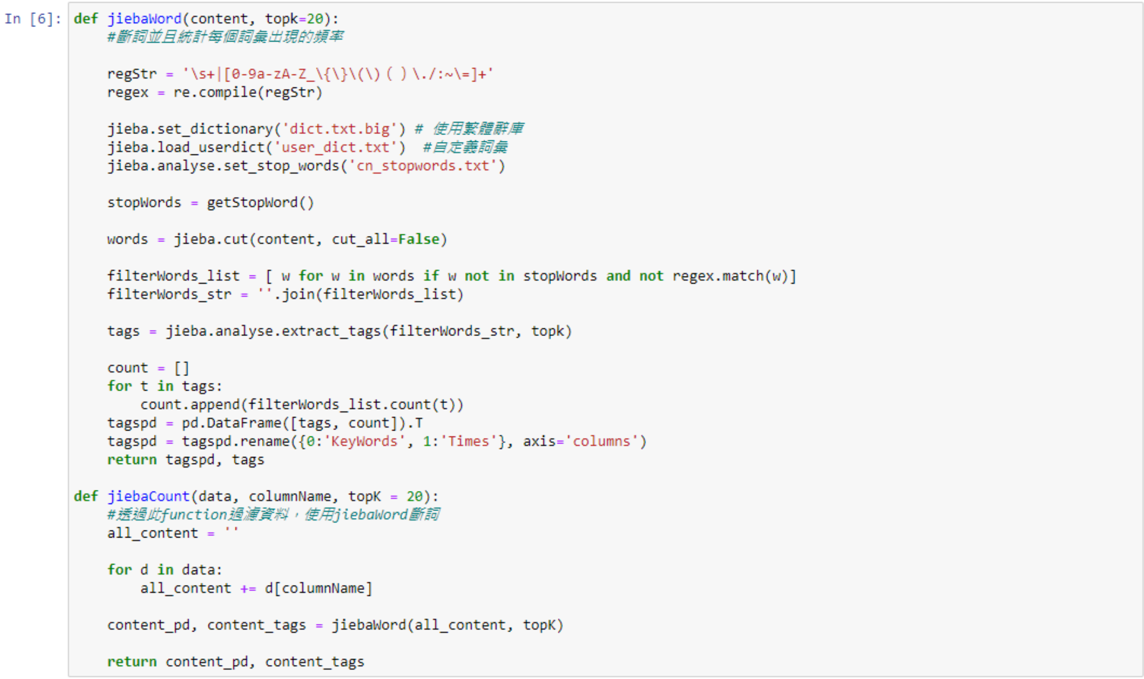


```
In [4]: def crawl_article(url, q_data, q_push):
            response = requests.get(url, cookies={'over18': '1'})
            response.encoding = 'utf-8'

            # 假設網頁回應不是 200 OK 的話, 我們視為傳送請求失敗
            if response.status_code != 200:
                print('Error - {} is not available to access'.format(url))
                return

            # 將網頁回應的 HTML 傳入 BeautifulSoup 解析器, 方便我們根據標籤 (tag) 資訊去過濾尋找
            soup = BeautifulSoup(response.text, 'html5lib')

            # 取得文章內容主體
            main_content = soup.find(id='main-content')

            # 假如文章有屬性資料 (meta), 我們在從屬性的區塊中爬出作者 (author), 文章標題 (title), 發文日期 (date)
            metas = main_content.select('div.article-metaline')
            author = ''
            title = ''
            date = ''
            author_id = ''
            author_name = ''
```

