電影推薦聊天機器人

資管碩一 611630020 尹姵雯

資管碩一 611630046 廖蔓瑄

資管碩一 611630384 鄭　榆

資管碩一 611630400 簡偉丞

目錄 / Content

01 製作動機

02 使用流程及工具

03 實作過程

04 成果展現

01

製作動機

面臨問題

- 評分與評論內容不一致
- 排行榜只將評分納入計算
- 難以瞭解電影真實的推薦指數

解決方法

爬蟲

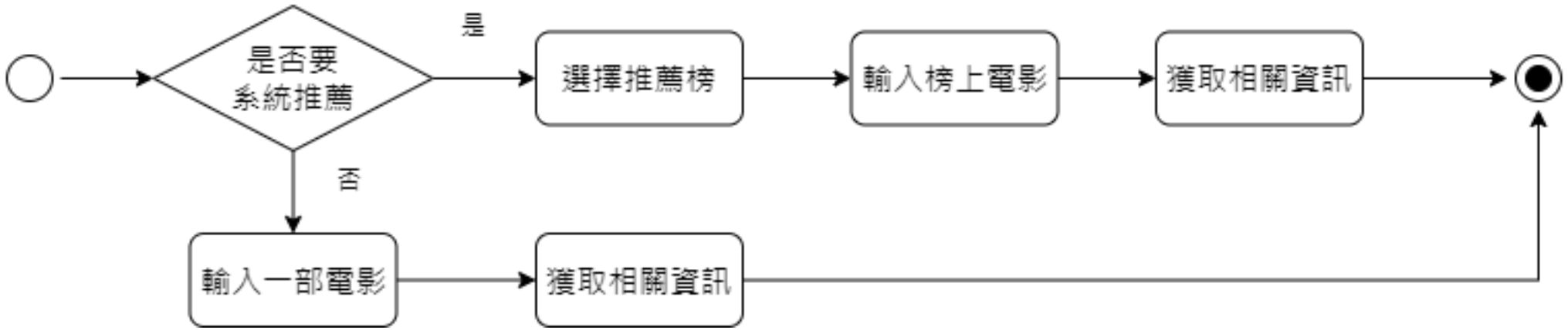
情緒分析

LINE Bot

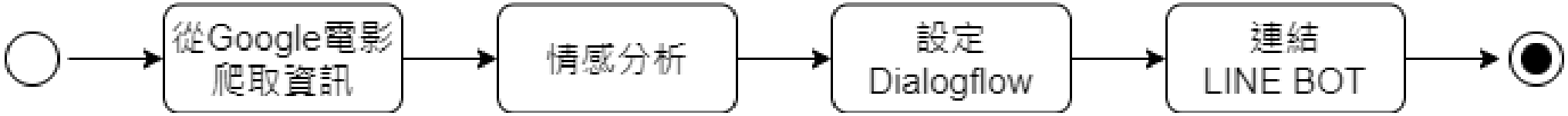
部屬到LINE

02 使用流程及工具

使用流程



系統設計流程

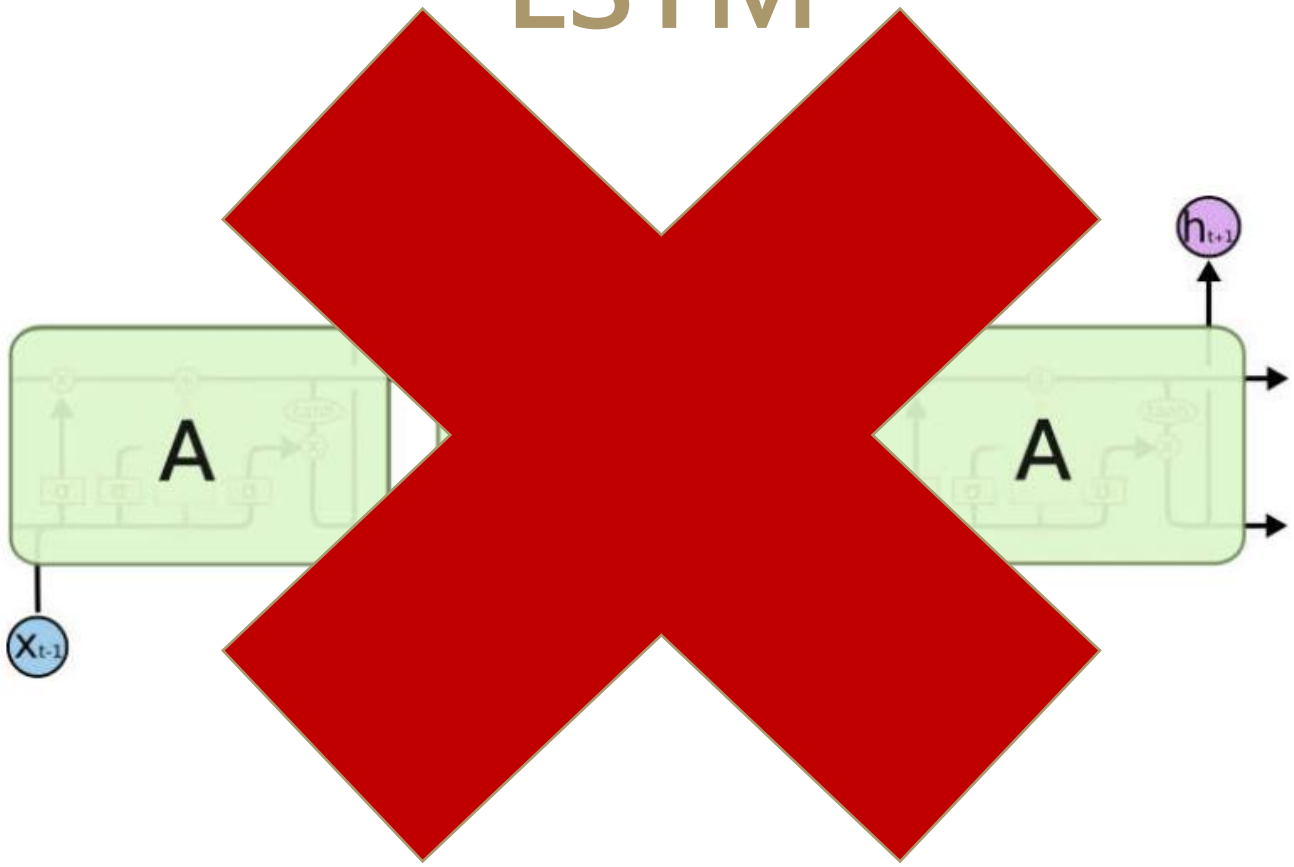


LSTM模型訓練

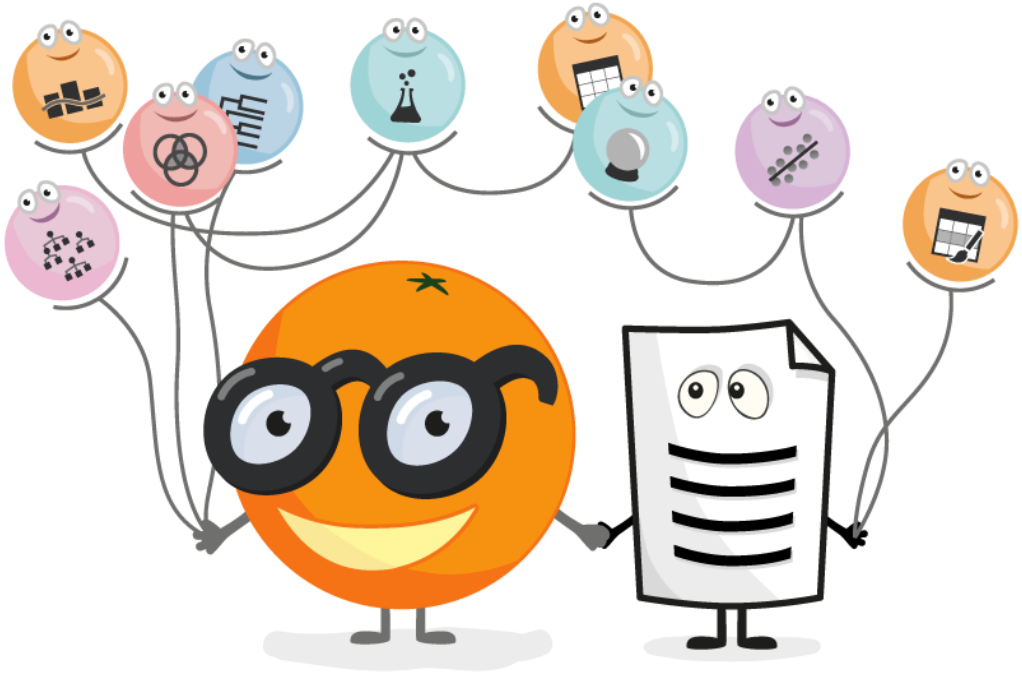
IMDB Review Dataset → Tokenizer → 訓練LSTM模型

(分詞，轉數字化)

LSTM



Orange



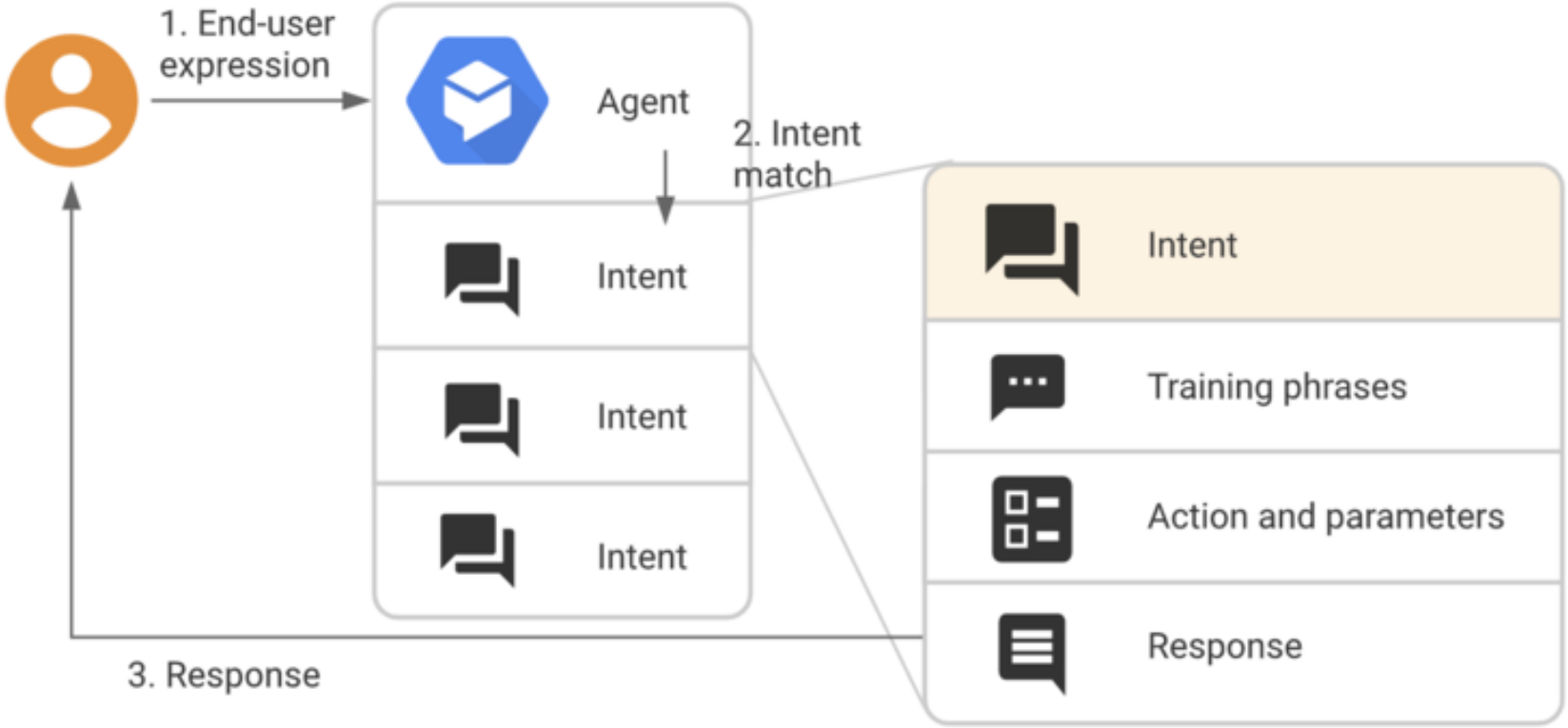
可視化程式設計工具，用於數據可視化與機器學習和建模預測

取得電影評分

爬蟲電影 → Orange → 正向評論指數(0~1)
負向評論指數(0~1)

比較兩者，判斷該評論視為正評或負評 → 與星等(0~5)百分比取平均

DialogFlow


1. End-user expression
Agent
2. Intent match
Intent
Intent
Intent
3. Response
Intent
Training phrases
Action and parameters
Response

03

實作過程

載入IMDB資料集

```
[ ] import warnings
    warnings.filterwarnings('ignore')
    sns.set()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
[ ] imdb = pd.read_csv("/content/drive/MyDrive/ML/IMDB Dataset.csv")
    imdb.head()
```

