IIE Individual Research Project

Youtube 熱門影片趨勢分析

學生：109034540 胡心玫

指導老師：邱銘傳 教授

摘要

近年來有越來越多人投入 Youtuber 這個行業，為了成為成功的 Youtuber，勢必得先了解 Youtube 發燒影片的趨勢。首先我利用 Kaggle 的公開訓練資料集—Trending YouTube Video Statistics 來觀察有哪些變數是影響點閱數的重要因素，發現 likes 和 views 的歷史資料是影響點閱數的重要因素，LSTM 適合用來解決時間序列問題，因此本實驗利用 LSTM 模型來進行 Youtube 熱門影片趨勢分析，希望可以透過此研究幫助 Youtuber 決定他們的經營策略。

關鍵字：Youtube、點閱數、深度學習、長短期記憶 模型(LSTM)

目錄

壹、 研究動機與目的

近年來有越來越多人投入 Youtuber 這個行業，為了成為成功的 Youtuber，勢必得先了解 Youtube 發燒影片趨勢。YouTube 發燒影片列表中表現最佳的通常是音樂 MV 以及隨機爆紅的影片。根據《Variety》雜誌的說法：「要確定當年最熱門的影片，YouTube 會綜合考慮多種因素，包括衡量用戶互動、觀看次數、分享次數、評論和喜歡的次數。請注意，整年度觀看次數最多的影片並不等於就是當年最熱門影片」。我們可以透過此研究幫助 Youtuber 決定他們的經營策略。

以下將透過 5W1H 手法進行問題定義分析：

What?	Youtube 熱門影片趨勢分析
When?	2017-2018
Who?	Youtuber
Where?	美國
Why?	Youtuber 成為熱門職業，想要成為成功 Youtuber 勢必得先了解 Youtube 發燒影片趨勢。
How?	資料預處理、相關性分析與資料可視化、LSTM

貳、 文獻探討

一、長短期記憶網路(Long Short Term Memory Network, LSTM)

LSTM 是遞歸神經網路 (Recurrent Neural Network, RNN)的其中一種模型。RNN 主要是用來解決時間序列的問題， 一般的 RNN 透過將 Hidden layer 的 output 存在 Memory 裡，當下次 input 資料進去時，會同時考慮上一次存在 Memory 的值進行計算。

但是一般的 RNN 在長期記憶的表現並沒有很好，因此有學者研發了 LSTM 用來改善 RNN 在長期記憶的不足。

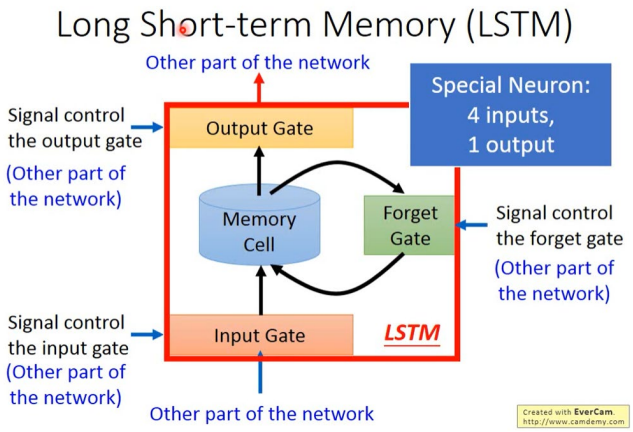

圖 1 LSTM 簡易模型(截圖自李宏毅老師投影片)

LSTM 主要由四個 Component 組成，分別是輸入閥(Input Gate)、輸出閥(Output Gate)、記憶單元(Memory Cell)以及遺忘閥(Forget Gate)。

1. Input Gate: 當將 feature 輸入時，input gate 會去控制是否將這次的值輸入
2. Memory Cell: 將計算出的值儲存起來，以利下個階段拿出來使用
3. Output Gate: 控制是否將這次計算出來的值 output
4. Forget Gate: 是否將 Memory 清掉(format)