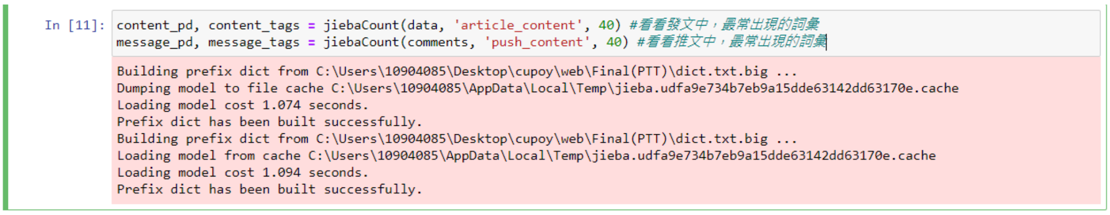


```
In [11]: import time

         start = time.time()
         urls = crawl_comment_urls(500)
         data, comments = multi(urls)
         save_data(data)
         print('time difference:', time.time() - start, 'seconds')

         Parse Re: [轉錄] 王婉諭-召委投洪申翰 - https://www.ptt.cc/bbs/HatePolitics/M.1600835805.A.9AC.html
         Parse [討論] 師父是不是管不住黨跟B如了? - https://www.ptt.cc/bbs/HatePolitics/M.1600835909.A.1BC.html
         Parse [討論] 大巨蛋 政績+1 - https://www.ptt.cc/bbs/HatePolitics/M.1600836010.A.880.html
         Parse Re: [討論] 為什麼台北有居住正義人口還拼命外移? - https://www.ptt.cc/bbs/HatePolitics/M.1600836172.A.818.html
         Parse Re: [討論] 總統又上新聞了 - https://www.ptt.cc/bbs/HatePolitics/M.1600836298.A.B98.html
         Parse [轉錄] 蔡英文FB 這不只是我個人的榮譽 - https://www.ptt.cc/bbs/HatePolitics/M.1600836308.A.098.html
         Parse Re: [討論] Cindy McCain支持Biden - https://www.ptt.cc/bbs/HatePolitics/M.1600836316.A.0B8.html
         Parse [討論] 柯粉現在還要狡辯民眾黨不是小藍嗎 ？  - https://www.ptt.cc/bbs/HatePolitics/M.1600836562.A.5EC.html
         Parse [新聞] 游淑慧警告:台灣人再做這「3件」事大陸真 - https://www.ptt.cc/bbs/HatePolitics/M.1600836674.A.E19.html
         Parse [新聞] 馬酸丁允恭讚羅智強「得過文學獎卻沒 - https://www.ptt.cc/bbs/HatePolitics/M.1600836745.A.770.html
         Parse [轉錄] 蔡總統第二任期想做的事：克拉奇的一小步 - https://www.ptt.cc/bbs/HatePolitics/M.1600836836.A.5DD.html
```

運用 jiebraWord 與 jiebaCount，讀取推文與發文中，最常出現的詞彙

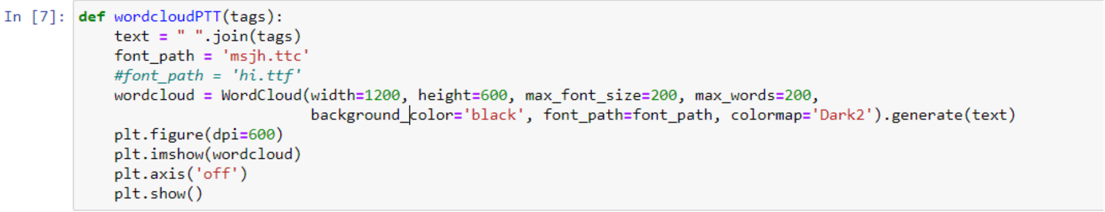


```
In [6]: def jiebaWord(content, topk=20):
            #斷詞並且統計每個詞彙出現的頻率

            regStr = '\s+|[0-9a-zA-Z_\{\}\(\)（）\./:~\=]+'
            regex = re.compile(regStr)

            jieba.set_dictionary('dict.txt.big') # 使用繁體辭庫
            jieba.load_userdict('user_dict.txt')  #自定義詞彙
            jieba.analyse.set_stop_words('cn_stopwords.txt')

            stopWords = getStopWord()

            words = jieba.cut(content, cut_all=False)

            filterWords_list = [ w for w in words if w not in stopWords and not regex.match(w)]
            filterWords_str = ''.join(filterWords_list)

            tags = jieba.analyse.extract_tags(filterWords_str, topk)

            count = []
            for t in tags:
                count.append(filterWords_list.count(t))
            tagspd = pd.DataFrame([tags, count]).T
            tagspd = tagspd.rename({0:'KeyWords', 1:'Times'}, axis='columns')
            return tagspd, tags

        def jiebaCount(data, columnName, topK = 20):
            #透過此function過濾資料，使用jiebaWord斷詞
            all_content = ''

            for d in data:
                all_content += d[columnName]

            content_pd, content_tags = jiebaWord(all_content, topK)

            return content_pd, content_tags
```



