社群媒體分析期末專案

Youtube 合作對象推薦系統

組別：第一組

組員：魏靖軒(R11722025)、江子涵(R12922082)、葉冠宏(R11943113)、許聖慧(R12H41006)

分工

- 葉冠宏 (R11943113)
  口頭簡報及製作、書面報告、架構四、資料集蒐集
- 魏靖軒 (R11722025)
  口頭簡報及製作、書面報告、架構三
- 江子涵 (R12922082)
  口頭簡報及製作、書面報告、架構五
- 許聖慧 (R12H41006)
  口頭簡報及製作、書面報告、架構二

研究動機

- 透過 Youtube 的演算法，觀眾會被基於過去的瀏覽紀錄推播各種影片，可能是常看的 Youtube 頻道，也可能是內容符合觀眾喜好的新 YouTuber 影片
- 那有沒有專門為創作者(Youtuber)設計的演算法呢？
- Youtuber 彼此合作常能為影片帶來更高的流量，但要怎麼找到合適的合作對象？又應該合作甚麼類型的影片能達到最高效益？

希望透過蒐集過往 Youtuber 的頻道影片資料、合作資料，並設計一些演算法來推薦新的合作組合以及適合合作的影片類型。

資料蒐集

1. 爬取工具(都有諸多限制)

Youtube Data Api:channelId、訂閱人數
Scrapetube:影片標題、影片發行日期、觀看人數
ChatGPT:影片類別判斷

2. 特徵爬取

影片個人風格:星座=>人工標記
影片類型:'travel','unbox','challenge','culture','game','personal','talk','education','music'=>chatgpt
觀看受眾的年齡分布:資料集中年齡大多介於20歲至30歲=>人工標記
觀看次數:=>scrapetube
訂閱人數:=>youtube data api
性別:男性、女性、團體=>人工標記
爭議度:道歉影片發布的時間點作為爭議的起點。影片發布的時間點和爭議發生的時間點相隔越遠，爭議度越少=>scrapetube
共同評論觀眾的向量:由評論者的id判斷共同的target audience=>youtube data api

資料蒐集

3. Youtuber節點選取
選定兩個較常和別人合作的 youtuber作為首兩位youtuber節點。其影片中有合作的對象均作為節點的候選清單。當我們審視候選清單中的 youtuber時，其合作對象中不在候選清單中的節點將不再新加入候選清單中，並標記該影片為跳過。
此外，藝人、因為爭議已刪除頻道大部分 內容的節點亦不納入。
最後，有些youtuber有開設多個子頻道，我們也將那些頻道合併算做同一節點。

問題:independent set

4. 標記合作對象
影片未和其他youtuber合作:標記為-1
影片中出現多位youtuber合作對象:以逗號隔開每位youtuber節點id

問題:
有些合作對象標記在影片中
有些合作對象不是標記在 @, feat之後
有些合作對象有多個綽號，或是其名字和一般常見物品重疊，例如 :牛排、胡椒

目標任務

1. 兩兩 Youtuber 是否適合合作？
2. 若適合合作，該合作甚麼影片類型？
3. 合作後的成功度(影片觀看量)？

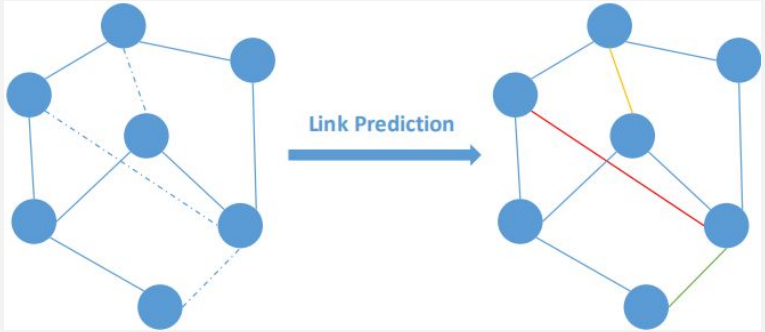


image source

方法

	任務1	任務1+2	任務1+3
Feature-based		架構四	
Graph-based	架構二	架構三	
Feature-based + Graph-based			架構五