※是否將這次計算出來的值 output，以及是否將 Memory 清掉(format)，可透過神經網路學習。

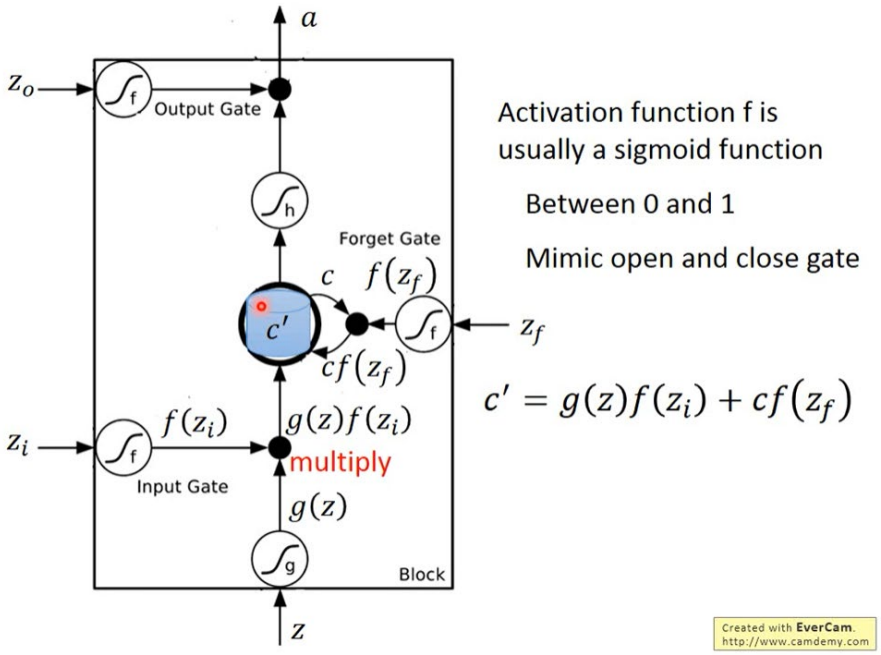

圖 2 LSTM 數學模型(截圖自李宏毅老師投影片)

如圖 2 所示，當 input 資料進去可表示為g(z)，而 input gate 則使用$f(z_i)$，一般來說 Activation function f 會使用 sigmoid function，因爲要表示開啟門的機率。因此會把兩個值相乘 $g(z)f(z_i)$，就可以得出在 input gate 開啟的機率下，input 的值。接著，Memory cell，就會紀錄當下 input 值加上前一次 input 值並乘上 forget gate 的機率，看是否要遺忘前一次紀錄 [$c' = g(z)f(z_i) + cf(z_f)$]。

參、 研究方法

本研究使用 Kaggle 的公開訓練資料集—Trending YouTube Video Statistics 進行分析與深度學習模型訓練，內含多個國家及地區的 Youtbue 熱門影片資料(包括影片 ID、標題、上榜日期、點閱數、喜歡/不喜歡次數、描述等欄位)，並於 python 環境中利用 keras 及多種套件，建構長短期記憶模型，對標準化後資料的進行訓練及預測影片的點閱數。

肆、　個案研究與實作

一、資料前處理

1.　資料集描述

Trending YouTube Video Statistics 內含多個國家及地區的 Youtbue 熱門影片資料，我選擇美國的資料進行分析，資料包含 json 檔及 csv 檔，json 檔包含影片 category_id 對應的 category_title 及 country_code，經過處理後可得 category_id 對應的 category_title 及 country_code 如下圖所示：

category_id	category_title	country_code
1	Film & Animation	US
2	Autos & Vehicles	US
10	Music	US
15	Pets & Animals	US
17	Sports	US
18	Short Movies	US
19	Travel & Events	US
20	Gaming	US
21	Videoblogging	US
22	People & Blogs	US
23	Comedy	US
24	Entertainment	US
25	News & Politics	US
26	Howto & Style	US
27	Education	US
28	Science & Technology	US
29	Nonprofits & Activism	US
30	Movies	US
31	Anime/Animation	US
32	Action/Adventure	US
33	Classics	US
34	Comedy	US
35	Documentary	US
36	Drama	US
37	Family	US
38	Foreign	US
39	Horror	US
40	Sci-Fi/Fantasy	US
41	Thriller	US
42	Shorts	US
43	Shows	US
44	Trailers	US

圖 3 category id 對應標題

csv 檔共 40949 筆資料，包含影片 ID、標題、上榜日期、點閱數、喜歡/不喜歡次數、描述等以下 16 種欄位：

1. **Video_id:** identification code for the YouTube Video
2. **Trending_date:** Date on which the Video was Trending
3. **Title:** Title of the YouTube Video
4. **Channel_title:** Title of the YouTube Channel uploading the Video
5. **Category_id:** Unique ID of the Video's Category (e.g. Entertainment, Music, Sports)
6. **Publish_time:** Date and Time in which the video was published
7. **Tags:** Hashtags added to the Video to make it easier to find by the public
8. **Views:** Number of Views the Video Obtained by the time the dataset was downloaded
9. **Likes:** Number of Likes the Video Obtained by the time the dataset was downloaded
10. **Dislikes:** Number of DisLikes the Video Obtained by the time the dataset was downloaded
11. **Comment_count:** Number of Comments the Video Obtained by the time the dataset was downloaded
12. **Thumbnail_link:** Thumbnail link of the Video (image representing a link)
13. **Comments_disabled:** Dummy Variable indicating whether the comments were disabled on the video
14. **Ratings_disabled:** Dummy Variable indicating whether the ratings were disabled on the video
15. **Video_error_or_removed:** Dummy Variable indicating the presence of a video error or removal of the video
16. **Description:** a piece of metadata that helps YouTube understand the content of a video. Well optimized descriptions can lead to higher rankings in YouTube search.