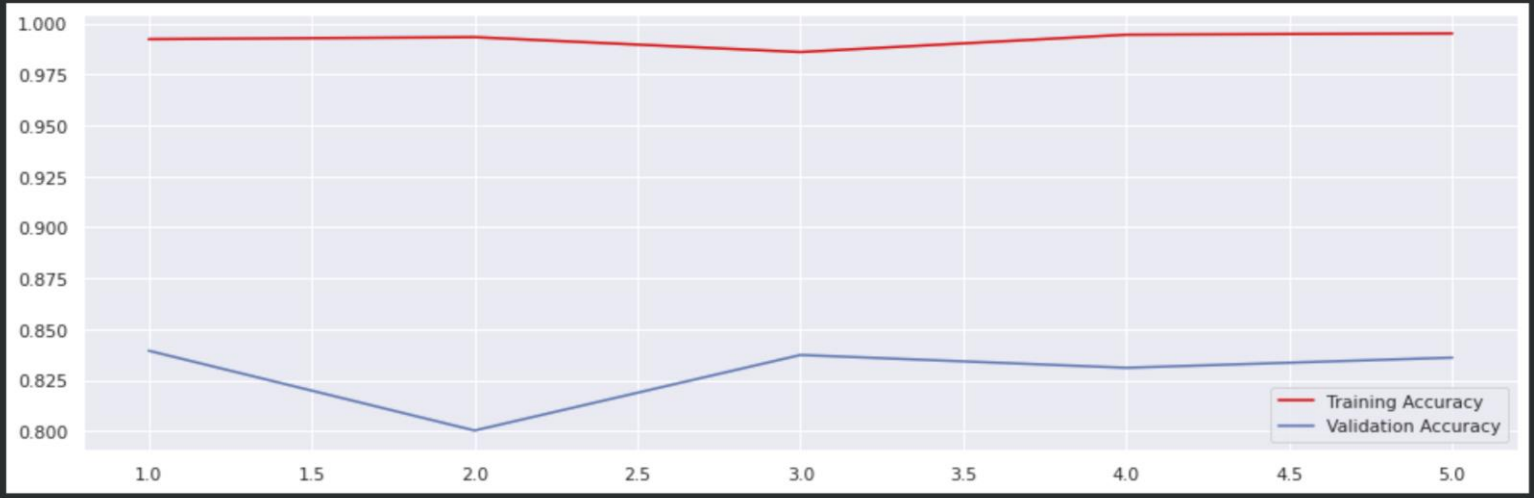
	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

```
[ ] imdb.sentiment.value_counts()
```

```
positive    25000
negative    25000
Name: sentiment, dtype: int64
```

訓練後模型準確率

```
[ ] plt.figure(figsize=(16,5))
    epochs = range(1, len(history.history['accuracy'])+1)
    plt.plot(epochs, history.history['accuracy'], 'b', label='Training Accuracy', color='red')
    plt.plot(epochs, history.history['val_accuracy'], 'b', label='Validation Accuracy')
    plt.legend()
    plt.show()
```




```
[ ] model.save('imdb_lstm.h5')
```

```
[ ] from keras.models import load_model
    lstm_model = load_model('imdb_lstm.h5')
```

查看預測結果

```
[ ]  #查看預測結果
     SentimentDict={1:'正面的',  0:'負面的'}
```

```
[ ]  from  collections.abc  import  Sequence
     #寫一個將文字轉為向量的define

     def  predict_review(input_text):
         input_seq  =  tokenizer.texts_to_sequences([input_text])
         pad_input_seq  =  pad_sequences(input_seq,  maxlen=128)
         #predict_classes  func  remove  at  v2.6
         #predict_result=model.predict_classes(pad_input_seq)
         predict_result  =  np.argmax(model.predict(pad_input_seq),  axis=1)
         print(predict_result)
         print(SentimentDict[predict_result[0]])
```

```
predict_review('I  paid  $24.99  for  this  movie  and  could  only  watch  is  once.  Why  is  that?  I  chose  to  buy  it  for  a  reason  not  rent  it.')
```

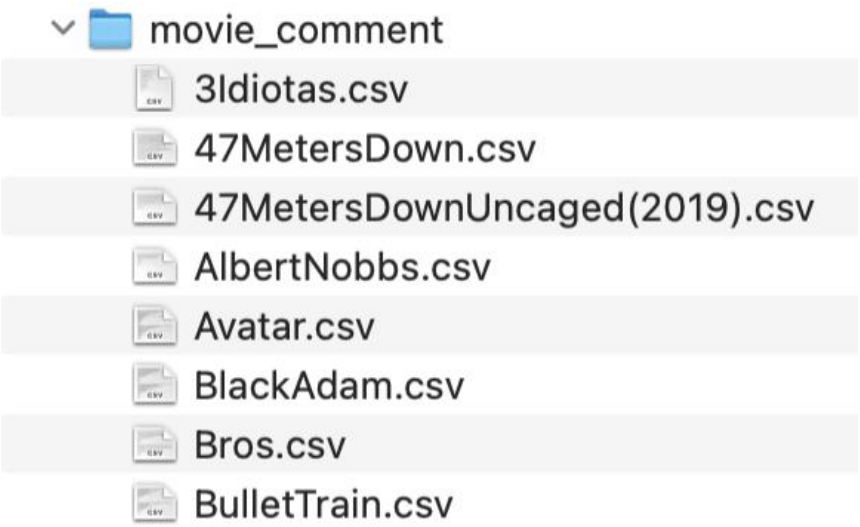
```
1/1 [==============================] - 0s 60ms/step
[0]
負面的
```