註: 架構一為 proposal 中提出之方法，但後續認為不適用，因此在此次報告中不詳細討論

模型效能評估方式

- 為設計一個**與真實世界情況相近**的有效驗證方法，我們將所有合作資料以一個**標準時間線**切分。以時間線以前的關係作為訓練資料(如果是feature-based的，僅考慮有合作過的youtuber組合作為訓練)，時間線以前尚未合作的組合作為測試資料。

- 我們爬取到的資料範圍從2009-05-13 至 2024-05-04

- 選取 **2021-12-18** 作為標準時間線
  - 約切分成 **80% 訓練 / 20% 測試**
  - 已確保移除測試資料中的link 後仍只有一個 connected component

模型效能評估方式

真實 \ 預測	標準時間線後 仍沒有合作	標準時間線後 才開始有合作
標準時間線後 仍沒有合作	A	B
標準時間線後 才開始有合作	C	D

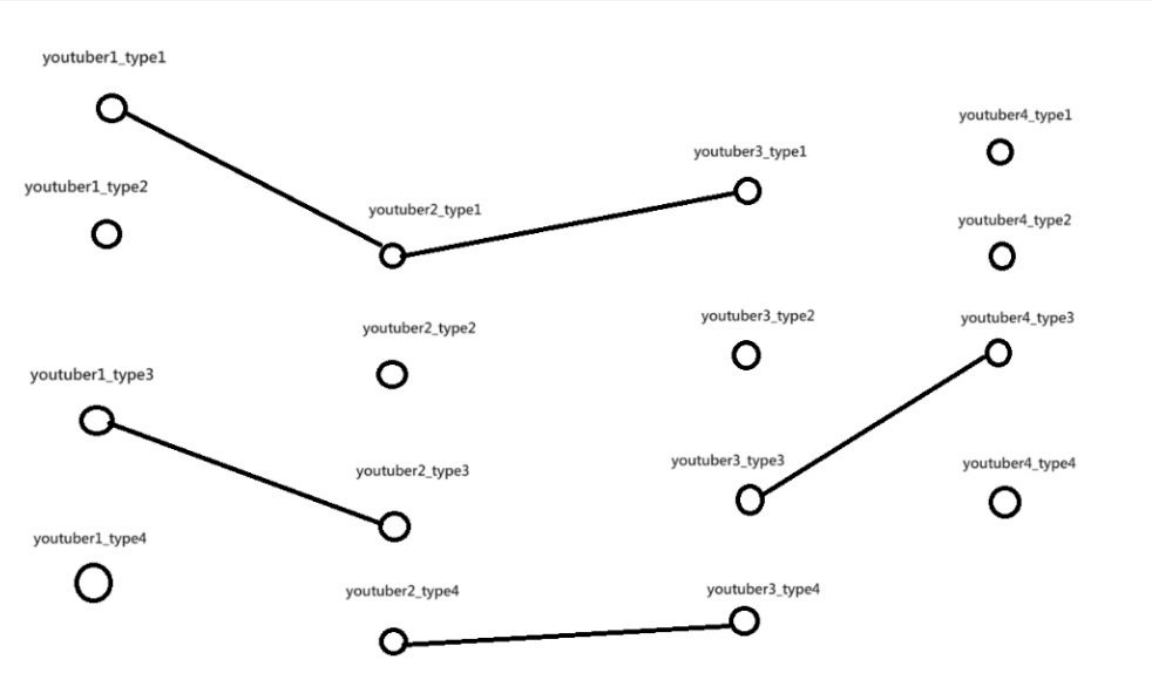
- Accuracy = (A+D) / (A+B+C+D)
- Recall = D / (C+D)
- F1 score = (2*D) / (2*D+B+C)
- ROC-AUC

各架構間著重比較 **Recall** 與 **ROC-AUC**

- Recall
  - 優點:可以判斷有連線的是否都有被找到
  - 缺點:只看這項數據可能導致模型直接全部預測有合作
- ROC-AUC
  - 優點:解決 recall 的問題，也不會受到 threshold 選擇的影響
  - 缺點:沒有合作不代表不適合合作，可能只是在目前資料蒐集範圍中還沒有合作，未來仍可能合作。由於不合作和合作的資料量分布是不平均的，如果沒有加以處理，可能會造成模型均預測不合作以獲得較好的數據。