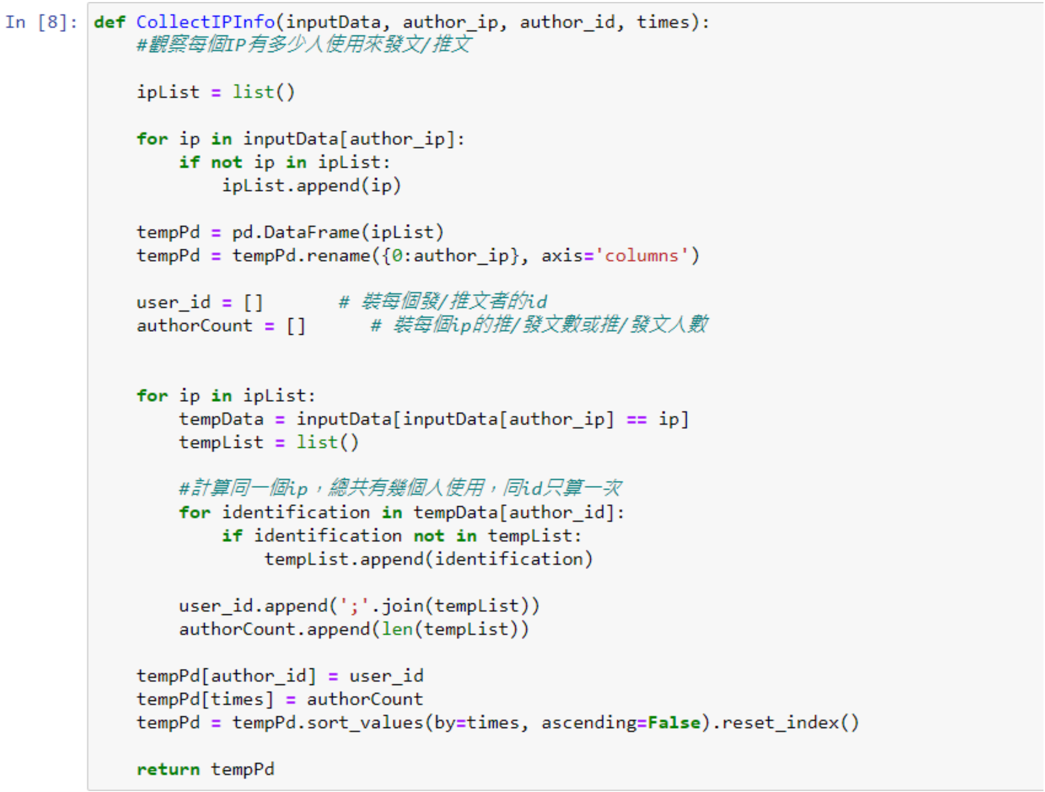
```
In [11]: content_pd, content_tags = jiebaCount(data, 'article_content', 40) #看看發文中，最常出現的詞彙
         message_pd, message_tags = jiebaCount(comments, 'push_content', 40) #看看推文中，最常出現的詞彙

         Building prefix dict from C:\Users\10904085\Desktop\cupoy\web\Final(PTT)\dict.txt.big ...
         Dumping model to file cache C:\Users\10904085\AppData\Local\Temp\jieba.udfa9e734b7eb9a15dde63142dd63170e.cache
         Loading model cost 1.074 seconds.
         Prefix dict has been built successfully.
         Building prefix dict from C:\Users\10904085\Desktop\cupoy\web\Final(PTT)\dict.txt.big ...
         Loading model from cache C:\Users\10904085\AppData\Local\Temp\jieba.udfa9e734b7eb9a15dde63142dd63170e.cache
         Loading model cost 1.094 seconds.
         Prefix dict has been built successfully.
```

