文字探勘X電影評論

05170107 巨資三A 侯聖恩

以《黑豹》為例

05170107 巨資三A 侯聖恩

目錄

專案簡介

1 動機

1 動機

1 目的

1 目的

1 好評關鍵字

從Yahoo電影討論區抓取四星五星好評並找出評論中較常出現的關鍵字。

2 情感分析

用SnowNLP分析PTT上評論留言的情感平均分。

資料集介紹

2 資料集

Yahoo 電影討論區

一部電影的討論大約30-50頁
每頁10則留言

- 評分1-5顆星
- 一二星差評、四五星的好評
- support較大的關鍵字
- 以一頁為一文本

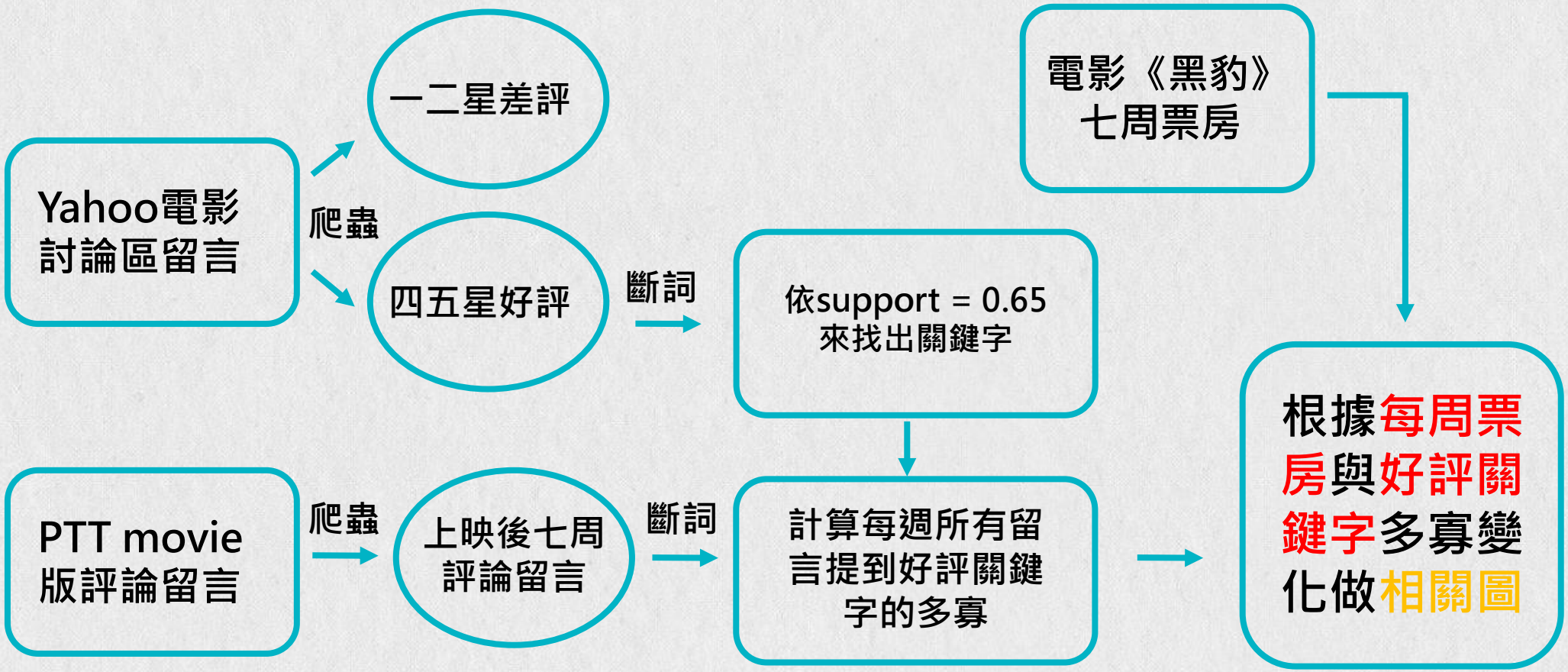
PTT movie版

大約30頁，每頁20則討論，
每則可能有10到上百則留言不等

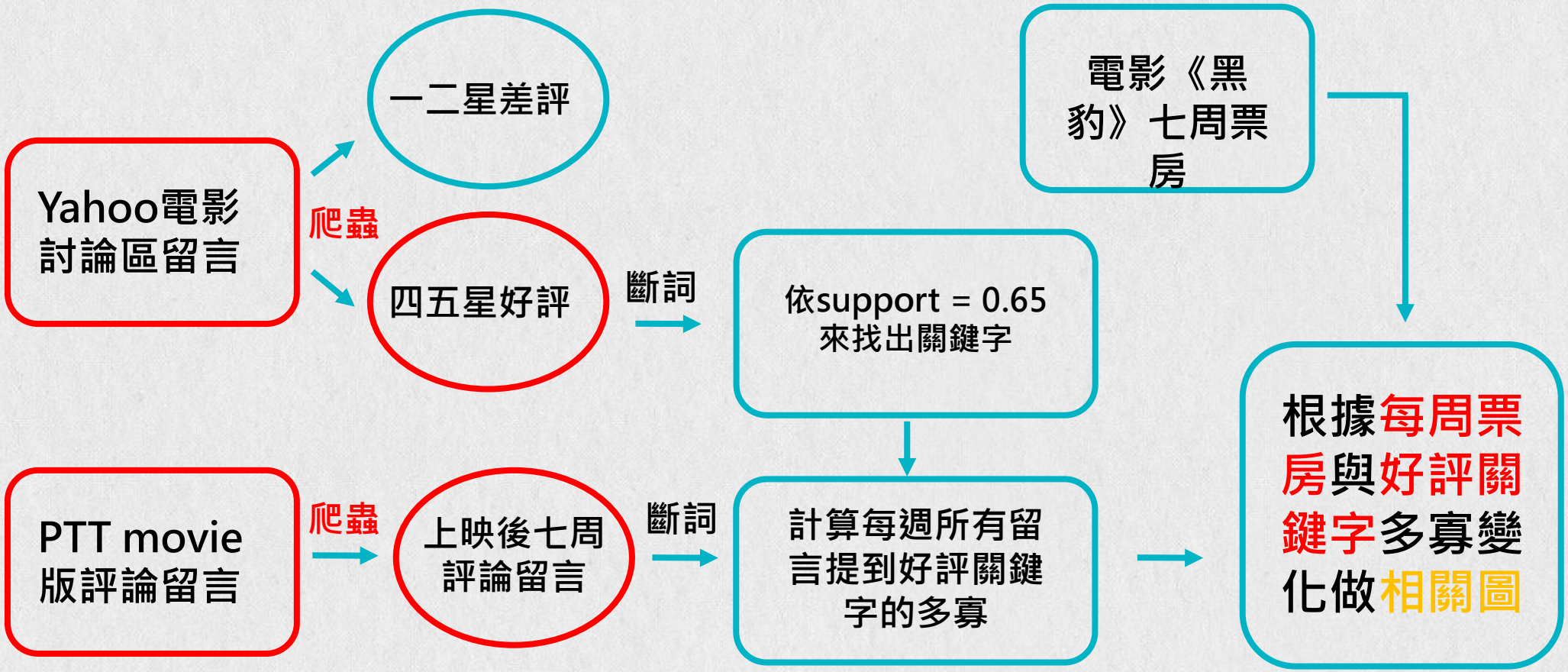
- 資料量較大
- 以周為區分，分上映後七週
- 好/負雷、新聞
- 只取留言，不取內文

實作方法與流程

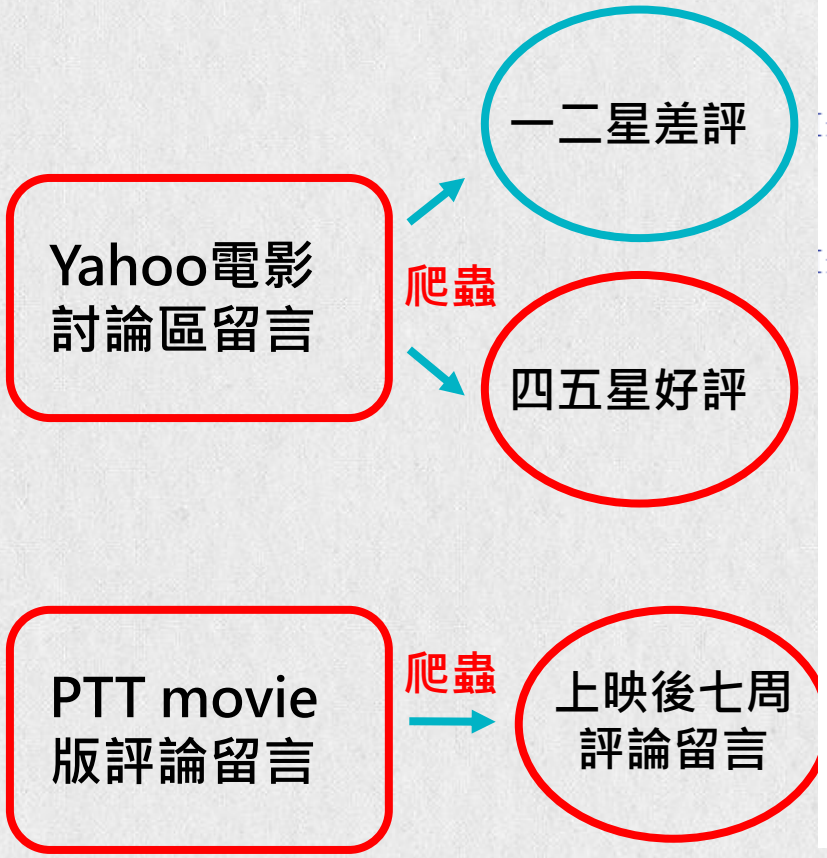
3 好評關鍵字實作流程


Yahoo電影
討論區留言
爬蟲
一二星差評
四五星好評
斷詞
依support = 0.65
來找出關鍵字
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
斷詞
計算每週所有留
言提到好評關鍵
字的多寡
根據每周票
房與好評關
鍵字多寡變
化做相關圖

3 好評關鍵字實作流程


Yahoo電影
討論區留言
爬蟲
一二星差評
四五星好評
斷詞
依support = 0.65
來找出關鍵字
電影《黑豹》七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
斷詞
計算每週所有留言提到好評關鍵字的多寡
根據每周票房與好評關鍵字多寡變化做相關圖