架構方法與實驗結果

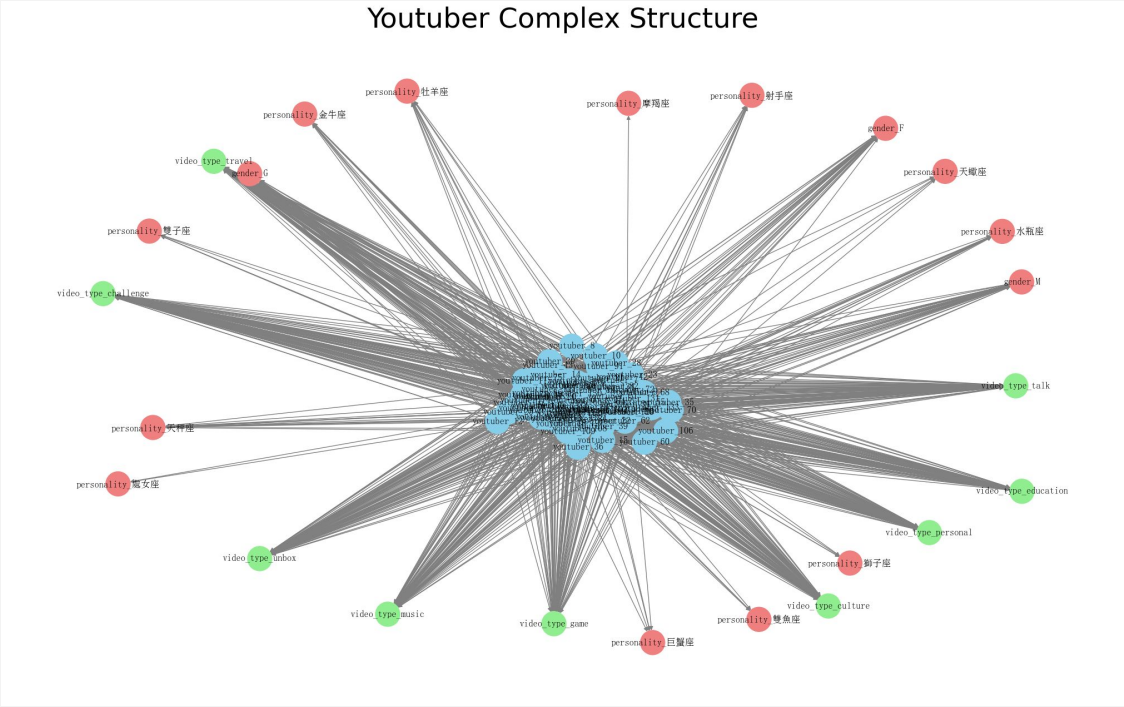
架構一(期中提案之架構)


youtuber1_type1
youtuber1_type2
youtuber2_type1
youtuber3_type1
youtuber4_type1
youtuber4_type2
youtuber2_type2
youtuber3_type2
youtuber4_type3
youtuber1_type3
youtuber2_type3
youtuber3_type3
youtuber4_type4
youtuber1_type4
youtuber2_type4
youtuber3_type4

架構二(預測是否合作)

節點

1. YouTuber
2. 影片類型
3. 性別、星座



架構二(預測是否合作)

邊權重設定

YouTuber$_i$ 與 YouTuber$_j$ 之間

$$權重(W_{i,j}): S_{i,j} * C_{i,j} * \sum_{YouTuber\ j\ Appear} L_{i,j,t,y} + S_{j,i}$$

其中,

a. $訂閱指標(S_{i,j}): S_{i,j} = \frac{YouTuber_j訂閱數}{YouTuber_i訂閱數+YouTuber_j訂閱數}$

b. $合作次數(C_{i,j}): C_{i,j} = \frac{1}{YouTuber_i頻道中YouTuber_j出現次數}$

c. $觀看表現(L_{i,j,t,y}): L_{i,j,t,y} = \frac{YouTuber_i頻道中YouTuber_j出現影片觀看量}{YouTuber_i頻道中影片Type_t於Year_y平均觀看量}$

架構二(預測是否合作)