利用 wordcouldPTT 取得貼文與推文的文字雲，了解最常出現的 40 個詞彙

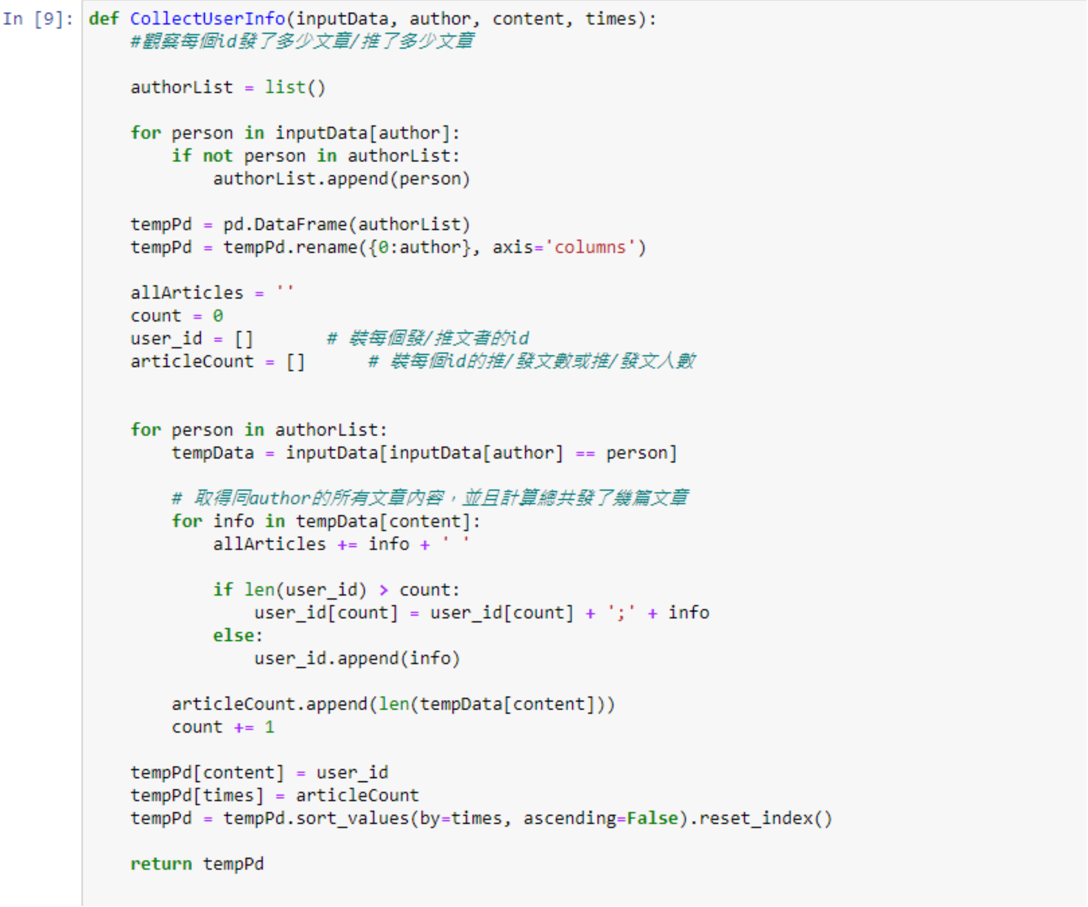


```
In [7]: def wordcloudPTT(tags):
            text = " ".join(tags)
            font_path = 'msjh.ttc'
            #font_path = 'hi.ttf'
            wordcloud = WordCloud(width=1200, height=600, max_font_size=200, max_words=200,
                                  background_color='black', font_path=font_path, colormap='Dark2').generate(text)
            plt.figure(dpi=600)
            plt.imshow(wordcloud)
            plt.axis('off')
            plt.show()
```