3 好評關鍵字實作流程



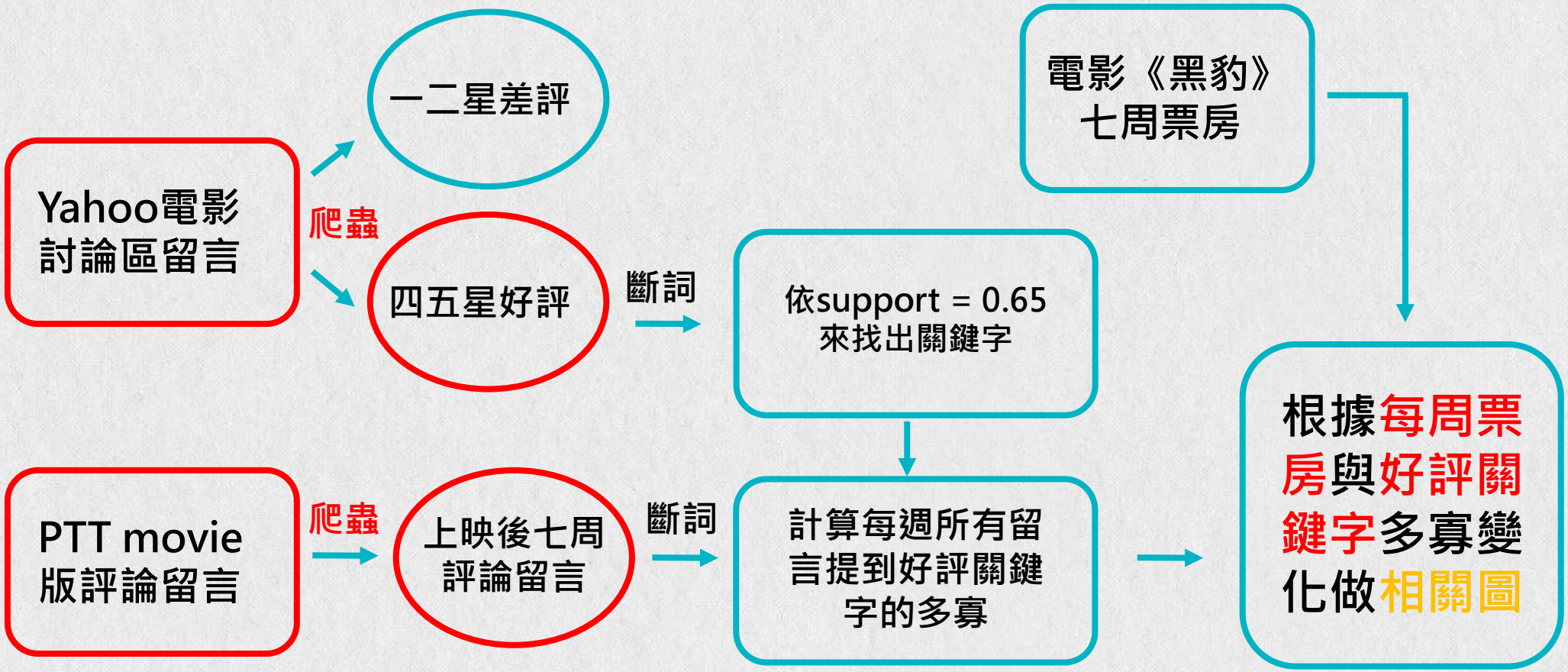
爬下四星五星好評資料

In [4]:
```
url_set1 = ['https://movies.yahoo.com.tw/movieinfo_review.html/id=6953?sort=ra
```

In [5]:
```
all_text = []
for i in url_set1:
    response = requests.get(i)
    soup = BeautifulSoup(response.text, 'lxml')
    articles = soup.find_all('div', 'usercom_inner _c')
    text = []
    for article in articles:
        messages = article.find('span',None).getText().replace(':','').strip()
        text.append(messages)
    all_text.append(text)
print(all_text[0])
```

```
['沒有冷場，超好看的啊，評價怎麼只有3.9\r\n個人覺得第一集總是要鋪陳和講述起
棒，有別於以往Marvel風格，值得一看', '覺得很好看 不知道為什麼有人只給1顆星真
麼才是爽片，趕快去看吧！北美票房已經破史上紀錄了57億....', '去看就對了啦，廢
有台灣人給3.9分真的笑掉人家大牙'  '節奏緊湊 動作頻繁 台式對白 會心一笑 (可以
```

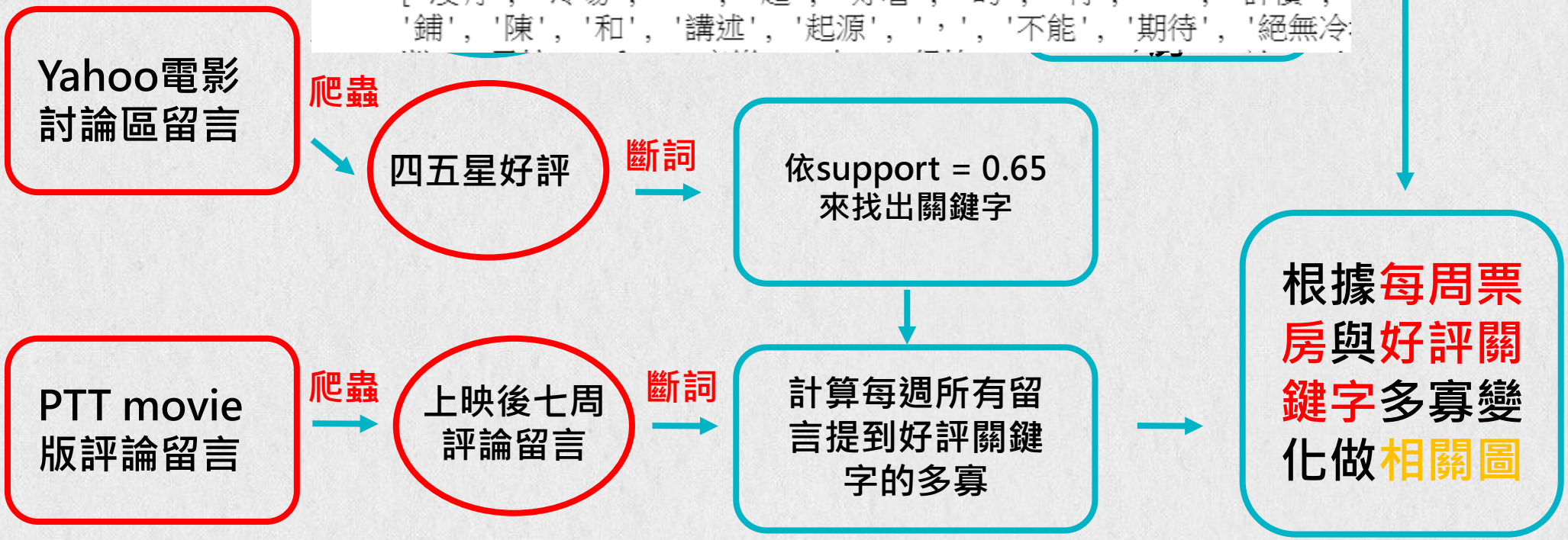
3 好評關鍵字實作流程


Yahoo電影
討論區留言
爬蟲
一二星差評
四五星好評
斷詞
依support = 0.65
來找出關鍵字
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
斷詞
計算每週所有留
言提到好評關鍵
字的多寡
根據每周票
房與好評關
鍵字多寡變
化做相關圖

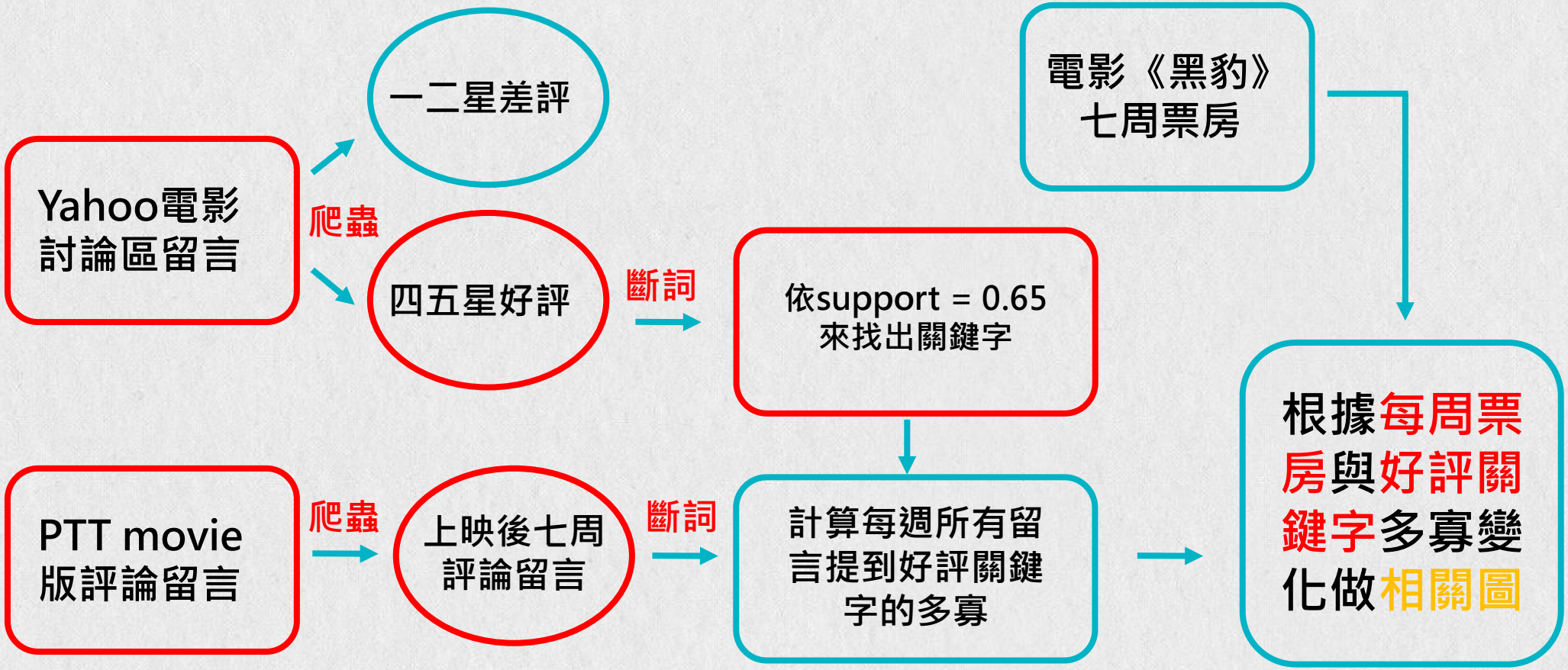
```
]: cut_text = [[' '.join(jieba.cut(w))for w in t] for t in all_text]
   document = [[s.split(' ') for s in i]for i in cut_text]
   clean_documents = [[ b for c in d for b in c]for d in document]
   print(clean_documents[0])
```

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\coco8\AppData\Local\Temp\jieb
Loading model cost 0.867 seconds.
Prefix dict has been built succesfully.
```