Orange

取得所有電影評論

爬蟲取得google movie所有電影，以電影為檔案名稱，

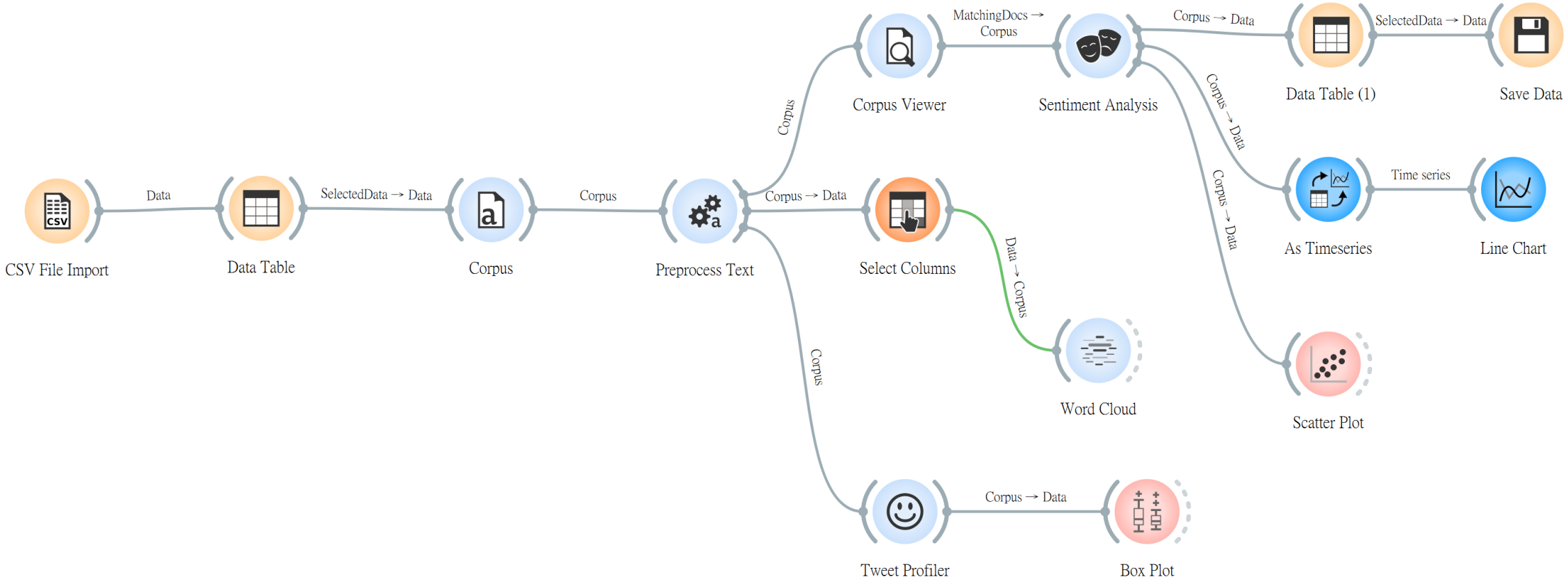
存放該電影所有評論



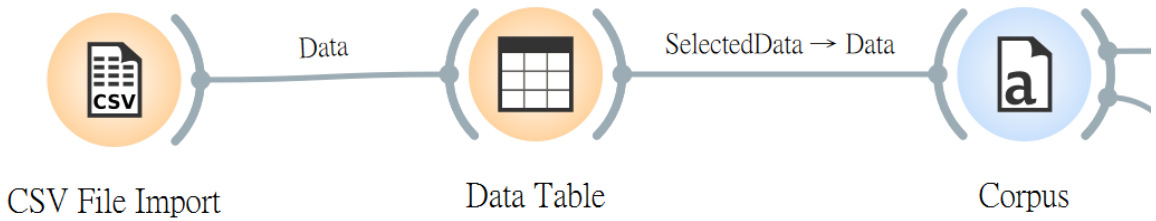
3Idiotas

comment
They didn't do a good enough job of explaining why he left the woman he loved for 5 years. Otherwise, good movie.
Rip off from Bollywood movie of same name 3 Idiots.
It's adopted from the Bollywood movie 3 idiots...