利用 CollectUserInfo 以及 CollectIPinfo
整理最常發文的 ip, id，或是最常推文的 id



```
In [8]: def CollectIPInfo(inputData, author_ip, author_id, times):
            #觀察每個IP有多少人使用來發文/推文

            ipList = list()

            for ip in inputData[author_ip]:
                if not ip in ipList:
                    ipList.append(ip)

            tempPd = pd.DataFrame(ipList)
            tempPd = tempPd.rename({0:author_ip}, axis='columns')

            user_id = []       # 裝每個發/推文者的id
            authorCount = []      # 裝每個ip的推/發文數或推/發文人數


            for ip in ipList:
                tempData = inputData[inputData[author_ip] == ip]
                tempList = list()

                #計算同一個ip，總共有幾個人使用，同id只算一次
                for identification in tempData[author_id]:
                    if identification not in tempList:
                        tempList.append(identification)

                user_id.append(';'.join(tempList))
                authorCount.append(len(tempList))

            tempPd[author_id] = user_id
            tempPd[times] = authorCount
            tempPd = tempPd.sort_values(by=times, ascending=False).reset_index()

            return tempPd
```



```
In [9]: def CollectUserInfo(inputData, author, content, times):
            #觀察每個id發了多少文章/推了多少文章

            authorList = list()

            for person in inputData[author]:
                if not person in authorList:
                    authorList.append(person)

            tempPd = pd.DataFrame(authorList)
            tempPd = tempPd.rename({0:author}, axis='columns')

            allArticles = ''
            count = 0
            user_id = []        # 裝每個發/推文者的id
            articleCount = []      # 裝每個id的推/發文數或推/發文人數


            for person in authorList:
                tempData = inputData[inputData[author] == person]

                # 取得同author的所有文章內容，並且計算總共發了幾篇文章
                for info in tempData[content]:
                    allArticles += info + ' '

                    if len(user_id) > count:
                        user_id[count] = user_id[count] + ';' + info
                    else:
                        user_id.append(info)

                articleCount.append(len(tempData[content]))
                count += 1

            tempPd[content] = user_id
            tempPd[times] = articleCount
            tempPd = tempPd.sort_values(by=times, ascending=False).reset_index()

            return tempPd
```

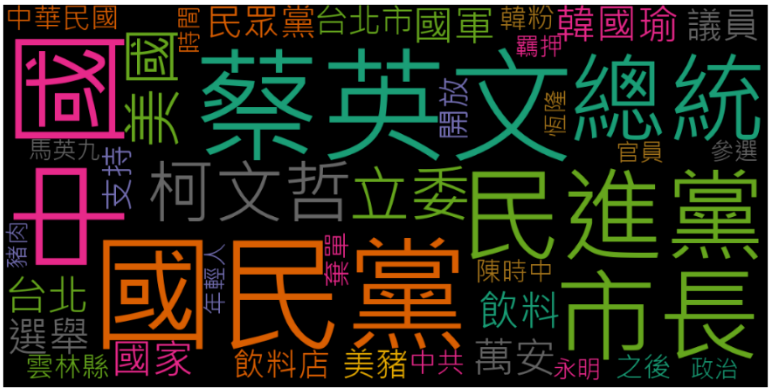
三、成果展示 (介紹成果的特點為何，並撰寫心得。)