```
['沒有', '冷場', '，', '超', '好看', '的', '啊', '，', '評價',
'鋪', '陳', '和', '講述', '起源', '，', '不能', '期待', '絕無冷
```



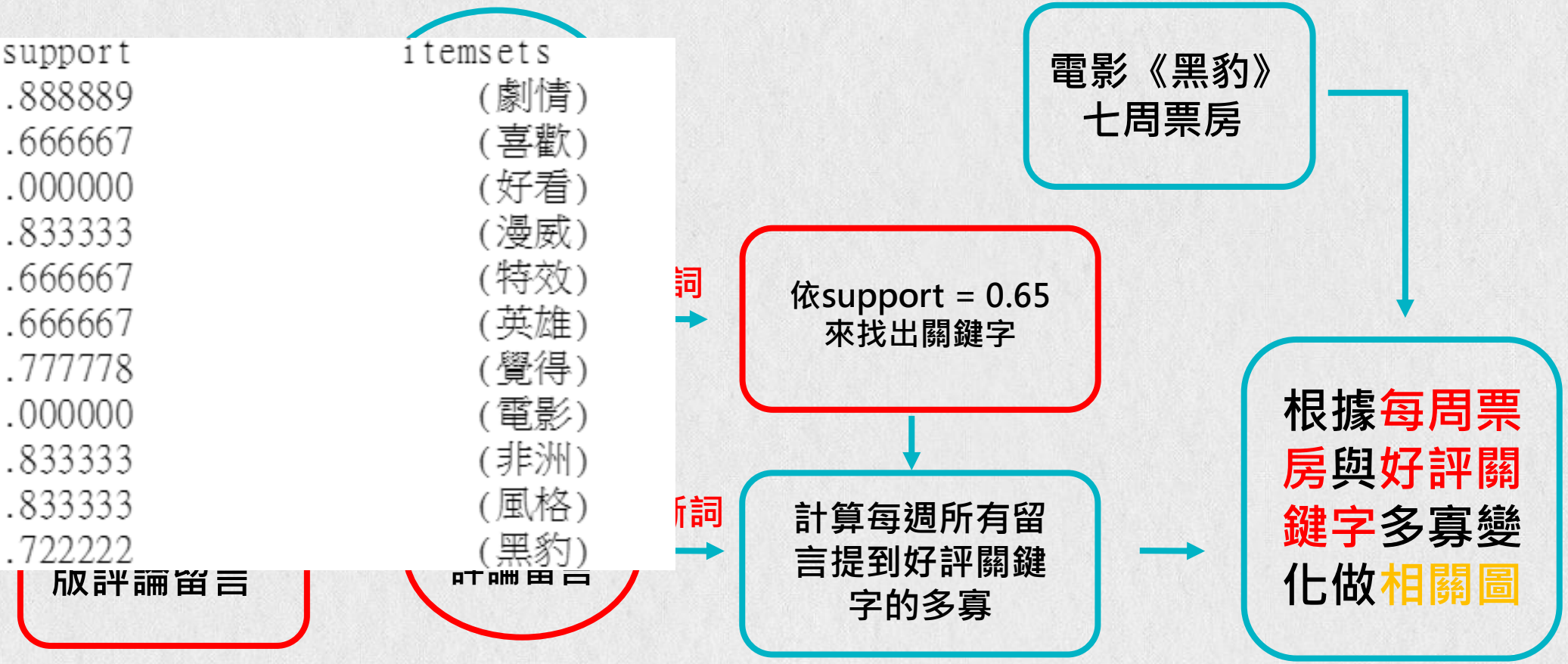
3 好評關鍵字實作流程


Yahoo電影
討論區留言
爬蟲
一二星差評
四五星好評
斷詞
依support = 0.65
來找出關鍵字
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
斷詞
計算每週所有留
言提到好評關鍵
字的多寡
根據每周票
房與好評關
鍵字多寡變
化做相關圖

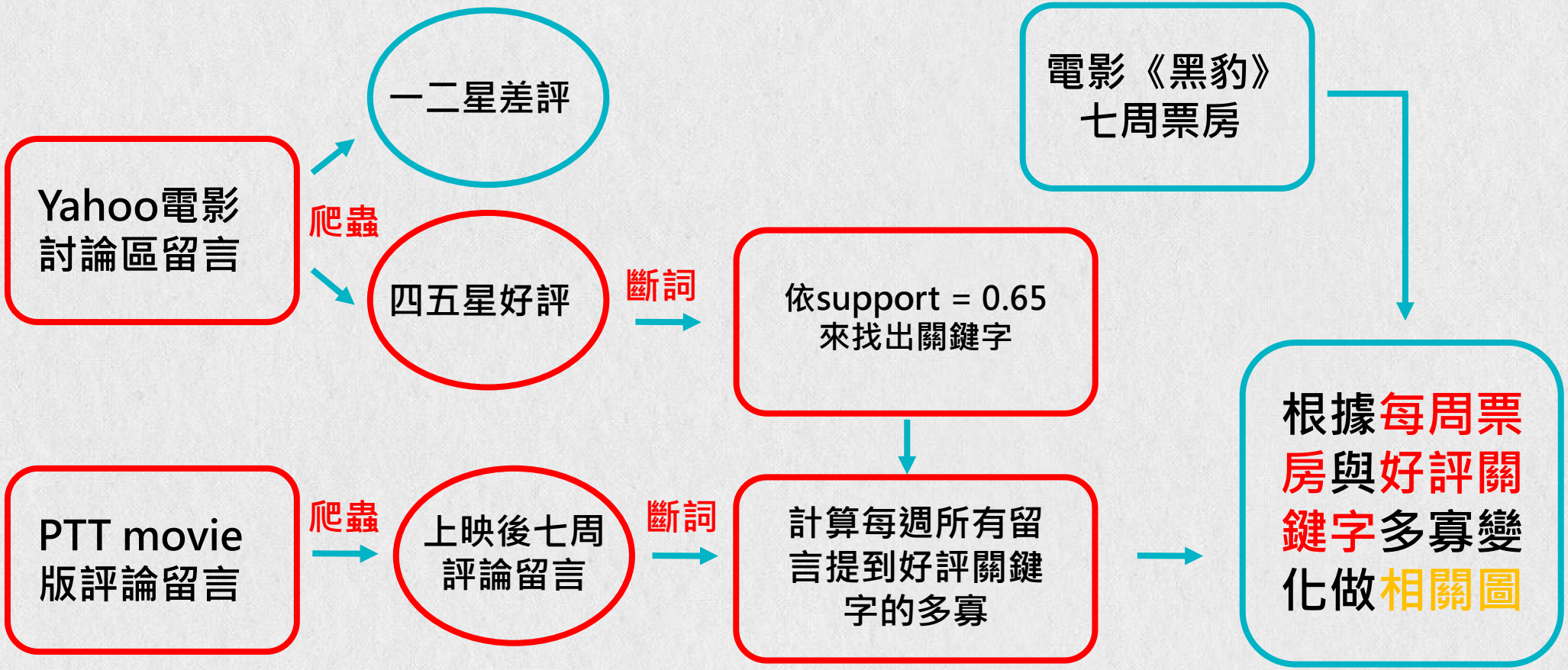
3

```
In [10]: te = TransactionEncoder()
         te_ary = te.fit(clean_documents).transform(clean_documents)
         df = pd.DataFrame(te_ary, columns=te.columns_)
         frequent_itemsets = apriori(df, min_support=0.65, use_colnames=True)
         print(frequent_itemsets)
```

	support	itemsets
0	0.888889	(劇情)
1	0.666667	(喜歡)
2	1.000000	(好看)
3	0.833333	(漫威)
4	0.666667	(特效)
5	0.666667	(英雄)
6	0.777778	(覺得)
7	1.000000	(電影)
8	0.833333	(非洲)
9	0.833333	(風格)
10	0.722222	(黑豹)



3 好評關鍵字實作流程


Yahoo電影
討論區留言
爬蟲
一二星差評
四五星好評
斷詞
依support = 0.65
來找出關鍵字
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
斷詞
計算每週所有留
言提到好評關鍵
字的多寡
根據每周票
房與好評關
鍵字多寡變
化做相關圖

3 好

```
In [53]: ptt_cut_text1 = [[' '.join(jieba.cut(w))for w in t] for t in all_text_week1]
         ptt_document1 = [[s.split(' ') for s in i]for i in ptt_cut_text1]
         ptt_clean_documents1 = [[ b for c in d for b in c]for d in ptt_document1]
         new_ptt_clean_documents1 = []
         for a in ptt_clean_documents1:
             for b in a:
                 new_ptt_clean_documents1.append(b)

         from collections import Counter
         c_week1 = Counter(new_ptt_clean_documents1)
         for a in c_week1.most_common(200):
             word, freq = a
             print(word,freq)
```