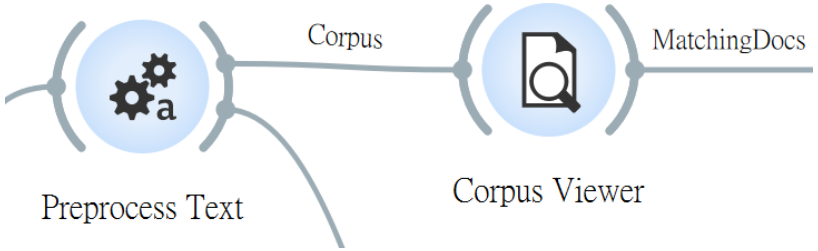
Orange Sentiment Analysis

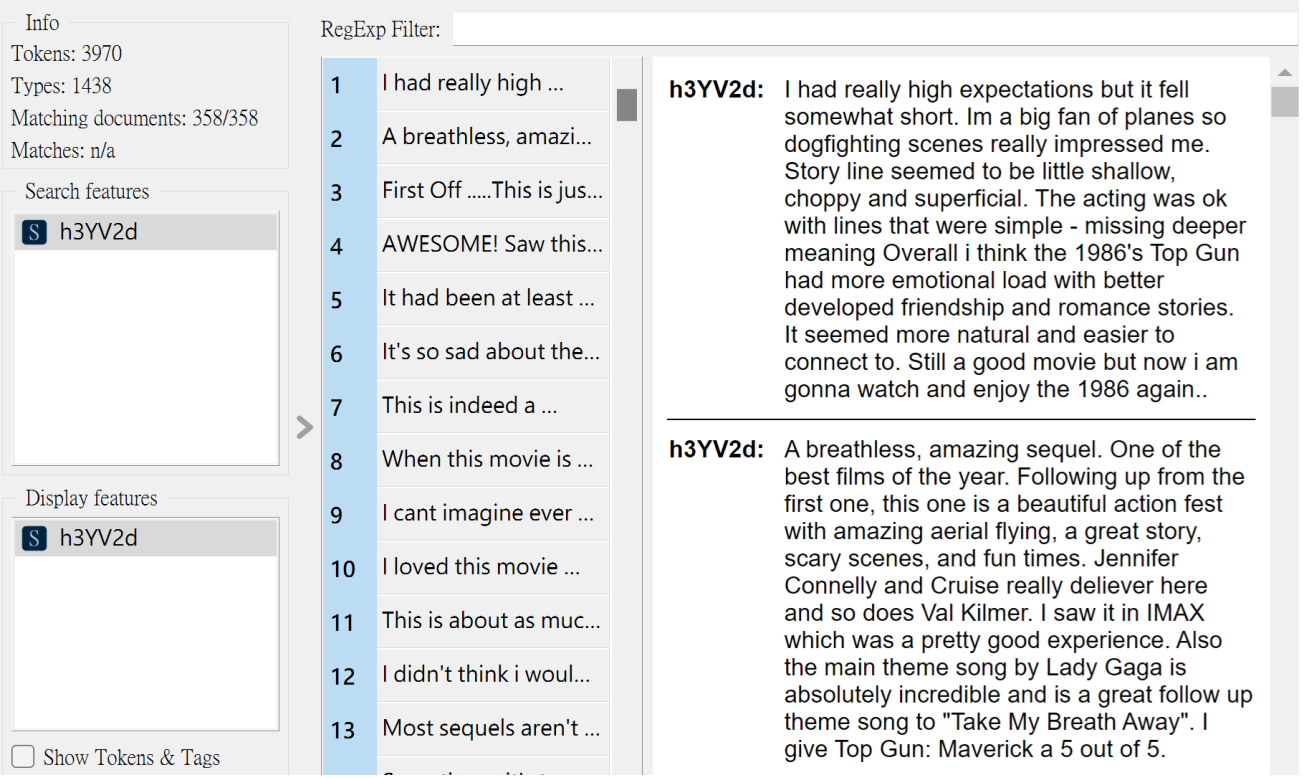


匯入csv檔進表格整理

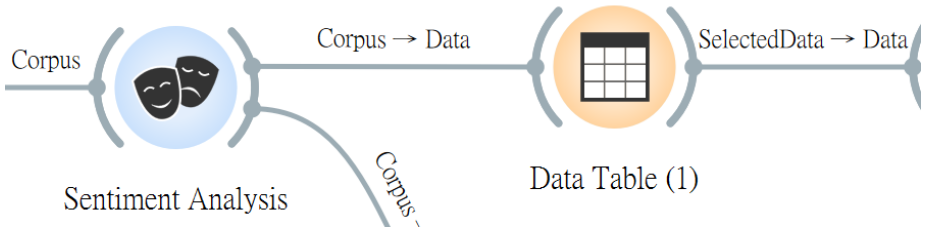


進行斷詞 標點符號清除


Corpus
MatchingDocs
Preprocess Text
Corpus Viewer

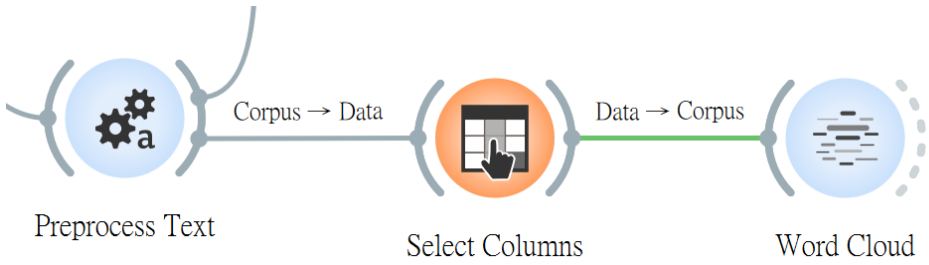

Info
Tokens: 3970
Types: 1438
Matching documents: 358/358
Matches: n/a
Search features
h3YV2d
Display features
h3YV2d
Show Tokens & Tags
RegExp Filter:
1 I had really high ...
2 A breathless, amazi...
3 First Off .....This is jus...
4 AWESOME! Saw this...
5 It had been at least ...
6 It's so sad about the...
7 This is indeed a ...
8 When this movie is ...
9 I cant imagine ever ...
10 I loved this movie ...
11 This is about as muc...
12 I didn't think i woul...
13 Most sequels aren't ...
h3YV2d: I had really high expectations but it fell somewhat short. Im a big fan of planes so dogfighting scenes really impressed me. Story line seemed to be little shallow, choppy and superficial. The acting was ok with lines that were simple - missing deeper meaning Overall i think the 1986's Top Gun had more emotional load with better developed friendship and romance stories. It seemed more natural and easier to connect to. Still a good movie but now i am gonna watch and enjoy the 1986 again..
h3YV2d: A breathless, amazing sequel. One of the best films of the year. Following up from the first one, this one is a beautiful action fest with amazing aerial flying, a great story, scary scenes, and fun times. Jennifer Connelly and Cruise really deliever here and so does Val Kilmer. I saw it in IMAX which was a pretty good experience. Also the main theme song by Lady Gaga is absolutely incredible and is a great follow up theme song to "Take My Breath Away". I give Top Gun: Maverick a 5 out of 5.

進行情緒分析



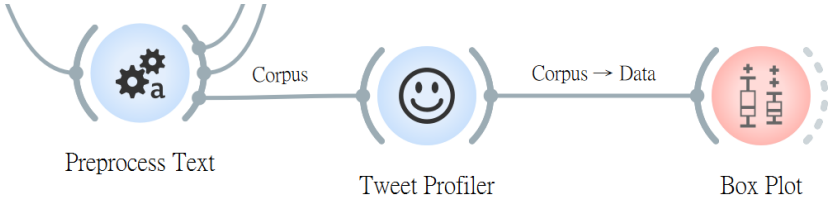
title	h3YV2d True	pos	neg	neu	compound
10	I loved this movie from start to finish. It's a cinematic experi...	0.419	0.017	0.564	0.993
1	I had really high expectations but it fell somewhat short. Im...	0.373	0.051	0.576	0.9902
21	Wow. Simply wow. Easily the best movie out there. This is a...	0.431	0	0.569	0.9887
27	Exceptional! This is a great movie and a worthy sequel, trul...	0.386	0	0.614	0.9857
2	A breathless, amazing sequel. One of the best films of the y...	0.309	0.049	0.642	0.9842
76	The best movies of 2022 so far. The movie had a good stor...	0.564	0	0.436	0.9796