圖 4 欄位說明

由於 csv 的原始碼是以 ASCII 編碼，使用 Excel 打開時會產生亂碼，因此先利用 Notepad++將 ASCII 轉為 UTF-8-BOM，即可解決亂碼問題。

2. 刪除、補值及標準化

利用 Excel 確認有無缺值，將重複的資料列刪除，計算 tag 的數量，將日期呈現方式標準化，最後輸出新的 csv 檔，如下圖所示。

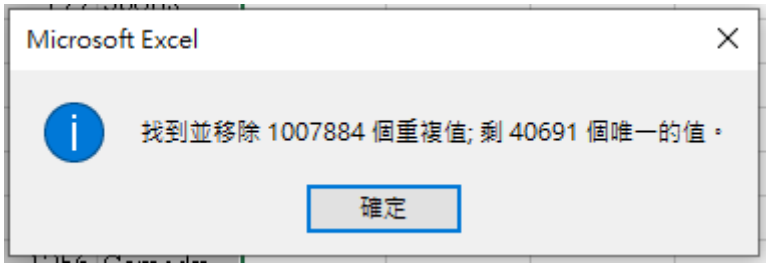

圖 5 刪除重複值

3. 相關性分析與資料可視化

首先將明顯不相干的欄位：thumbnail_link（縮圖連結）、comments_disabled（是否允許評論）、ratings_disabled（是否允許評分）、video_error_or_removed（影片是否損壞或移除）刪除。

	views
views	1
likes	0.850315
dislikes	0.472267
comment_count	0.619633
tags num	-0.02918

表 1 連續型變數與 views 之相關係數

為了挑選適合的特徵變數來預測 views，因此對連續型變數及 views 進行相

關性分析，發現 likes 和 views 的相關係數大於 0.8，為高度相關，故建議納入訓練模型之特徵；comment_count 和 views 的相關係數介於 0.5 至 0.8 之間，為中度相關；dislikes 和 views 的相關係數介於 0.3 至 0.5 之間，為低度相關；tags_num 和 views 的相關係數絕對值小於 0.3，故為不相關。分析結果符合我們的直覺，觀看數高的影片受到使用者的喜愛(likes)、使用者偏向在喜歡的影片下評論。因為 views 為時間序列資料，因此將 views 的過去資料也納入訓練模型之特徵。

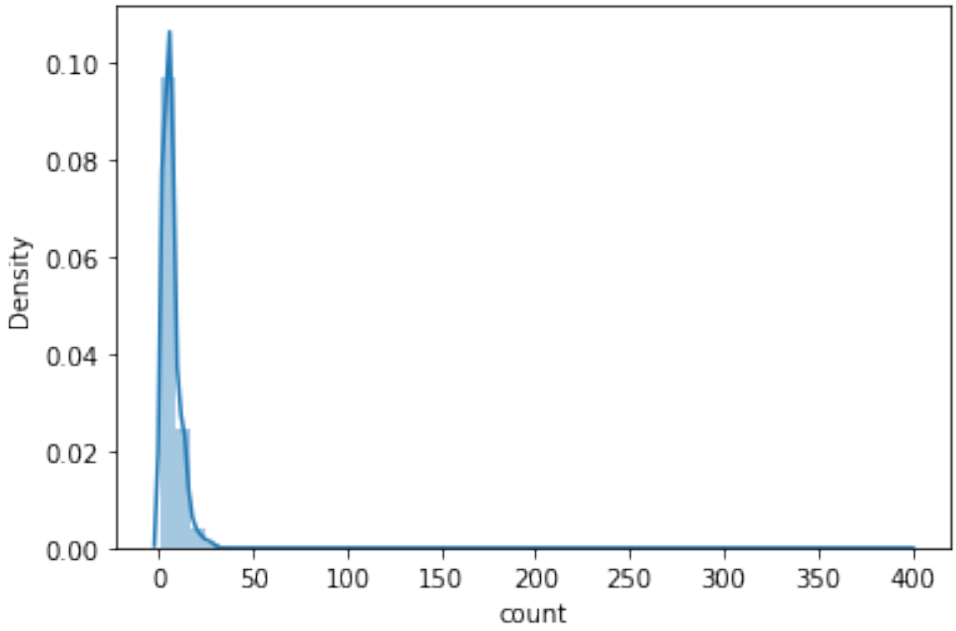

圖 6 連續上榜天數分布圖

	count
count	6282.000000
mean	6.518465
std	6.771645
min	1.000000
25%	3.000000
50%	6.000000
75%	8.000000
max	397.000000

表 2 連續上榜天數統計

超過 75%的熱門影片會連續上榜 3 天以上，中位數是 6 天，只有 25%的熱門影片會連續上榜 8 天以上，因此我們設定 time step=4 天。

非連續型的資料：category 和 publish_date 透過直方圖將資料可視化，從圖 7、表 3、表 4 可以發現 Entertainment 及 Music 類的影片點閱數最高；從圖 8 可以了解 publish_date 的分布狀況，但是 category 及 publish time 和 views 並無因果關係，故皆不納入訓練模型之特徵。