```
 3108
的 1379
， 1045
是 612
了 410
黑豹 359
就 345
我 341
有 340
也 321
很 291
看 289
? 268
```

Yahoo電影討論區留言

爬蟲

一二星差評

四五星好評

PTT movie版評論留言

爬蟲

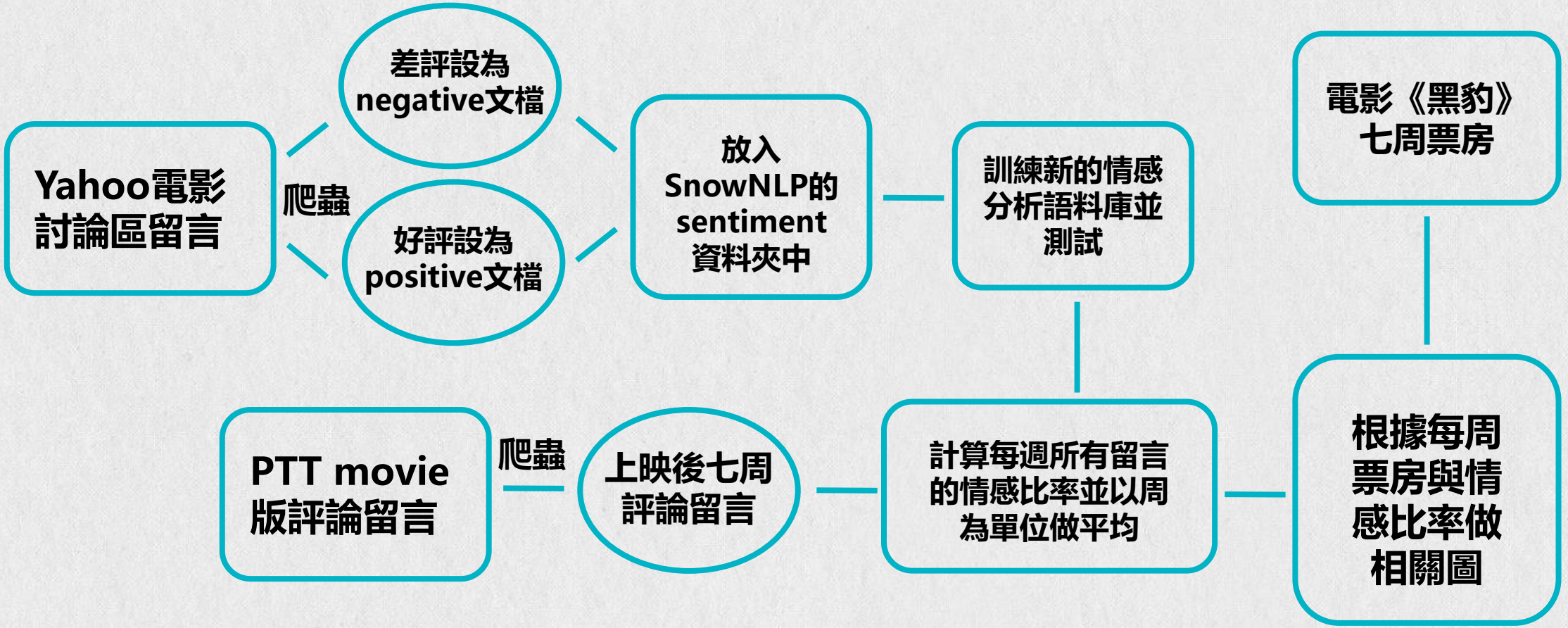
上映後七周評論留言

斷詞

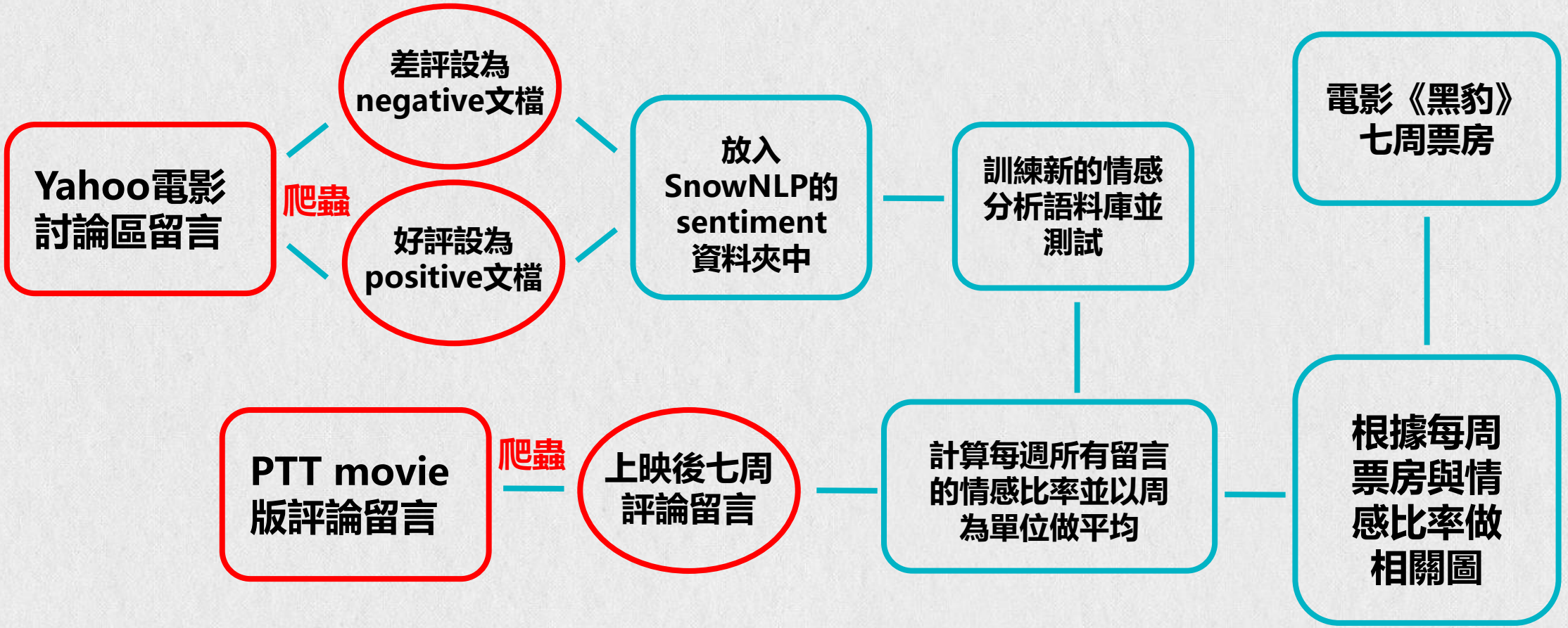
計算每週所有留言提到好評關鍵字的多寡

房與好評關鍵字多寡變化做相關圖

3 情感分析實作流程


Yahoo電影
討論區留言
爬蟲
差評設為
negative文檔
好評設為
positive文檔
放入
SnowNLP的
sentiment
資料夾中
訓練新的情感
分析語料庫並
測試
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
計算每週所有留言
的情感比率並以周
為單位做平均
根據每周
票房與情
感比率做
相關圖

3 情感分析實作流程


Yahoo電影
討論區留言
爬蟲
差評設為
negative文檔
好評設為
positive文檔
放入
SnowNLP的
sentiment
資料夾中
訓練新的情感
分析語料庫並
測試
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
計算每週所有留言
的情感比率並以周
為單位做平均
根據每周
票房與情
感比率做
相關圖

3

Yahoo電影
討論區留言

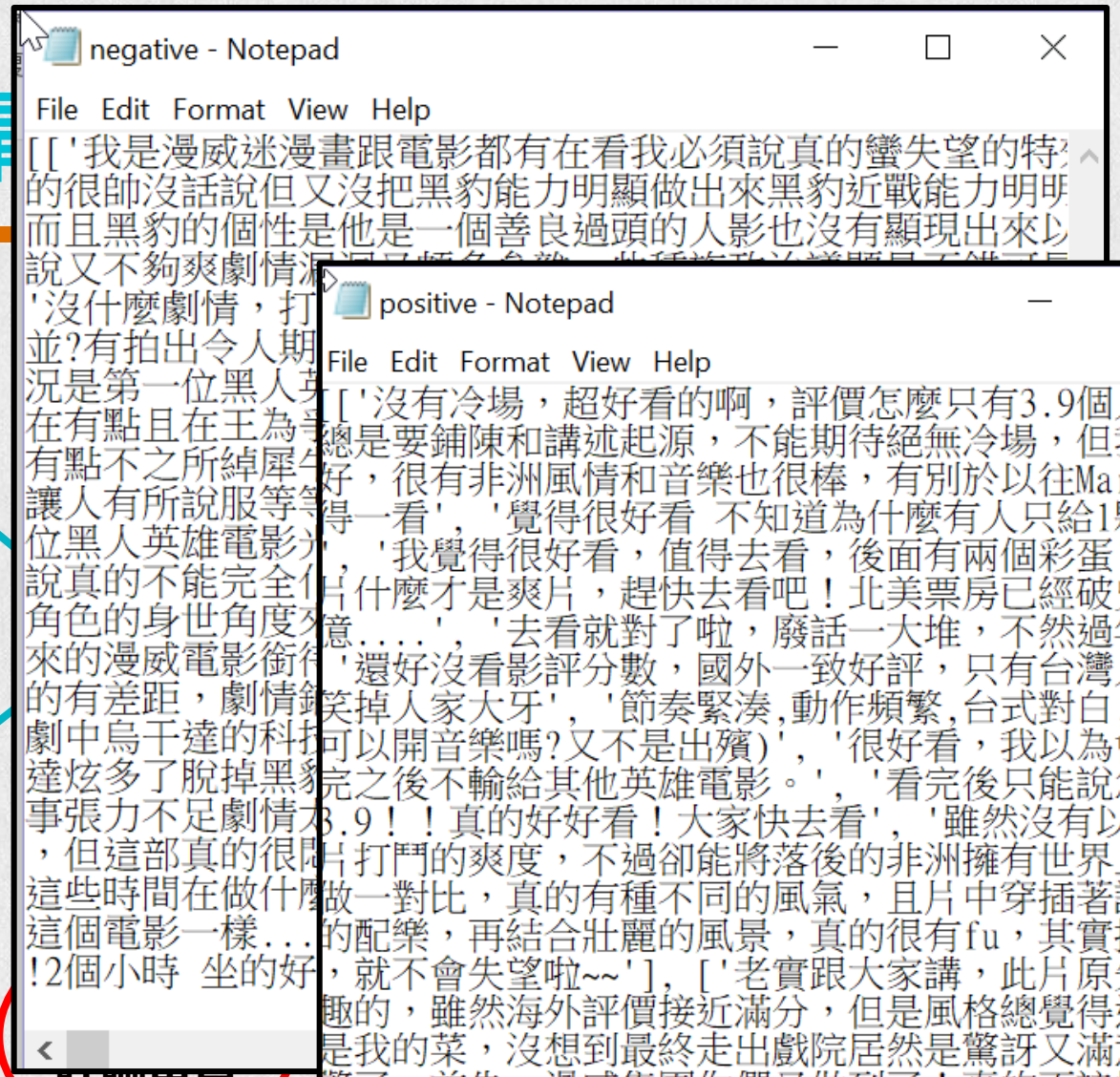
爬蟲

差評設為
negative文檔

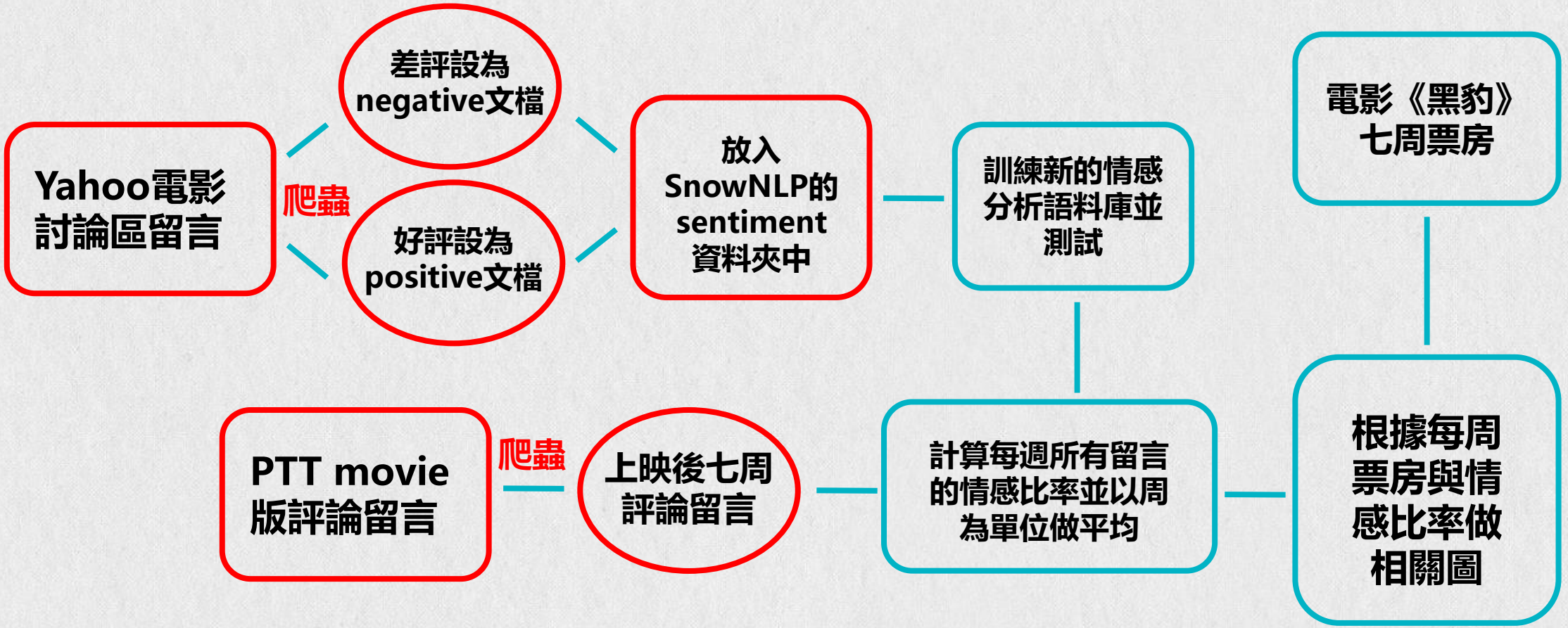
好評設為
positive文檔

PTT movie
版評論留言

爬蟲



3 情感分析實作流程


Yahoo電影
討論區留言
爬蟲
差評設為
negative文檔
好評設為
positive文檔
放入
SnowNLP的
sentiment
資料夾中
訓練新的情感
分析語料庫並
測試
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
計算每週所有留言
的情感比率並以周
為單位做平均
根據每周