邊權重設定

YouTuber$_i$ 與影片類型 Type$_t$ 之間

$$權重(W_{i,t})：W_{i,t} = \frac{YouTuber_i 中 Type_t 的平均觀看數}{YouTuber_i 的平均觀看數}$$

YouTuber 與性別、星座之間

如果 YouTuber 屬於該類別，則連線值為1，否則不連線。

架構二(預測是否合作)

使用方法:加權近似度演算法

基於同質性假設，使用**共同鄰居(Common Neighbors)、Jaccard係數、Adamic/Adar指數**三種相似度指標的改良，不僅考慮共同鄰居的存在，也加入與這些鄰居相連的邊的權重。

對於上述方法，我們將計算出的加權指數進行**Min-Max標準化**，將標準化後指標視作合作機率。

架構二(預測是否合作)

使用方法:圖卷積網絡(GCNs)

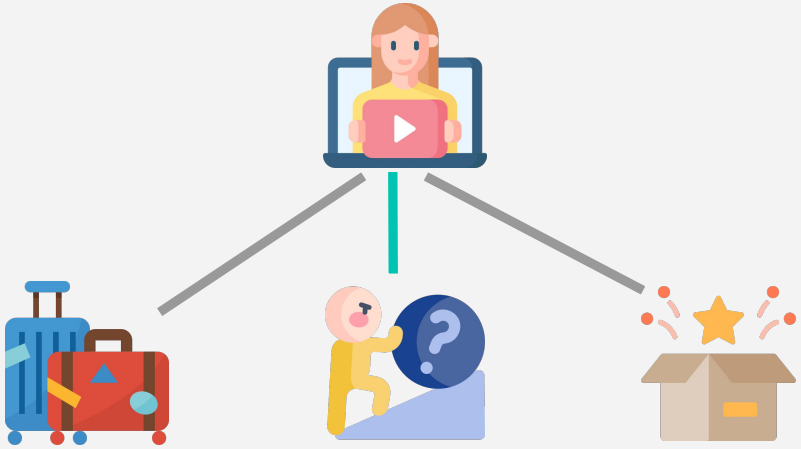
- **節點特徵的轉換**:將節點特徵(如YouTuber、影片類型、性別、星座)轉換為邊的權重，以此來捕捉各節點間的互動關係。
- **平衡訓練**:合成不存在的邊(負樣本)來進行訓練。
- **損失計算和優化**:使用二元交叉熵損失計算預測輸出和真實標籤之間的損失。

架構二(預測是否合作)

實驗結果

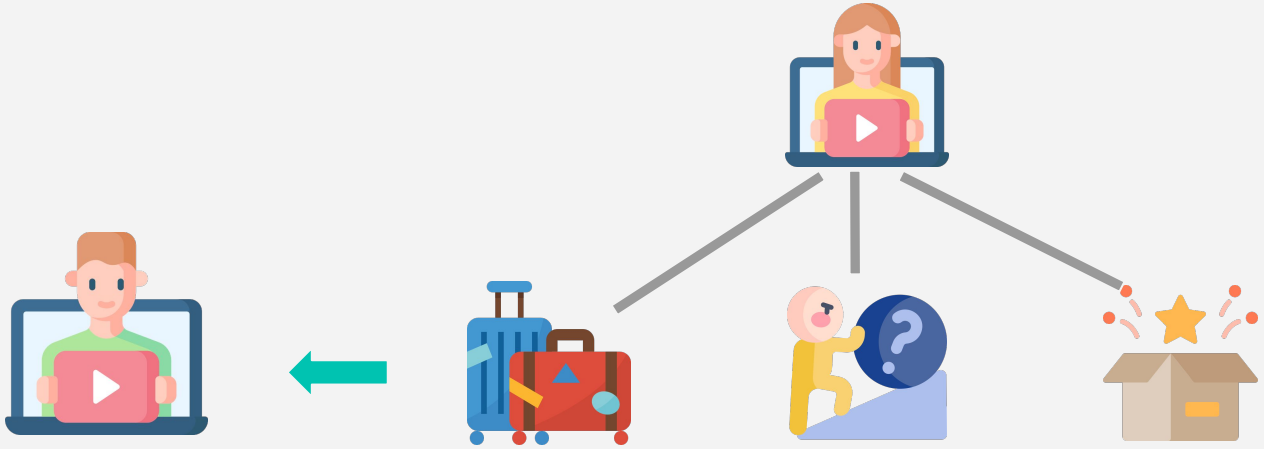
Metric	Weighted Common Neighbors (Scaled)	Weighted Jaccard Coefficient	Weighted Jaccard Coefficient (Scaled)	Weighted Adamic/Adar Index (Scaled)	GCN
Precision	0.0769	0.0404	0.0472	0.1275	**0.5385**
Recall	0.0683	0.1024	0.0488	0.0634	**0.5097**
AUC	0.50994	0.4507	0.4507	0.5458	**0.5812**