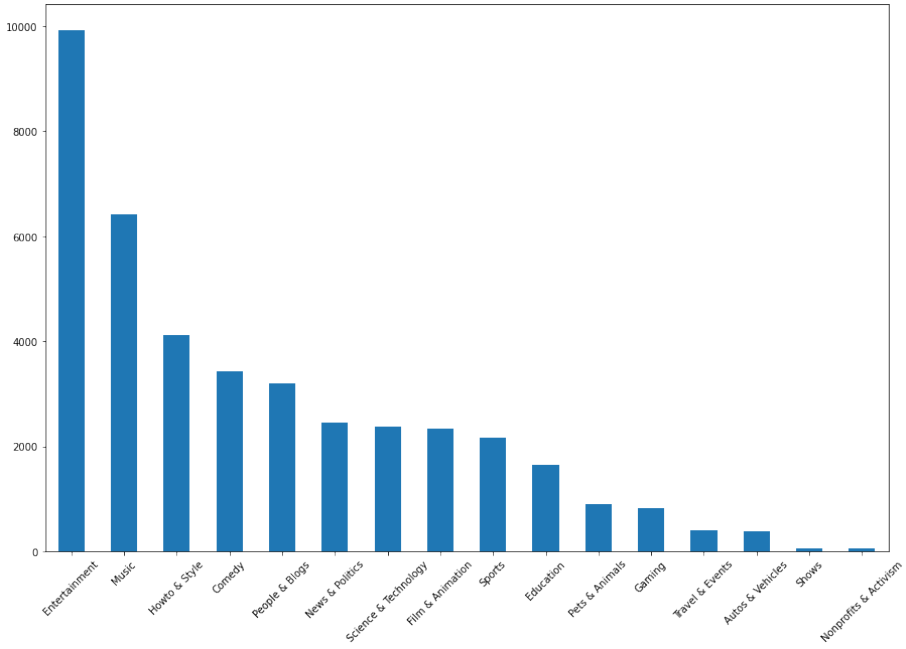

圖 7 category

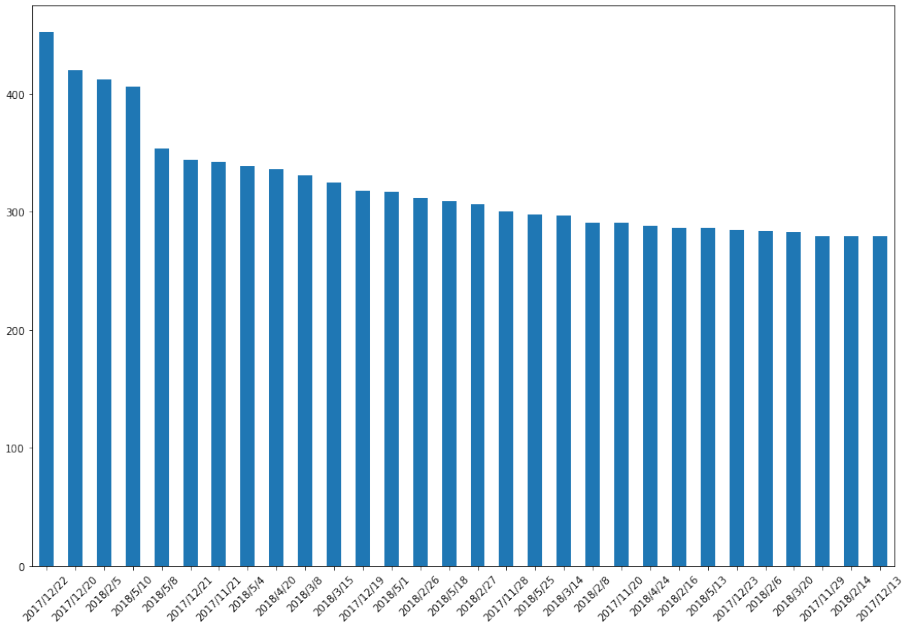

圖 8 publish date(取前 20 名)

count	313.000000
mean	130.000000
std	110.203803
min	1.000000
25%	6.000000
50%	120.000000
75%	223.000000
max	452.000000

表 3 publish date 分布情形

2017/12/22	452
2017/12/20	420
2018/2/5	412
2018/5/10	406
2018/5/8	354
...	
2010/4/21	1
2006/7/23	1

表 4 publish date 部分資料

經過相關性分析及資料可視化之後，最後僅挑選出兩項變數：likes 和 views 進行模型訓練。最後輸入模型的資料大小為(3942,4,2)。

二、模型與程式碼說明

本實驗中的 LSTM 模型的程式碼及架構如圖所示共有四層，透過加入 Dropout 層來減少模型過擬合發生的機率。

```
model = Sequential()
model.add(LSTM(64, return_sequences=True,input_shape=(9, 2),activation='tanh'))
model.add(Dropout(0.2))
model.add(LSTM(128,return_sequences=False,activation='tanh'))
model.add(Dense(1, activation='linear'))#1 is output

optimizer = RMSprop(lr=0.005)
model.compile(loss='mean_squared_error', optimizer=optimizer)
model.summary()
```