票房與情
感比率做
相關圖

3 情感分析實作流程

Yahoo電影討論區留言

爬蟲

差評設為negative文檔

好評設為positive文檔

放入SnowNLP的sentiment資料夾中

Users > coco8 > Anaconda3 > Lib > site-packages > snownlp > sentiment

Name	Date modified	Type
__pycache__	12/27/2018 10:32 AM	File folder
__init__.py	12/27/2018 10:32 AM	PY File
neg	12/29/2018 4:46 PM	Text Document
negative	12/31/2018 1:36 PM	Text Document
pos	12/27/2018 10:32 AM	Text Document
positive	12/29/2018 4:47 PM	Text Document
sentiment.marshal	12/27/2018 10:32 AM	MARSHAL File
sentiment.marshal.3	12/27/2018 2:04 PM	3 File
sentiment2.marshal.3	12/27/2018 1:52 PM	3 File
sentimentHW.marshal.3	12/30/2018 9:11 PM	3 File

PTT movie版評論留言

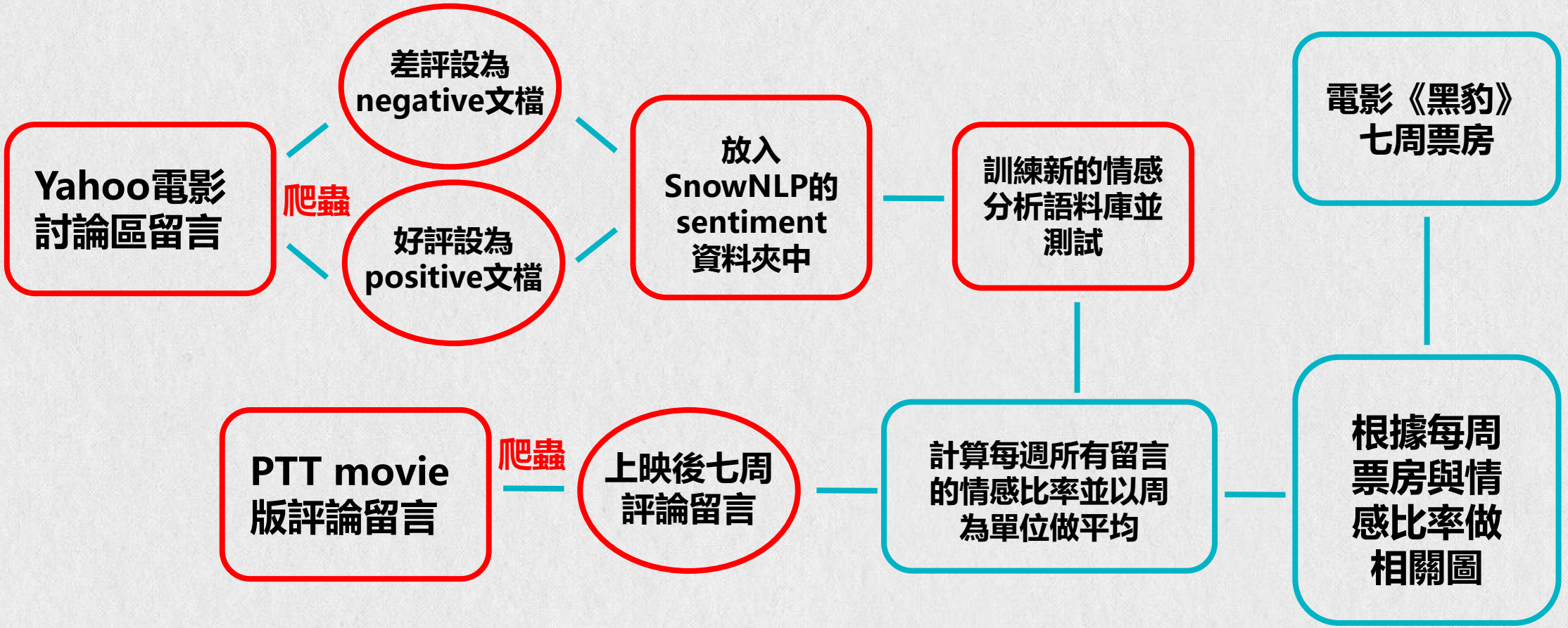
爬蟲

上映後七周評論留言

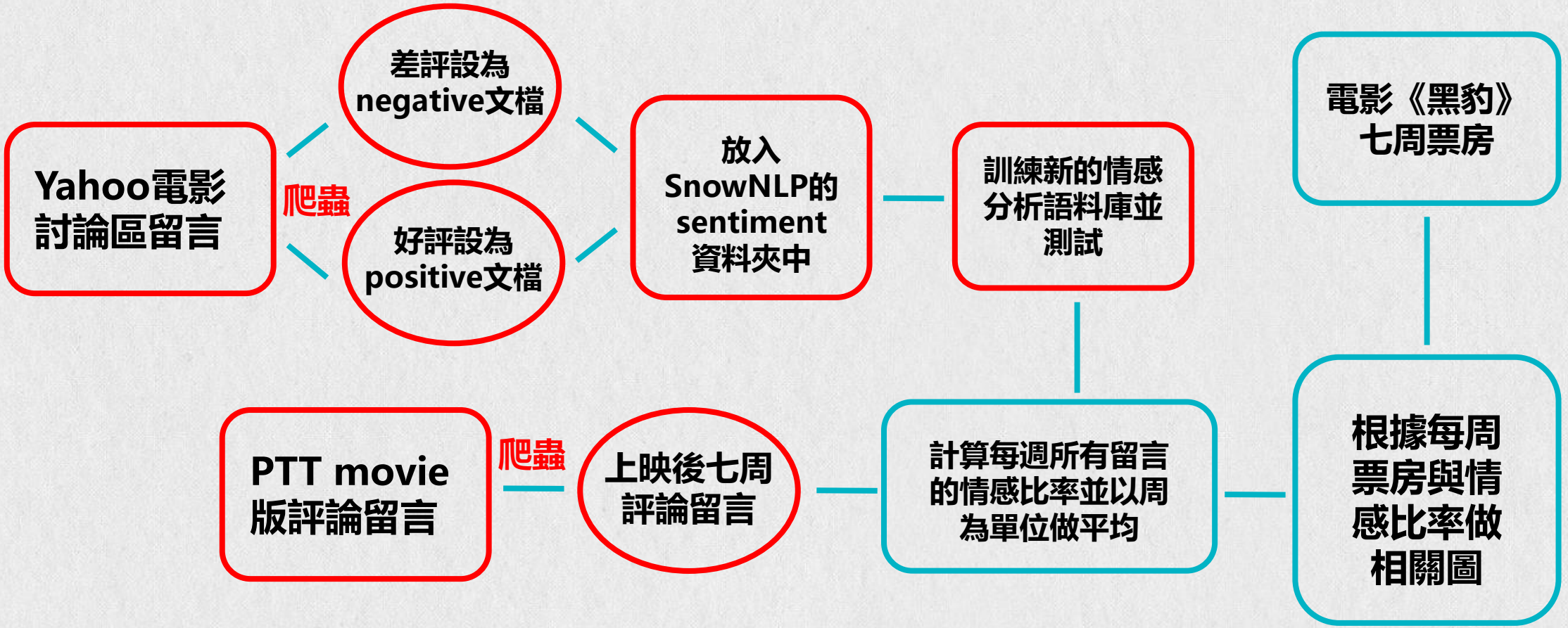
計算每週所有留言的情感比率並以周為單位做平均

根據每周票房與情感比率做相關圖

3 情感分析實作流程

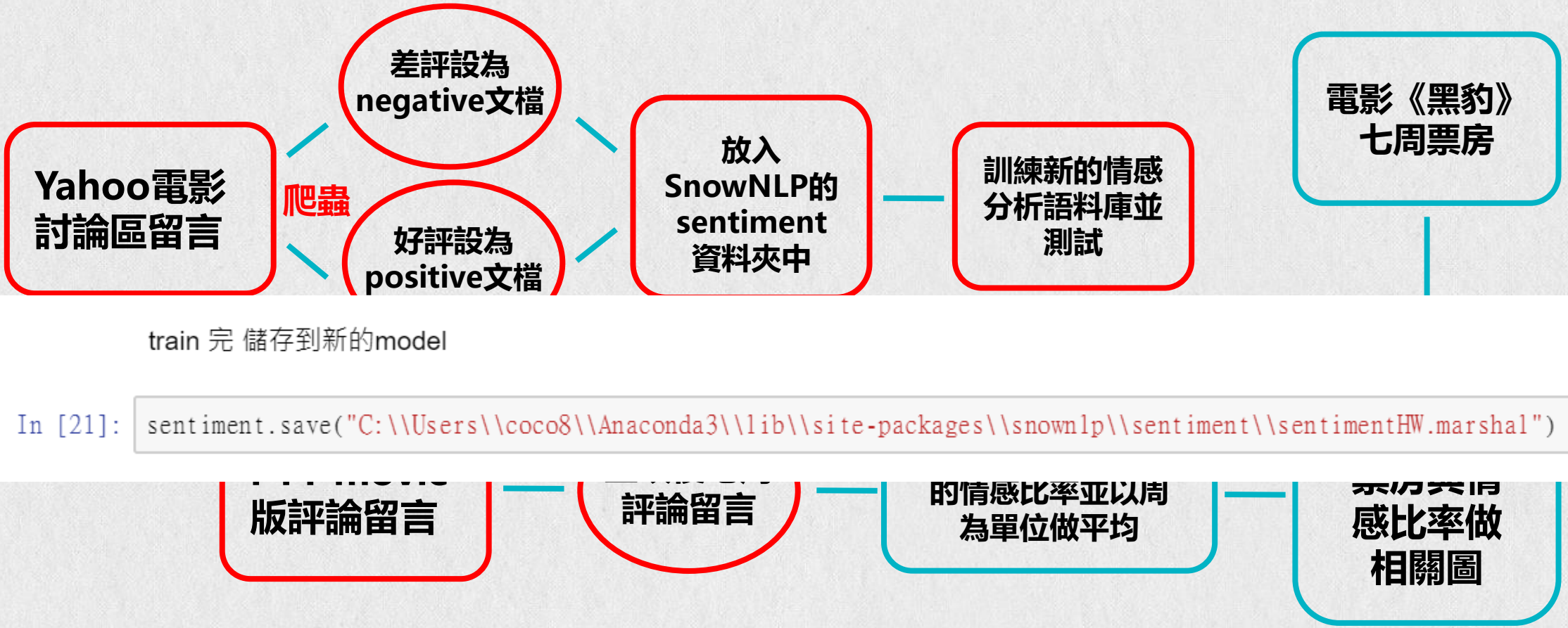

Yahoo電影
討論區留言
爬蟲
差評設為
negative文檔
好評設為
positive文檔
放入
SnowNLP的
sentiment
資料夾中
訓練新的情感
分析語料庫並
測試
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
計算每週所有留言
的情感比率並以周
為單位做平均
根據每周
票房與情
感比率做
相關圖