14	Sometimes, it's typed, " I normally don't post my opinions",...	0.315	0.027	0.658	0.9786
11	This is about as much fun as you can have with a movie. It t...	0.349	0.022	0.628	0.975
60	This is the 7th time I've seen the movie. Six times in cinema...	0.375	0	0.625	0.9737
7	This is indeed a masterpiece movie, loved every bit of this ...	0.29	0.022	0.688	0.9731
37	If you liked the original “Top Gun” you are going to love “T...	0.365	0.025	0.609	0.9716
12	I didn't think i would say this but, it's better than the first o...	0.217	0.049	0.733	0.9682
79	I just saw this movie in theaters and I thought it was a great...	0.417	0	0.583	0.9652
101	Definitely the best movie out there Best movie and a long t...	0.517	0	0.483	0.9652
29	One of the best movies I've seen in a long time. The love in...	0.266	0.019	0.715	0.9644
28	I went into this movie knowing it would b good and thinkin...	0.342	0.044	0.614	0.9632

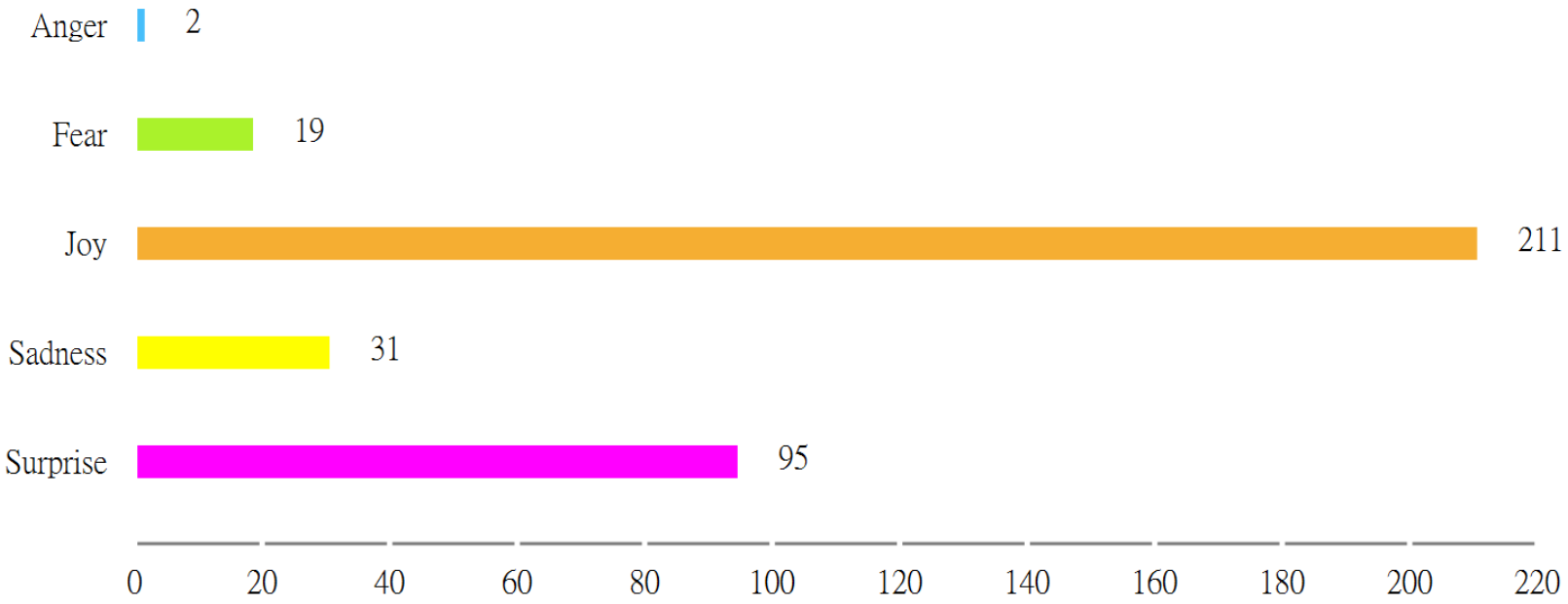
詞雲製作


Corpus → Data
Data → Corpus
Preprocess Text
Select Columns
Word Cloud

incredible
I thought
movie theater
I expected
Excellent
thought
liked
everything
expectations
emotional
storyline
fun movie
movie ever
better original
family
Tom Cruise
Absolutely
done
awesome
best movie
definitely
modern
could
watching
Awesome
seen long
different
still
Best movie
high
fighter
nice
made
work
Really
theater
line
Best
experience
back
watch
If
see
like
One
fantastic
Amazing
simple
Love
de
Thank
Watched
good movie
loved
Great
Gun
years
Cruise
I know
I loved
Loved
better
best
know
story
going
bad
kid
Better
NOT
Movie
Gun Maverick
little
top
Tom
I seen
good
first
ATop
enjoyed
kept
without
one I
long
Not
hit
fan
My
part
Very
release
well
great
short
since
1st
5
la
2
would
movies I
gun
acting
I saw
Just
I
It
seen
go
Most
film
really
think
From
I watched
Saw
times
movie
The
one
This
every
So
pretty
big
feel
long time
love
As
ok
saw
action
What
need
summer
cruise
take
much
seat
time
sequel
scenes
I think
see movie
first one
original
come
I really
When
new
amazing
want
good first
place
Don
fun
wait
10
ever
I ever
movies
Maverick
MOVIE
theaters
Great movie
Top Gun
watched
2022
character
always
absolutely
get even
worth
recommend
year
must
This movie
perfect
highly
This one
favorite
One best
say
expected
everyone
Was
best movies
first movie
put
ever seen
Better first
original Top
whole
great movie
seen years
great story
sequels