1. 特點

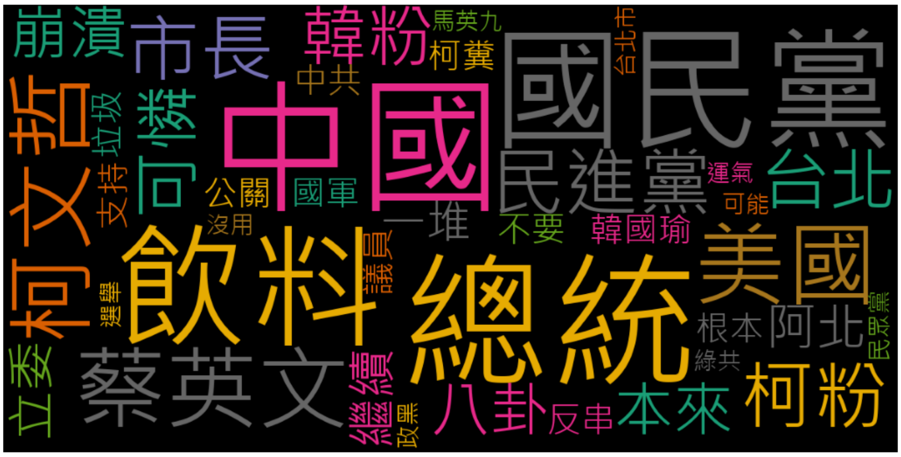
- 利用 pandas 整理每個推/發文者的全部推/發文及每個 ip 對應的使用