```
In [30]: from snownlp import sentiment

In [31]: sentiment.train("C:/Users/coco8/Anaconda3/Lib/site-packages/snownlp/sentiment/negative.txt",
                         "C:/Users/coco8/Anaconda3/Lib/site-packages/snownlp/sentiment/positive.txt")
```



```
In [30]: from snownlp import sentiment
```

```
In [31]: sentiment.train("C:/Users/coco8/Anaconda3/Lib/site-packages/snownlp/sentiment/negative.txt",
                         "C:/Users/coco8/Anaconda3/Lib/site-packages/snownlp/sentiment/positive.txt")
```



train 完 儲存到新的model

```
In [21]: sentiment.save("C:\\Users\\coco8\\Anaconda3\\lib\\site-packages\\snownlp\\sentiment\\sentimentHW.marshal")
```

```
In [30]: from snownlp import sentiment
```

```
In [31]: sentiment.train("C:/Users/coco8/Anaconda3/Lib/site-packages/snownlp/sentiment/negative.txt",
                         "C:/Users/coco8/Anaconda3/Lib/site-packages/snownlp/sentiment/positive.txt")
```

train 完 儲存到新的model

```
In [21]: sentiment.save("C:\\Users\\coco8\\Anaconda3\\lib\\site-packages\\snownlp\\sentiment\\sentimentHW.marshal")
```

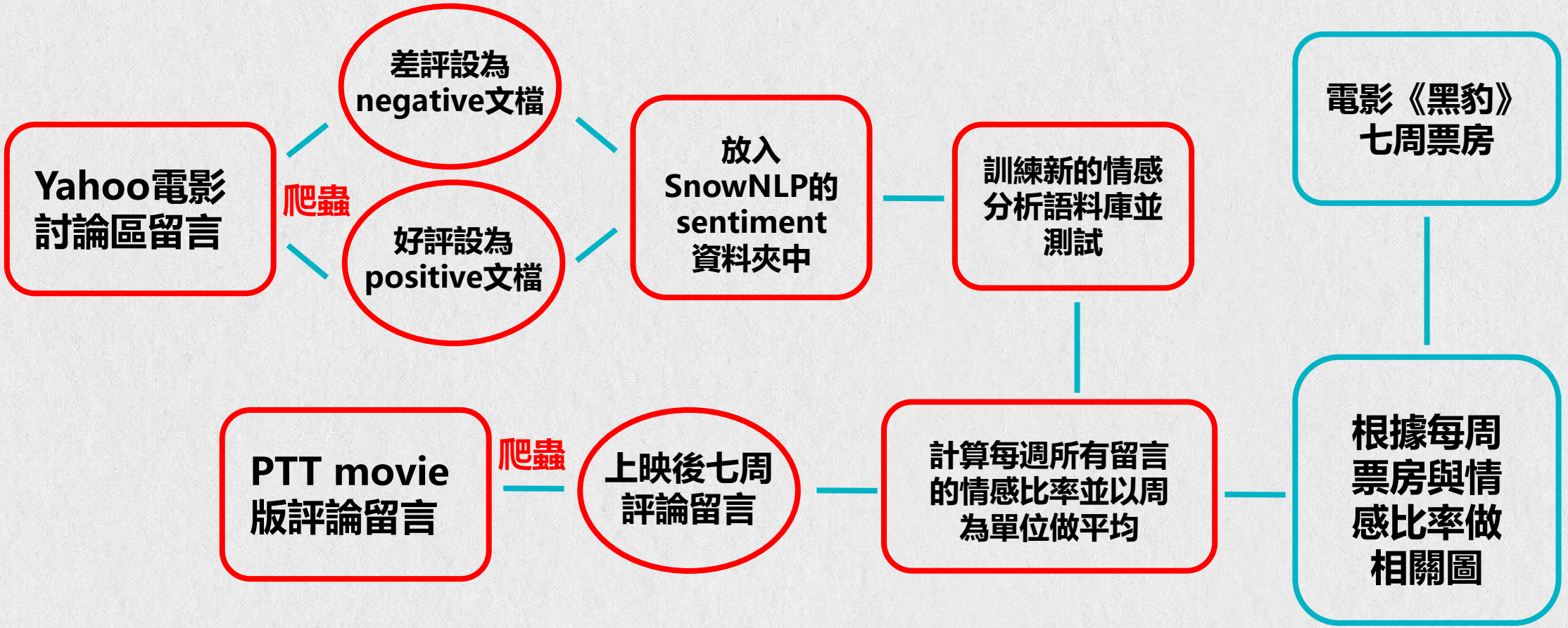
設定等等要使用的情感分析model =>指定路徑

```
In [22]: sentiment.data_path = "C:\\Users\\coco8\\Anaconda3\\lib\\site-packages\\snownlp\\sentiment\\sentimentHW.marshal"
```

3 情感分析實作流程


```
In [23]: text1 = "《黑豹》是一部很好看的電影"

In [24]: t = SnowNLP(text1)
         t.sentiments

Out[24]: 0.710521282967164 1

In [39]: text2 ='無聊、想睡'
         t2 = SnowNLP(text2)
         t2.sentiments

Out[39]: 0.04687695784575640 6
```

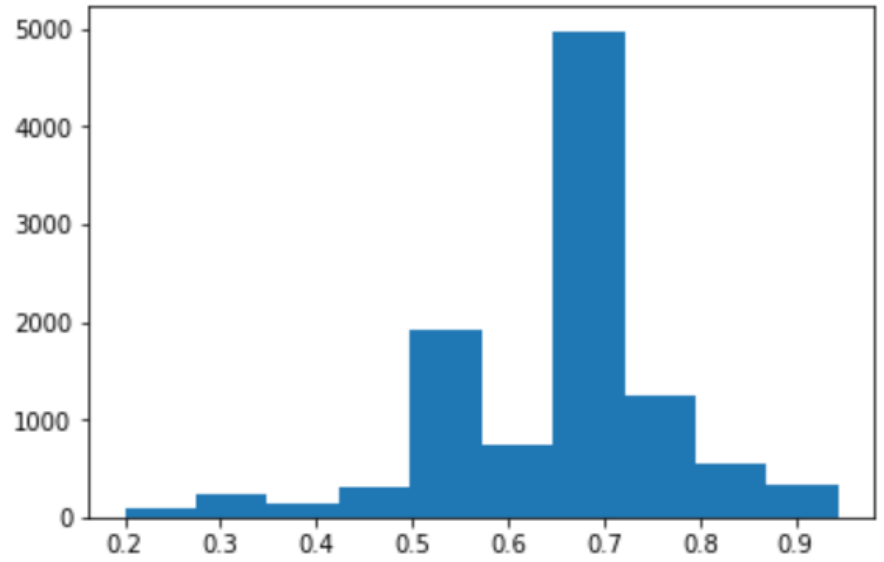
3 情感分析實作流程


Yahoo電影
討論區留言
爬蟲
差評設為
negative文檔
好評設為
positive文檔
放入
SnowNLP的
sentiment
資料夾中
訓練新的情感
分析語料庫並
測試
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
計算每週所有留言
的情感比率並以周
為單位做平均
根據每周
票房與情
感比率做
相關圖

3 情感分析實作流程

```
t2 = []
for w in new_all_text_week2:
    for a in w:
        r = SnowNLP(a)
        t2.append(r.sentiments)
print(t2)
plt.hist(t2)
plt.show()
```