圖 9 LSTM 模型程式碼

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_8 (LSTM)                (None, 9, 64)             17152
_________________________________________________________________
dropout_4 (Dropout)          (None, 9, 64)             0
_________________________________________________________________
lstm_9 (LSTM)                (None, 128)               98816
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 129
=================================================================
Total params: 116,097
Trainable params: 116,097
Non-trainable params: 0
_________________________________________________________________
```

圖 10 LSTM 模型架構

三、參數調整與模型訓練成果

1. 輸入 likes 及 views(time step=4)

輸入 views(time step=4)及 likes 資料，訓練集的 loss 和驗證集的 val_loss 如所示，雖然偶有發生 val_loss>loss 的情形，但是當 val_loss 下降時，loss 適時上升，最後收斂到 0.02 以下，代表此模型確實可以持續調整有效避免過擬合發生，並且可以精準預測目標值。

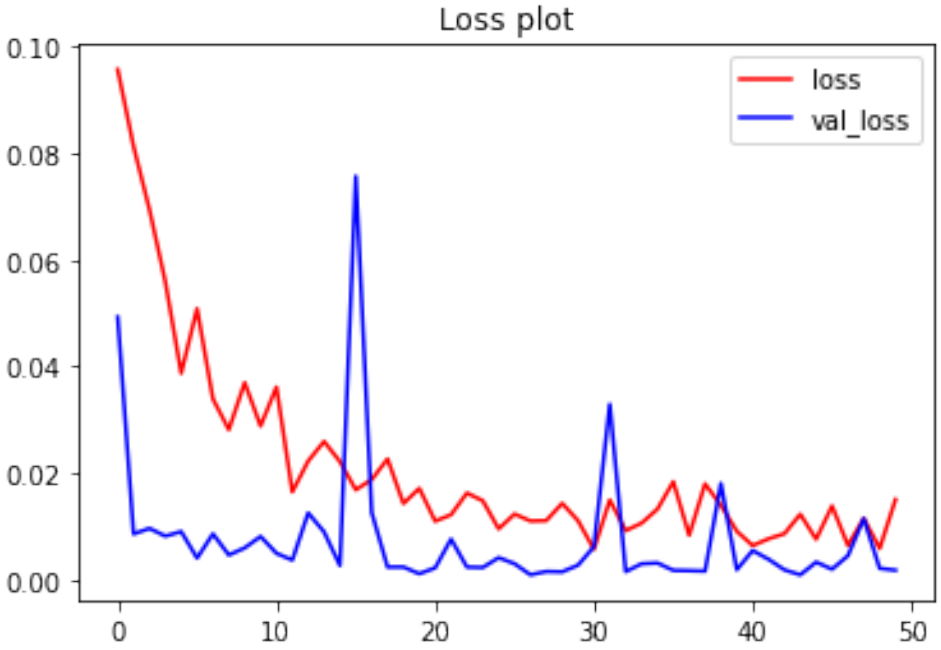

圖 11 Loss plot(輸入 likes 及 views(time step=4)