2. 成果

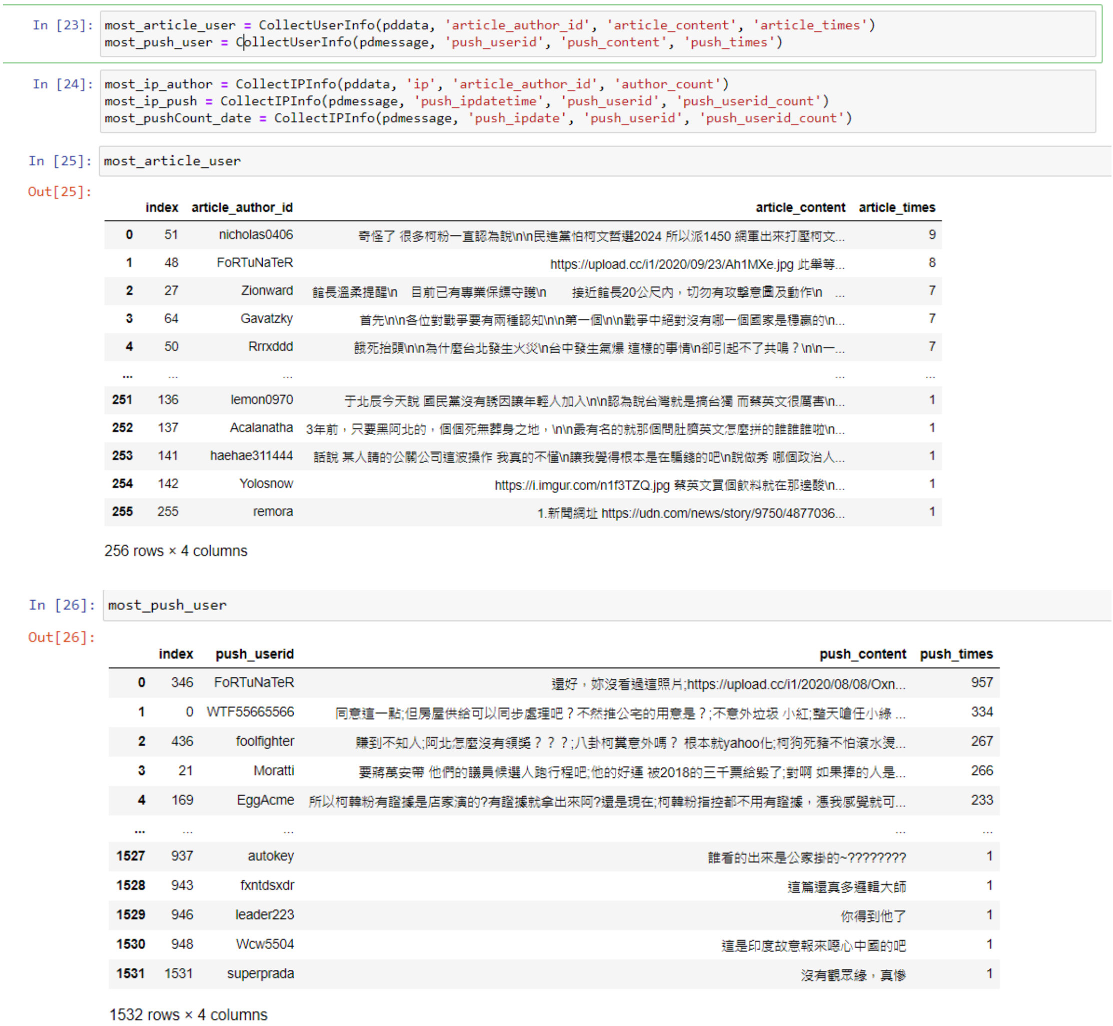
發文的文字雲



推文的文字雲



下列則為發文最多，推文最多，發文最多 id，推文最多 id，以及發文最多日期的詳細表格資料



```
In [23]: most_article_user = CollectUserInfo(pddata, 'article_author_id', 'article_content', 'article_times')
         most_push_user = CollectUserInfo(pdmessage, 'push_userid', 'push_content', 'push_times')

In [24]: most_ip_author = CollectIPInfo(pddata, 'ip', 'article_author_id', 'author_count')
         most_ip_push = CollectIPInfo(pdmessage, 'push_ipdatetime', 'push_userid', 'push_userid_count')
         most_pushCount_date = CollectIPInfo(pdmessage, 'push_ipdate', 'push_userid', 'push_userid_count')

In [25]: most_article_user
Out[25]:
```

	index	article_author_id	article_content	article_times
0	51	nicholas0406	奇怪了 很多柯粉一直認為說\n\n民進黨怕柯文哲選2024 所以派1450 網軍出來打壓柯文...	9
1	48	FoRTuNaTeR	https://upload.cc/i1/2020/09/23/Ah1MXe.jpg 此舉等...	8
2	27	Zionward	館長溫柔提醒\n 目前已有專業保鑣守護\n 接近館長20公尺內，切勿有攻擊意圖及動作\n ...	7
3	64	Gavatzky	首先\n\n各位對戰爭要有兩種認知\n\n第一個\n\n戰爭中絕對沒有哪一個國家是穩贏的\n...	7
4	50	Rrrxddd	餓死抬頭\n\n為什麼台北發生火災\n台中發生氣爆 這樣的事情\n卻引起不了共鳴？\n\n一...	7
...	...	...	...	...
251	136	lemon0970	于北辰今天說 國民黨沒有誘因讓年輕人加入\n\n認為說台灣就是搞台獨 而蔡英文很厲害\n...	1
252	137	Acalanatha	3年前，只要黑阿北的，個個死無葬身之地，\n\n最有名的就那個問肚臍英文怎麼拼的誰誰誰啦\n...	1
253	141	haehae311444	話說 某人請的公關公司這波操作 我真的不懂\n讓我覺得根本是在騙錢的吧\n說做秀 哪個政治人...	1
254	142	Yolosnow	https://i.imgur.com/n1f3TZQ.jpg 蔡英文買個飲料就在那邊酸\n...	1
255	255	remora	1.新聞網址 https://udn.com/news/story/9750/4877036...	1