```
0.5, 0.8999999999999999, 0.6906900775826866, 0.6855670103092784, 0
99999999, 0.727272727272727, 0.3999999999999999, 0.6906900775826866
775826866, 0.6666666666666666, 0.6666666666666666, 0.68556701030927
1818181818, 0.8888888888888888, 0.9375, 0.8888888888888888, 0.83333
```



電影《黑豹》七周票房

根據每周票房與情感比率做相關圖

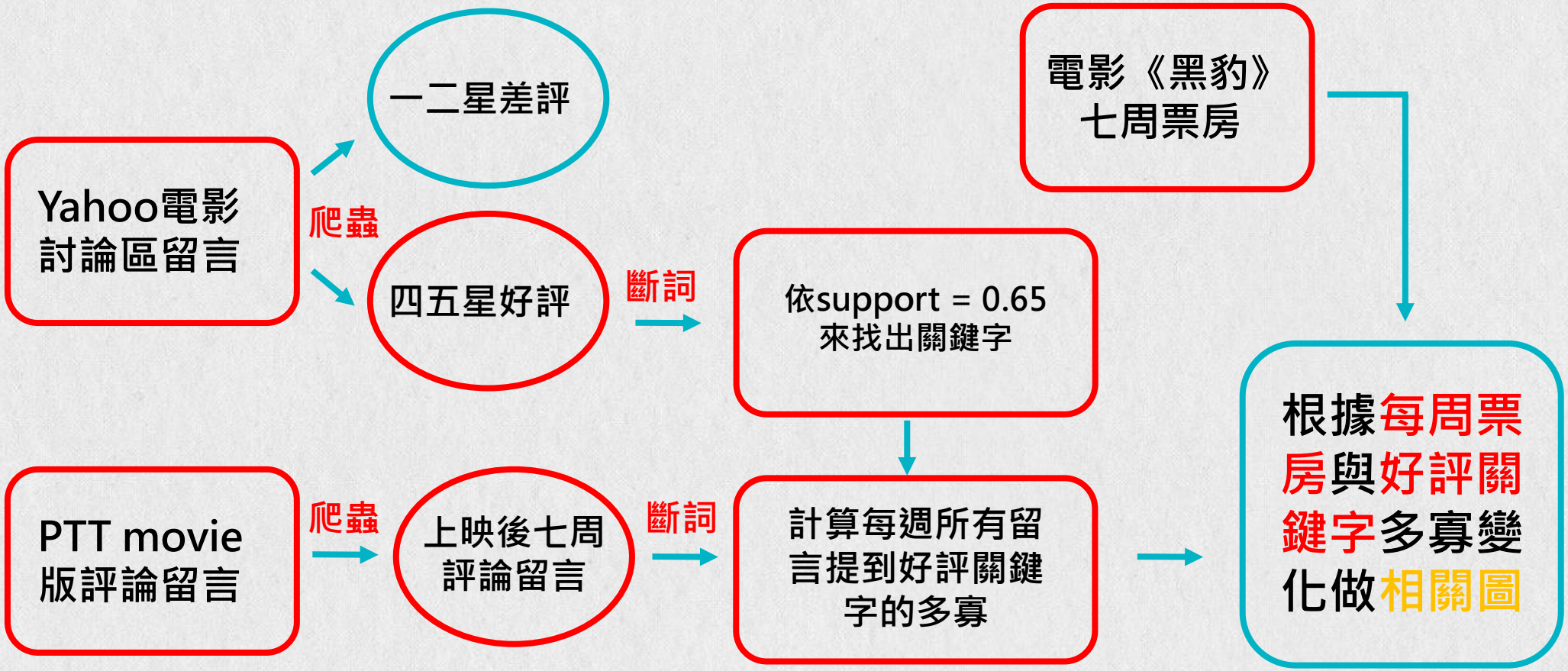
計算每週所有留言的情感比率並以周為單位做平均

訓練新的情感分析語料庫並測試

4

結果分析&結論

4 好評關鍵字實作流程


Yahoo電影
討論區留言
爬蟲
一二星差評
四五星好評
斷詞
依support = 0.65
來找出關鍵字
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
斷詞
計算每週所有留
言提到好評關鍵
字的多寡
根據每周票
房與好評關
鍵字多寡變
化做相關圖

4 好評關鍵字實作流程

- 上映七周票房的變化

本專案中為了消除各家電影業者的價格落差與隨著時間漸長上映的電影院數越少這兩項變因，故對於票房做出以下調整:

$$\text{各週票房} = \frac{\text{各週銷售電影票數}}{\text{各週上映電影院數}}$$

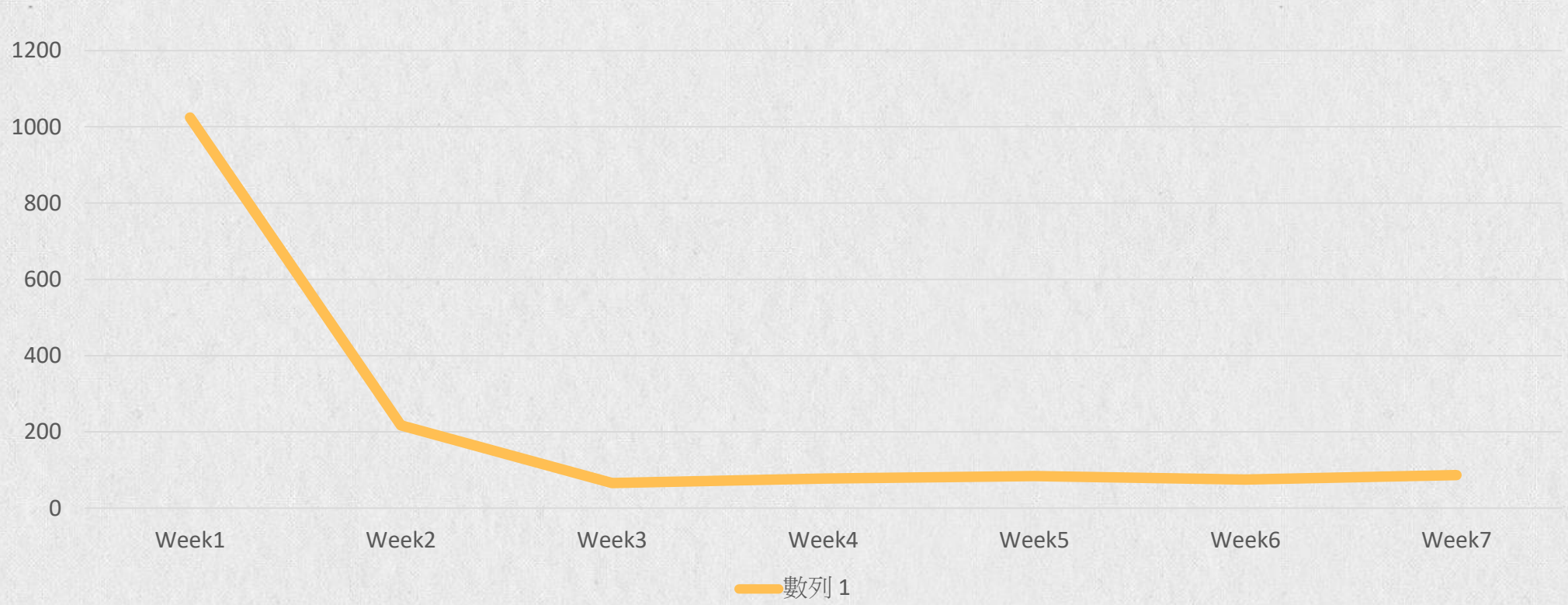
電影《黑豹》七周票房

.65
字

留
鍵

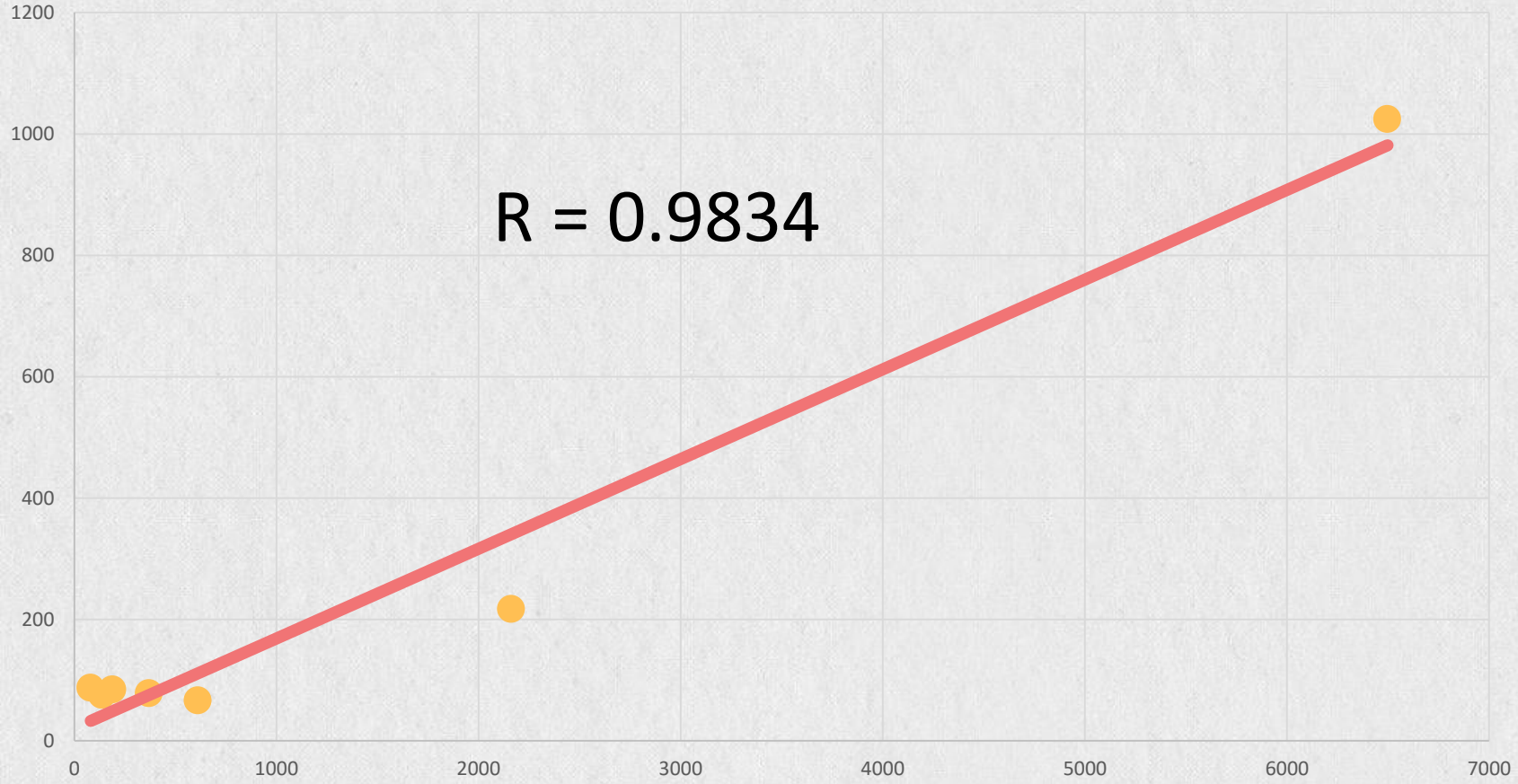
根據每周票房與好評關鍵字多寡變化做相關圖

4 好評關鍵字七週變化