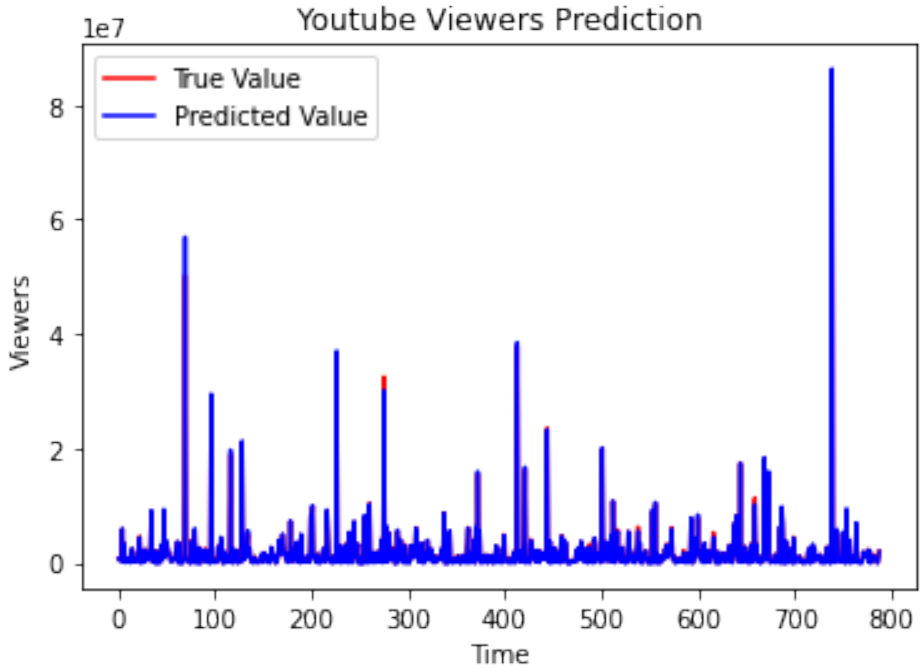

圖 12 Prediction(輸入 likes 及 views(time step=4)

	y_test	predict
count	7.890000e+02	7.890000e+02
mean	1.908541e+06	1.859762e+06
std	4.847343e+06	5.019882e+06
min	1.464000e+03	-1.194840e+05
25%	2.829120e+05	1.767588e+05
50%	7.323490e+05	6.458047e+05
75%	1.876178e+06	1.852529e+06
max	8.509207e+07	8.632411e+07

表 5 y_test 與 predict 分布(輸入 likes 及 views(time step=4)

2. 輸入 likes 及 views(time step=9)

接下來我嘗試調整參數讓模型可以更精確預測 Views，輸入 likes 及 views(time stp=9)之結果如圖所示：

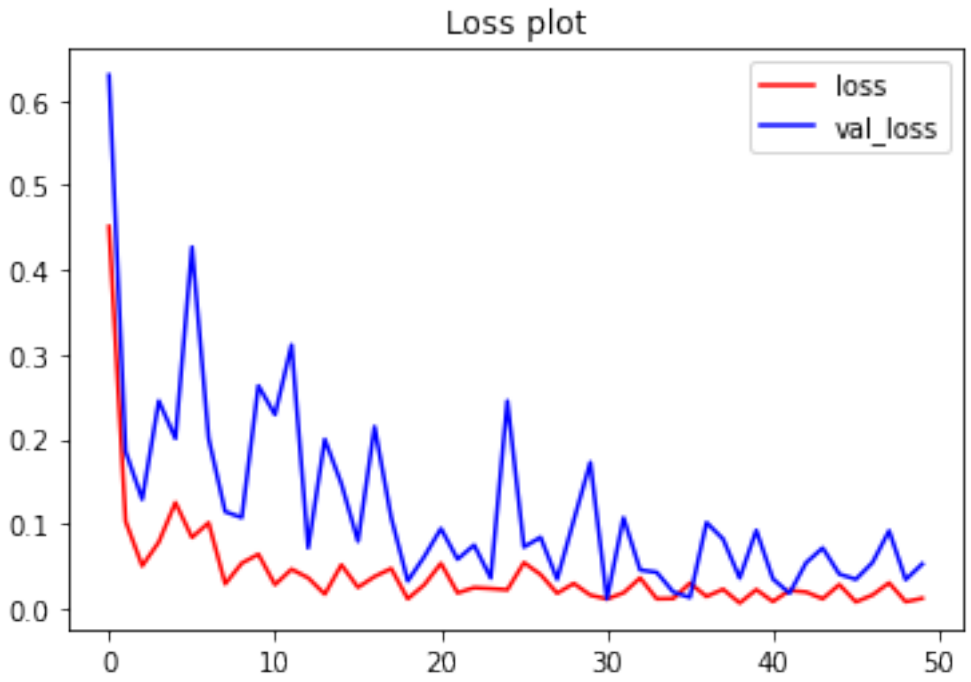

圖 13 Loss plot(輸入 likes 及 views(time step=9)

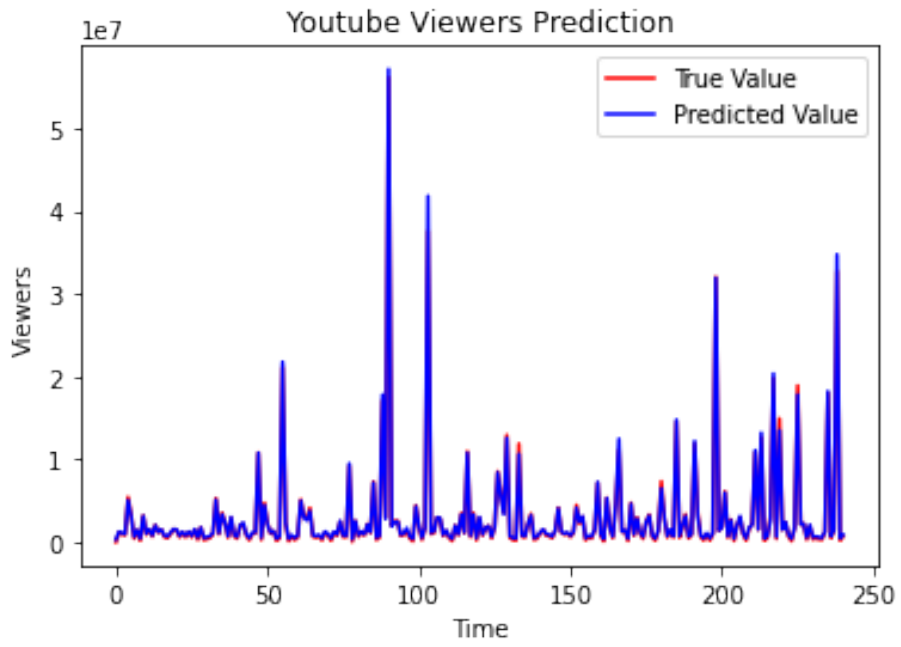

圖 14 Prediction(輸入 likes 及 views(time step=9)

	y_test	predict
count	2.410000e+02	2.410000e+02
mean	3.027963e+06	3.147387e+06
std	6.093774e+06	6.233731e+06
min	3.091800e+04	3.434237e+05
25%	5.707290e+05	7.674609e+05
50%	1.151425e+06	1.258726e+06
75%	2.580657e+06	2.570490e+06
max	5.611196e+0	5.712340e+07