評論情緒分布


Corpus
Corpus → Data
Preprocess Text
Tweet Profiler
Box Plot


Anger
2
Fear
19
Joy
211
Sadness
31
Surprise
95
0
20
40
60
80
100
120
140
160
180
200
220

匯出結果儲存至csv

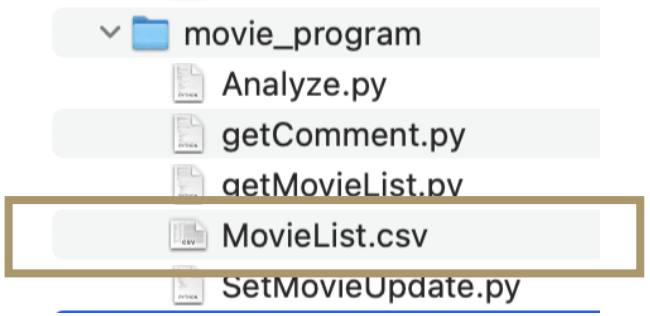
- movie_analyze
  - 3idiots.csv
  - 47MetersDown.csv
  - 47MetersDownUncaged(2019).csv
  - AlbertNobbs.csv
  - Avatar.csv
  - BlackAdam.csv

3idiots

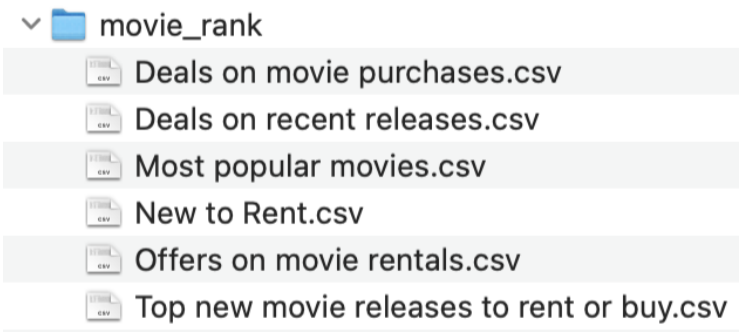
正向	負向	中性		評論
pos	**neg**	**neu**	**compound**	**comment**
continuous	continuous	continuous	continuous	string
				meta
0.152	0.187	0.661	0.0227	They didn't do a good enough job of explaining why he left th
0.0	0.0	1.0	0.0	Rip off from Bollywood movie of same name 3 Idiots.
0.0	0.0	1.0	0.0	It's adopted from the Bollywood movie 3 idiots...
0.0	0.0	1.0	0.0	Rip off of Bollywood's three idiots
0.328	0.0	0.672	0.5994	Loved it thought it was so funny...loved the actors
0.0	0.0	1.0	0.0	Copies Indian movie. Same name in India too. 3 idiooooos
0.0	0.617	0.383	-0.7548	Really terrible movie waste of time
0.365	0.0	0.635	0.3182	Rip off the original 3 idiots

取得電影基本資訊，分別存放至csv

所有電影



電影分類



CSV格式

MovieList

	name	star	star_pc	comment	result	describe	trailer
0	BigTrip2SpecialDelivery	5.0	1.0		1.0	After his big trip, Mic Mic the bear just want	https://play.google.com/video/lava/web/
1	StarsatNoon	5.0	1.0		1.0	A young American journalist (Margaret Qual	https://play.google.com/video/lava/web/
2	Showroom	5.0	1.0		1.0	A story about a young woman named Liezl who dreams of making it big by working a	
3	AChristmasStory(1983)	5.0	1.0		1.0	It's the final days before Christmas in early	https://play.google.com/video/lava/web/

電影分數排名，
由高分至低分，
取得前10名

- Name-電影名稱
- Star-星級(0~5顆星)
- Star_pc-星級(0~1)
- Comment-評論百分比(0~1)
- Result-星級 + 評論百分比/2
- Describe-電影簡介
- Trailer-電影預告片

CSV格式

MovieList

	name	star	star_pc	comment	result	describe	trailer
0	BigTrip2SpecialDelivery	5.0	1.0		1.0	After his big trip, Mic Mic the bear just want	https://play.google.com/video/lava/web/
1	StarsatNoon	5.0	1.0		1.0	A young American journalist (Margaret Qual	https://play.google.com/video/lava/web/
2	Showroom	5.0	1.0		1.0	A story about a young woman named Liezl who dreams of making it big by working a	
3	AChristmasStory(1983)	5.0	1.0		1.0	It's the final days before Christmas in early	https://play.google.com/video/lava/web/

評分計算方式

```
#資料筆數
cnt = df.shape[0]
for i in range(cnt):
    if(df['pos'][i] > df['neg'][i]):
        df['pos-neg'][i] = 1
    else:
        df['pos-neg'][i] = 0

#評論總正向評論數
sent_an = df['pos-neg'].sum()
#該筆評論的百分比(1:正向/0:負向)
comment = round(sent_an/cnt,2)
#movie name
movie = movie.replace('.csv','')
#將百分比更新回分析excel
pg['comment'] = np.where(pg['name'] == movie, comment, pg['comment'])
```

① 透過前面情緒分析結果，比較每個評論正負向指數

② 若正向>負向，則該評論視為正評(1)，反之為負評(0)