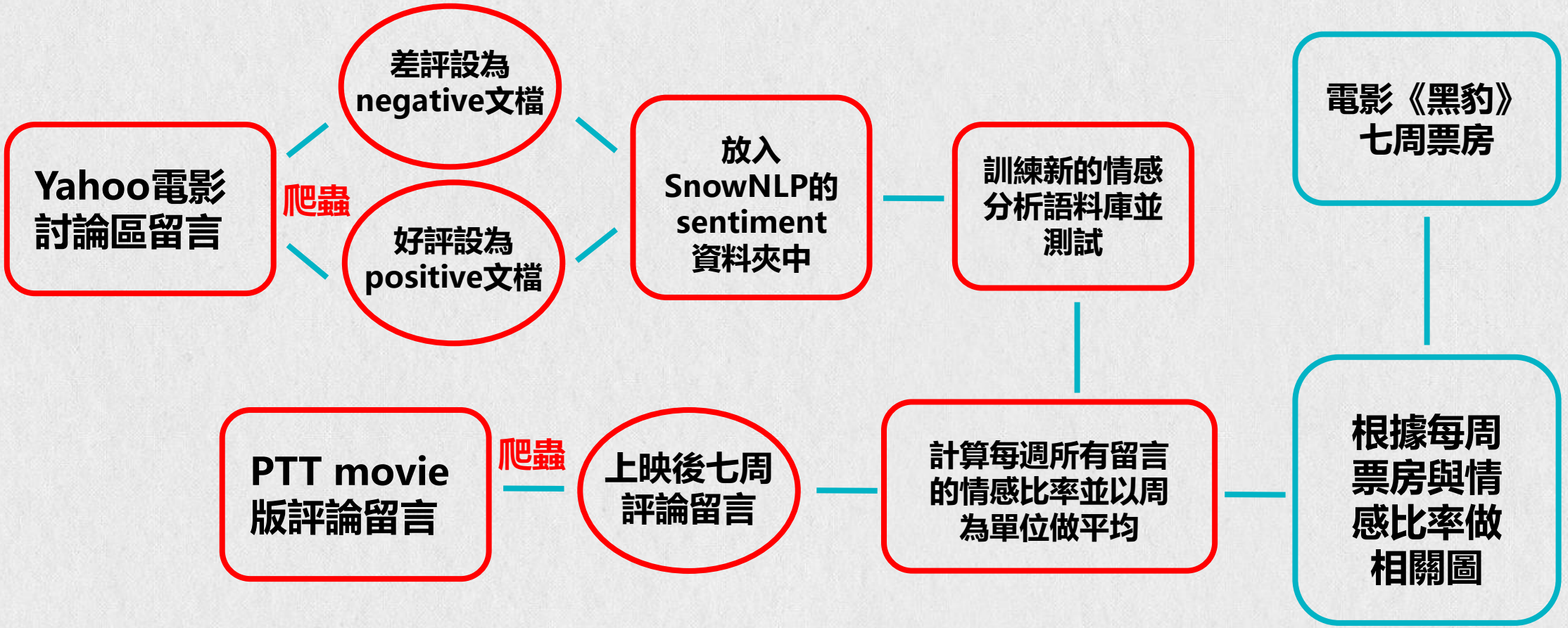

1200
1000
800
600
400
200
0
Week1
Week2
Week3
Week4
Week5
Week6
Week7
數列 1

4 好評關鍵字與票房關係

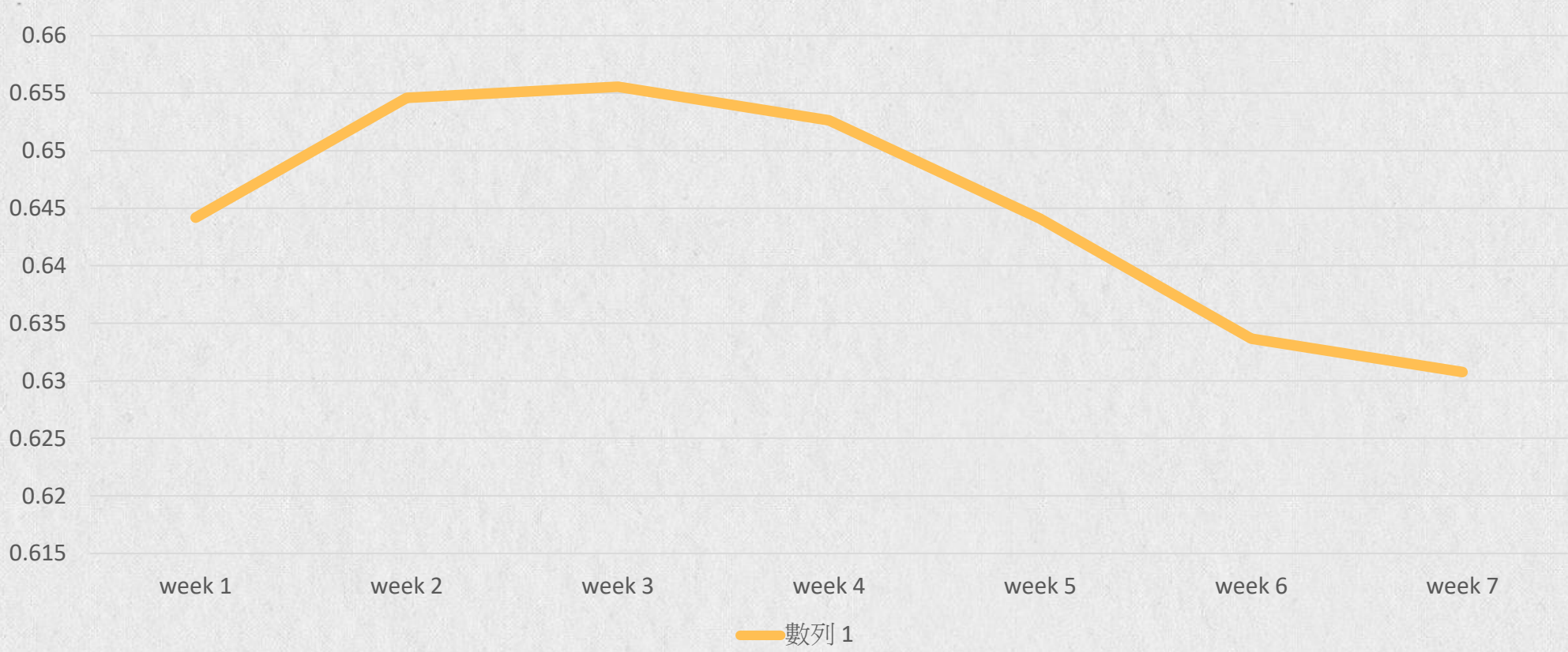
X = 票房 Y = 好評關鍵字



4 情感分析實作流程


Yahoo電影
討論區留言
爬蟲
差評設為
negative文檔
好評設為
positive文檔
放入
SnowNLP的
sentiment
資料夾中
訓練新的情感
分析語料庫並
測試
電影《黑豹》
七周票房
PTT movie
版評論留言
爬蟲
上映後七周
評論留言
計算每週所有留言
的情感比率並以周
為單位做平均
根據每周
票房與情
感比率做
相關圖

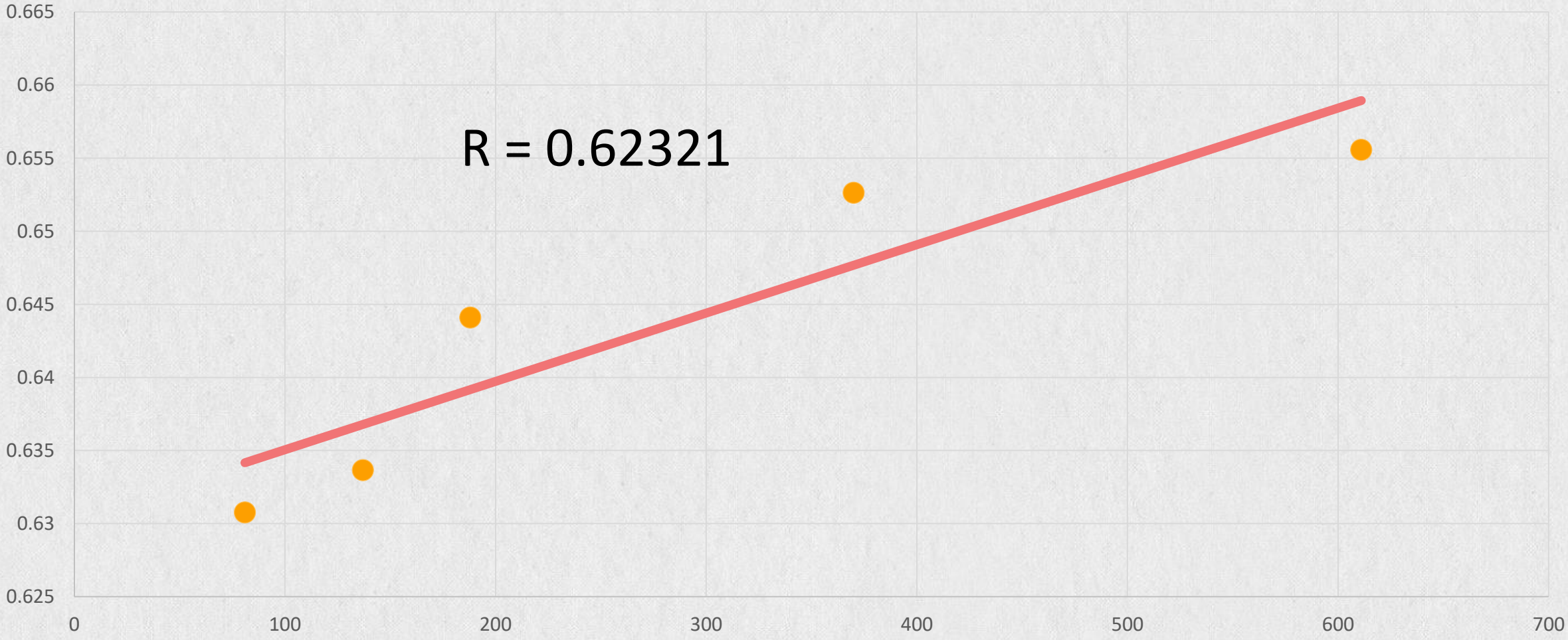
4 情感分析七週變化


0.66
0.655
0.65
0.645
0.64
0.635
0.63
0.625
0.62
0.615
week 1
week 2
week 3
week 4
week 5
week 6
week 7
數列 1

4 情感分析與票房關係


0.66
0.655
0.65
0.645
0.64
0.635
0.63
0.625
0
1000
2000
3000
4000
5000
6000
7000
R = 0.01223

4 情感分析與票房關係


R = 0.62321
0.665
0.66
0.655
0.65
0.645
0.64
0.635
0.63
0.625
0
100
200
300
400
500
600
700

4 結論&不足之處

首映的票房熱度
≠好評上升

1

關鍵詞減少≠好評
下降=>討論熱度?

2

樣本資料過少，
模型不夠完善

3

相關圖的資料
過少=>偏誤

4


Thank You