256 rows × 4 columns

```
In [26]: most_push_user
Out[26]:
```

	index	push_userid	push_content	push_times
0	346	FoRTuNaTeR	還好，妳沒看過這照片;https://upload.cc/i1/2020/08/08/Oxn...	957
1	0	WTF55665566	同意這一點;但房屋供給可以同步處理吧？不然推公宅的用意是？;不意外垃圾 小紅;整天嗆任小綠 ...	334
2	436	foolfighter	賺到不知人;阿北怎麼沒有領獎？？？;八卦柯糞意外嗎？ 根本就yahoo化;柯狗死豬不怕滾水燙...	267
3	21	Moratti	要蔣萬安帶 他們的議員候選人跑行程吧;他的好運 被2018的三千票給毀了;對啊 如果捧的人是...	266
4	169	EggAcme	所以柯韓粉有證據是店家演的?有證據就拿出來阿?還是現在;柯韓粉指控都不用有證據，憑我感覺就可...	233
...	...	...	...	...
1527	937	autokey	誰看的出來是公家掛的~???????	1
1528	943	fxntdsxdr	這篇還真多邏輯大師	1
1529	946	leader223	你得到他了	1
1530	948	Wcw5504	這是印度故意報來噁心中國的吧	1
1531	1531	superprada	沒有觀眾緣，真慘	1

1532 rows × 4 columns

In [27]:
```
most_ip_author
```

Out[27]:

	index	ip	article_author_id	author_count
0	0	223.140.254.110	hagousla	1
1	209	59.120.195.222	tigerzz3	1
2	216	111.240.123.4	RX00	1
3	215	114.27.114.48	pqbd22	1
4	214	101.12.20.105	Saint0822	1
...	...	...	...	...
312	105	1.160.82.71	XindeX	1
313	104	101.14.193.73	a520	1
314	103	49.214.180.77	andycat5566	1
315	102	118.150.253.173	goetze	1
316	316	101.137.38.123	CavendishJr	1

In [28]:
```
most_ip_push
```

Out[28]:

	index	push_ipdatetime	push_userid	push_userid_count
0	1127	09/22 13:23	Moratti;DarthCod;obey1110;Atkins13;boogieman;d...	21
1	1111	09/22 13:20	windom;WTF55665566;takuminauki;Atkins13;Moratt...	21
2	1300	09/22 10:58	gunng;ianbh;castalchen;tigerzz3;FoRTuNaTeR;Cav...	19
3	1376	09/22 10:28	lostman;Tiara5566;dakkk;yun0112;pbkfss;foolfig...	19
4	1110	09/22 13:19	windom;takuminauki;chenyei;zeuswell;boogieman;...	18
...	...	...	...	...
2013	1616	09/22 04:25	ota978	1
2014	1615	09/22 07:01	DameKyon	1
2015	1614	09/22 05:15	FoRTuNaTeR	1
2016	1613	09/22 04:10	FoRTuNaTeR	1
2017	2017	09/21 21:05	CavendishJr	1

2018 rows × 4 columns

In [29]:
```
most_pushCount_date
```

Out[29]:

	index	push_ipdate	push_userid	push_userid_count
0	2	09/22	platinum500a;pinacolada;Atkins13;virginia779;C...	1268
1	0	09/23	WTF55665566;latin0126;kairi5217;a2550099;uieas...	599
2	4	09/21	bruce2248;genheit;Moratti;MrJohn;wx190;elainak...	251
3	1	07/22	ubcs	1
4	3	192.192.154.43	luluhihi	1

四、結論 (總結本次專題的問題與結果)

- 不一定每個推文都會顯示其 ip
- 利用 thread 加速，成功縮短爬取時間，但須考慮網站的防衛機制，如果爬取速度過快可能會造成反效果
- 希望能做到分類每個字詞的傾向，不用去觀察發/推文者的全部關鍵字，就能了解每個發/推文者的傾向。
- 或是有關鍵詞庫網，不用自己定義太多文字

五、期末專題作者資訊 (請附上作者資訊)

1. 個人 Github 連結：https://github.com/garycjwu/1st-DL-CVMarathon/tree/master/Web_Crawler
2. 個人在百日馬拉松顯示名稱：Gary Wu