③ 總評論數為分母，正向評論數為分子，取得百分比

④ 評論百分比、星級百分比相加取平均為最終評分

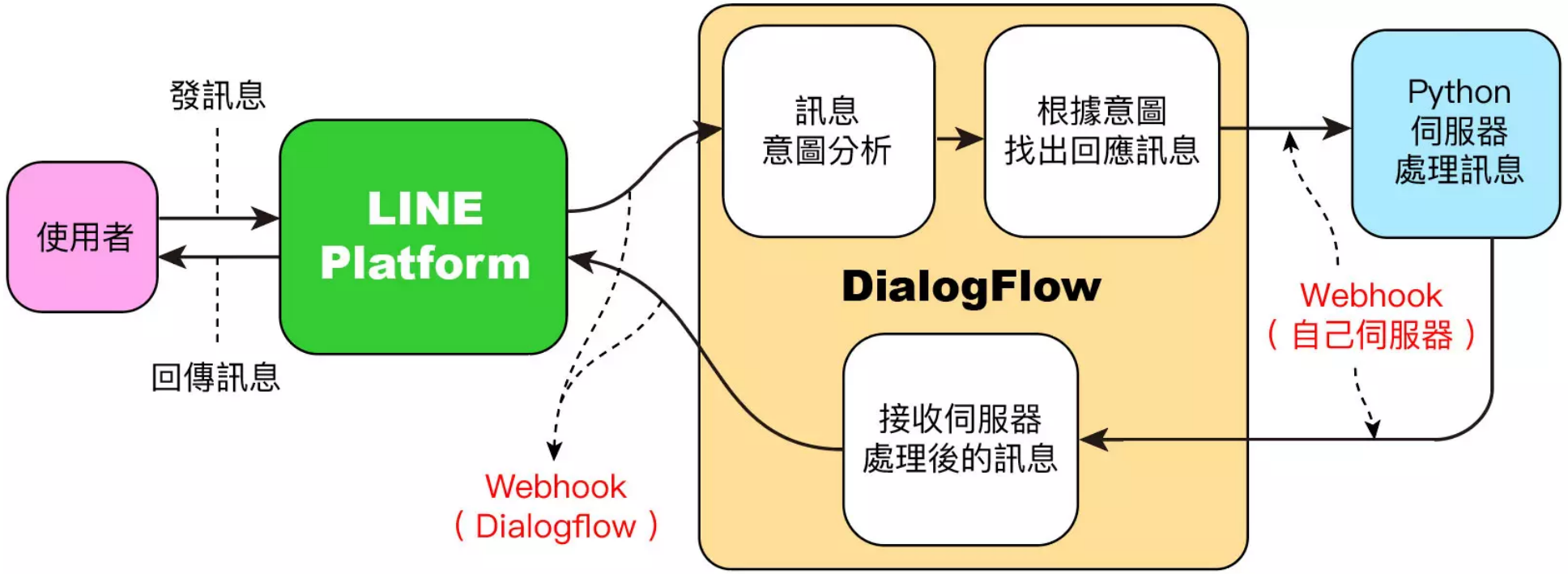
⑤ 若無評論，以星級百分比為最終評分

輸出結果

MovieList

	name	star	star_pc	comment	result	describe	trailer
0	BigTrip2SpecialDelivery	5.0	1.0		1.0	After his big trip, Mic Mic the bear just want	https://play.google.com/video/lava/web/
43	DieHard	4.7	0.94	0.71	0.82	Facing Christmas 3,000 miles from his estranged wife and two children, New York po	
44	TheWomanKing	4.1	0.82	0.83	0.82	Inspired by true events, this is the remarkab	https://play.google.com/video/lava/web/
45	3Idiotas	4.1	0.82		0.82	International star Martha Higareda shines in	https://www.youtube.com/embed/ttOj7_
46	BulletTrain	4.3	0.86	0.74	0.8	In Bullet Train, Brad Pitt stars as Ladybug, a	https://play.google.com/video/lava/web/
47	DrSeussHowtheGrinchStoleChristmas	4.6	0.92	0.69	0.8	You Better Watch Out! He's green...he's me	https://play.google.com/video/lava/web/

使用者訊息回覆流程


發訊息
使用者
回傳訊息
LINE Platform
Webhook
（Dialogflow）
訊息
意圖分析
根據意圖
找出回應訊息
DialogFlow
接收伺服器
處理後的訊息
Python
伺服器
處理訊息
Webhook
（自己伺服器）

DialogFlow訊息分類

Default Fallback Intent	
Default Welcome Intent	打招呼
end	結束對話
info_input	選擇想獲取資訊(推薦)
info_input1	選擇想獲取資訊(自選)

movie_ask	確認是否要看電影
movie_find	全部電影的排行榜
movie_find_info	輸入排行榜上電影
movie_input	輸入想找的電影
movie_type	顯示榜單類別

取得排行榜的類別

```
if query_result.get('action')=="movie_find": #取得電影類別，推薦相對應的電影
    req1=req.get("queryResult")
    param=req1.get("parameters")
    type=param.get("movie_type")
    if type=="全部電影": #輸入隨便，則會以全部電影的排行去推薦
      file_name="movie_list"
    elif type=="Deals on movie purchases" or type=="Deals on recent releases" or type=="Most popular movies" or
        type=="New to Rent" or type=="Offers on movie rentals" or type=="Top new movie releases to rent or buy" or
        type=="Analysis with NLP":
        file_name=type  #以接收的電影類別判斷要開啟相對應的檔案
    wb = openpyxl.load_workbook(file_name+".xlsx")
    ws = wb.active
    myList = [row for row in ws.values]
    myArr = numpy.array(myList)
    for i in range(1,11): #推薦10部電影
        string=" "+str(i)+". "+myArr[i][1]+"\n"
        movie_list.append(string)
    movie_recom= "".join(movie_list) #列表轉字串
    fullfillmentText="以下是"+type+"類的電影排行榜\n"+movie_recom+"輸入您想看的電影，我們可以為您提供電影的相關資訊" #回傳的文字
```

取得排行榜上的電影名稱

```
if query_result.get('action')=="movie_find_info": #取得電影名稱
    req1=req.get("queryResult")
    param=req1.get("parameters")
    global movie_name
    movie_name=param.get("movie_name")
    fullfillmentText="好的，輸入「1」可以查看電影簡介，輸入「2」可以查看電影預告片，輸入「3」可以查看電影在Google電影中的星等。\n" +
                      "請輸入您想查看的電影資訊："
```

回傳排行榜上的電影資訊

```
if query_result.get('action')=="info_input": #取得回傳資訊的代碼，回傳電影資訊
    info_num={"1":"電影簡介:","2":"電影預告片:","3":"電影星等"}
    req1=req.get("queryResult")
    param=req1.get("parameters")
    num=param.get("movie_info")
    wb = openpyxl.load_workbook("movie_list.xlsx")
    ws = wb.active
    myList = [row for row in ws.values]
    myArr = numpy.array(myList)
    for i in range(1,len(myArr)):
        if myArr[i][1]==movie_name:
            index=i
    if num=="1":
        describe=myArr[index][5]
        if describe==="":
            describe="暫無電影簡介"
        fullfillmentText="以下是"+movie_name+"的"+info_num[num]+"\n"+describe
    elif num=="2":
        trailer=myArr[index][6]
        if trailer==="":
            trailer="暫無預告片"
        fullfillmentText="以下是"+movie_name+"的"+info_num[num]+"\n"+trailer
    elif num=="3":
        star=myArr[index][2]
        if star==="":
            star="暫無星等"
        fullfillmentText=movie_name+"在google電影上的"+info_num[num]+"為"+str(star)
    else:
        fullfillmentText="請重新輸入電影名稱"
```