表 6 y_test 與 predict 分布(輸入 likes 及 views(time step=9)

3. 輸入 views(time step=4)

只輸入 views(time stp=4)之結果如圖所示：

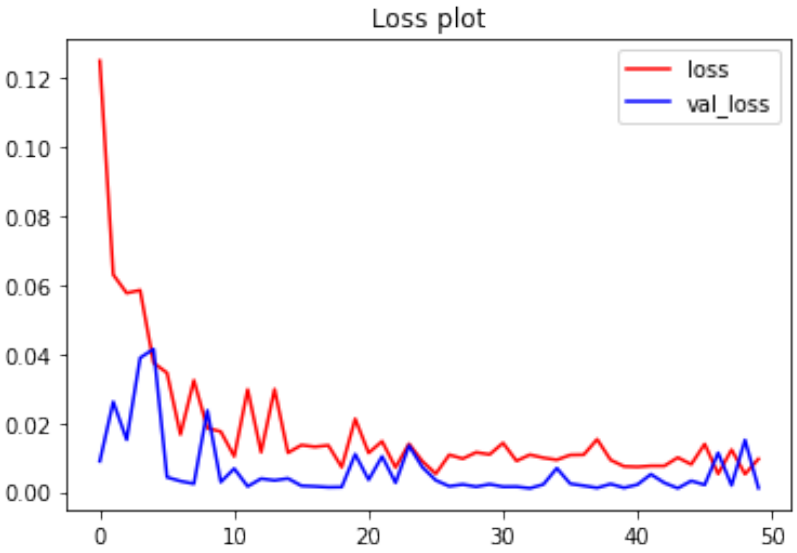

圖 15 Loss plot(輸入 views(time step=4)

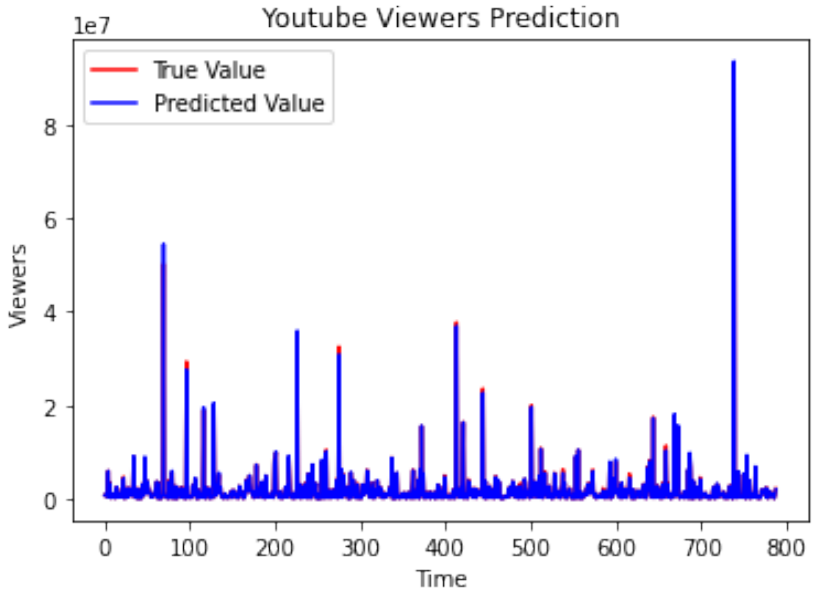

圖 16 Prediction(輸入 views(time step=4)

	y_test	predict
count	7.890000e+02	7.890000e+02
mean	1.908541e+06	1.902993e+06
std	4.847343e+06	5.070968e+06
min	1.464000e+03	1.935354e+04
25%	2.829120e+05	2.875418e+05
50%	7.323490e+05	7.105782e+05
75%	1.876178e+06	1.840289e+06
max	8.509207e+07	9.360760e+07

表 7 y_test 與 predict 分布(輸入 views(time step=4)

4. 輸入 views(time step=9)

time step 由 4 天增加至 9 天之結果如圖所示：

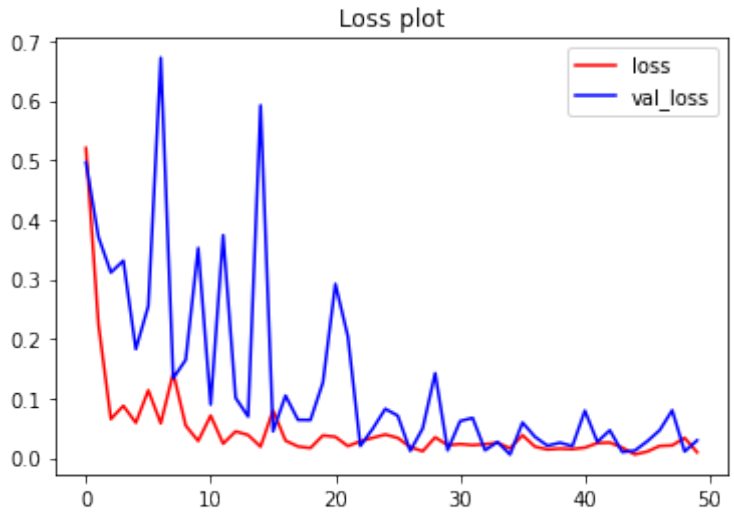

圖 17 Loss plot(輸入 views(time step=9)

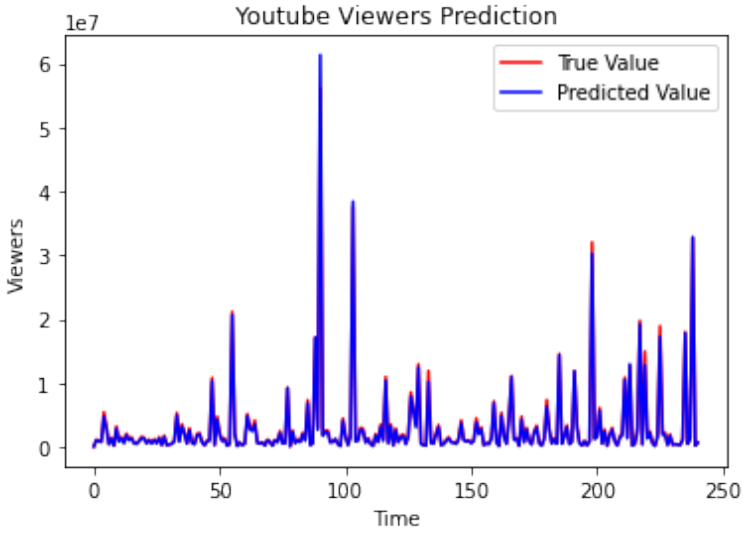

圖 18 Prediction(輸入 views(time step=9)

	y_test	predict
count	2.410000e+02	2.410000e+02
mean	3.027963e+06	2.852364e+06
std	6.093774e+06	6.215746e+06
min	3.091800e+04	6.870095e+04
25%	5.707290e+05	5.092782e+05
50%	1.151425e+06	9.913957e+05