取得自選電影的電影名稱

```
if query_result.get('action')=="movie_input": #取得電影名稱，並先爬蟲爬取電影資訊
    req1=req.get("queryResult")
    param=req1.get("parameters")
    global movie_name1,trailer1,describe1,tomato
    movie_name1=param.get("new_movie_name")
    url="https://play.google.com/store/search?q="+movie_name1+"&c=movies&hl=zh_TW&gl=US"
    r=requests.get(url)
    resp = BeautifulSoup(r.text, 'html.parser')
    detail="https://play.google.com"+resp.find("a","Si6A0c ZD8Cqc").get("href")
    r1=requests.get(detail)
    resp1 = BeautifulSoup(r1.text, 'html.parser')
    describe1=resp1.find('div','bARER').text
    try:
        star=resp1.find('div', 'jILTFe').text.replace('\n','').replace(' ','').replace('star','')
    except:
        star="無"
    try:
        trailer1 = resp1.find('div','kuvzJc').find("button").get("data-trailer-url")
    except:
        trailer1 = None
    fullfillmentText="好的，"+movie_name1+"在Google電影中的星評為"+str(star)+"顆星。\n"+
        "另外，輸入「A」可以查看電影簡介，輸入「B」可以查看電影預告片。\n請輸入您想查看的電影資訊："
```

回傳自選電影的電影資訊

```
if query_result.get('action')=="info_input1": #取得回傳資訊的代碼，回傳電影資訊
    req1=req.get("queryResult")
    param=req1.get("parameters")
    num1=param.get("movie_info")
    if num1=="A":
        fullfillmentText="以下是"+movie_name1+"的電影簡介：\n"+describe1
    if num1=="B":
        fullfillmentText="以下是"+movie_name1+"的電影預告片：\n"+trailer1
return {     #以json格式回傳
        "fulfillmentMessages": [
            {
                "text": {
                "text": [
                    fullfillmentText
                ]
                }
            }
        ]
        }
```

04 成果展現

聊天機器人QR code

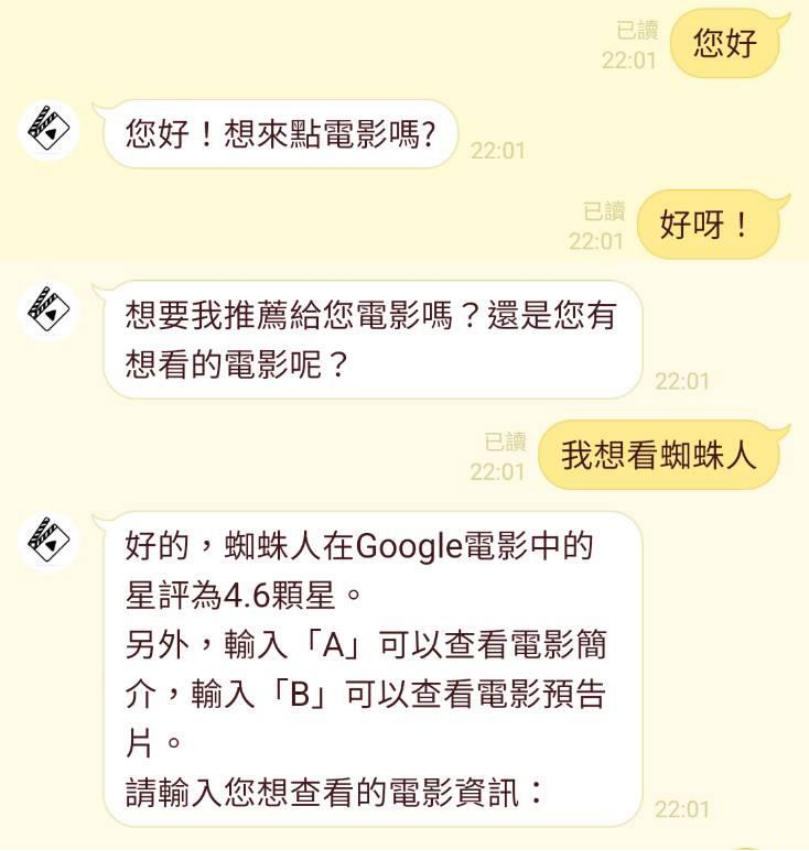
尋找在排行榜上的電影

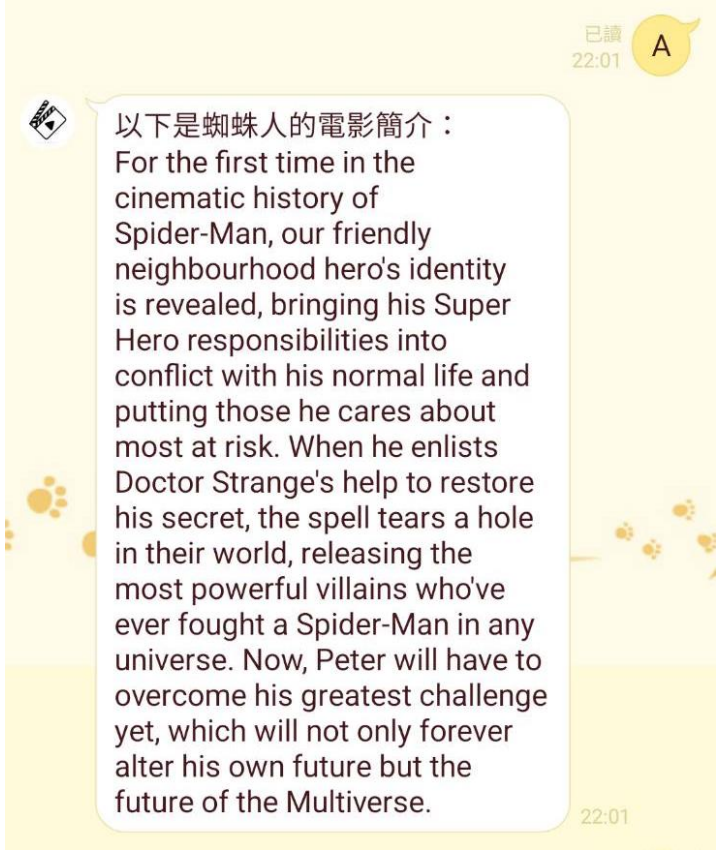
已讀 22:04 安安
嗨 想不想來點好看的電影？ 22:04
已讀 22:04 想！
想要我推薦給您電影嗎？還是您有想看的電影呢？ 22:04
已讀 22:04 請推薦給我
我們有Deals on movie purchases、Deals on recent releases、Most popular movies、New to Rent、Offers on movie rentals、Top new movie releases to rent or buy、Analysis with NLP，你想看哪一類呢? 22:04

已讀 22:04 Analysis with NLP
以下是Analysis with NLP類的電影排行榜
1. BigTrip2SpecialDelivery
2. AChristmasStory(1983)
3. SummitFever
4. OneLastDeal
5. StarsatNoon
6. Waiting
7. TheInhabitant
8. 捍衛任務
9. Lyle,Lyle,Crocodile
10. Wonder
輸入您想看的電影，我們可以為您提供電影的相關資訊 22:04
已讀 22:05 SummitFever
好的，輸入「1」可以查看電影簡介，輸入「2」可以查看電影預告片，輸入「3」可以查看電影在Google電影中的星等。
請輸入您想查看的電影資訊： 22:05

已讀 22:05 2
以下是SummitFever的電影預告片:
https://play.google.com/video/lava/web/player/yt:movie:pBg2WnmLEcA.P?autoplay=1&embed=play
Google Play 電影
登入
22:05
已讀 22:05 3
SummitFever在google電影上的電影星等為5.0 22:05
已讀 22:05 謝謝你！！
謝謝您，祝您有個美好的電影時光 22:05

尋找不在排行榜上的電影





THANK YOU