75%	2.580657e+06	2.294445e+06
max	5.611196e+07	6.133464e+07

表 8 y_test 與 predict 分布(輸入 views(time step=9)

伍、 結論

	Train Score(RMSE)	Val Score(RMSE)
輸入 likes 及 views(time step=4)	367757.64	338026.47
輸入 likes 及 views(time step=9)	956439.82	408942.70
輸入 views(time step=4)	484946.44	393396.12
輸入 views(time step=9)	815038.07	480950.99

表 9 Train Score 與 Val Score

以上的參數組合中，表現最好的是「輸入 likes 及 views(time step=4)」，推測是因為連續上榜超過四天的熱門影片數較多，所以可以訓練的資料較多，因此預測較為精準。

總結以上，經由資料前處理(包含觀察缺失值、相關性分析與資料可視化)等手法找出特徵值作為訓練模型變數，最終挑選出 2 項變數作為訓練模型之因子，而透過 keras 深度學習模型建立，並透過實驗針對輸入變數、time step 等參數進行調整，最終得出良好的訓練模型。

從本次的 Project 中，可以發現適當的篩選特徵變數及資料前處理手法對於後續的建立 Model 及辨識準確率極為重要，而對於深度學習各項參數的理解與如何適當調整參數，更是報告帶給我們的學習重點，所以我們認為可以持續建立不同深度學習模型並依資料處後之特性來調整參數或是運用於其他 Youtube 預測資料集(例如，台灣 Youtube 資料等)來實作驗證，以此延伸本次研究內容與實際落地。

陸、 參考資料

一、 LSTM Prediction on Trending YouTube Videos Views

二、 YouTube 热门视频榜单 Excel 数据分析

三、 LSTM 深度學習 股價預測

四、 ML Lecture 21-1: Recurrent Neural Network (Part I)