架構三(預測是否合作及合作影片類型)



$$W_{\text{類別 k 權重}} = \frac{\text{youtuber i 影片類型 k 的平均觀看數}}{\text{youtuber i 頻道的平均觀看數}}$$

架構三(預測是否合作及合作影片類型)



$$W_{\text{合作權重}} = \frac{1}{\text{類別 k 中 youtuber j 出現的數量}} \times \frac{\text{youtuber j 在類別 k 中的平均觀看數}}{\text{類別 k 的平均觀看數}}$$

架構三(預測是否合作及合作影片類型)

可能的連線類型

1. Youtuber 對 Youtuber (不考慮)
2. Topic 對 Youtuber
3. Youtuber 對 Topic
4. Topic 對 Topic
    a. Youtuber i topic 對 Youtuber j
    b. Youtuber i 對 Youtuber j topic

架構三(預測是否合作及合作影片類型)

預測結果: 預測Youtuber間是否會合作

	Weighted Jaccard Coefficient	Weighted Common Neighbor	Weighted Adamic/Adar index
Precision	0.31	0.34	0.32
Recall	0.14	0.05	0.07
F1 Score	0.19	0.08	0.11
AUC	0.55	0.52	0.53

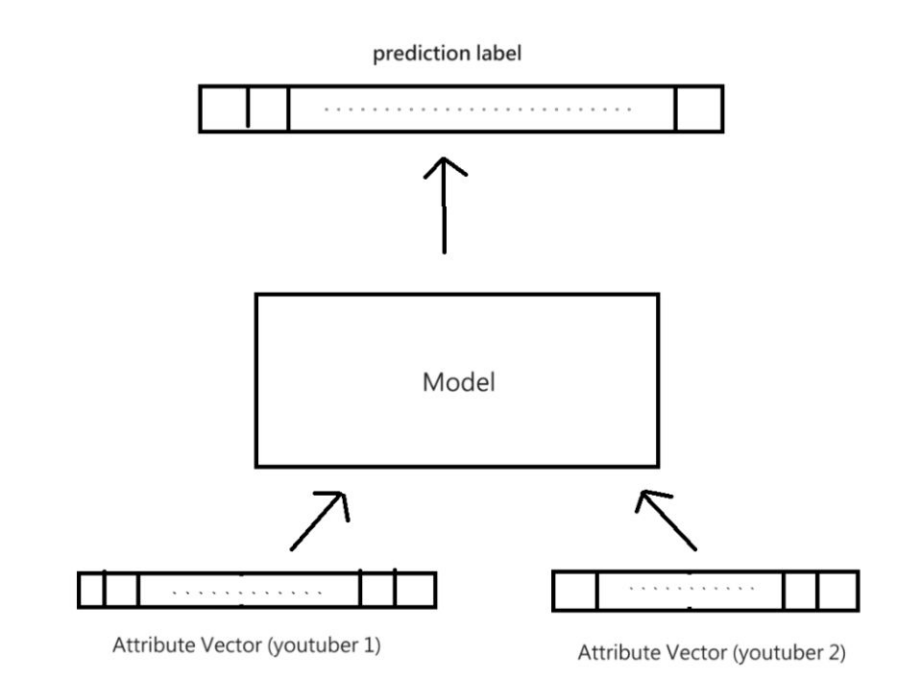
架構三(預測是否合作及合作影片類型)

預測結果: 預測是否會針對特定類別合作

	Weighted Jaccard Coefficient	Weighted Common Neighbor	Weighted Adamic/Adar index
Precision	0.13	0.15	0.13
Recall	0.26	0.09	0.12
F1 Score	0.17	0.11	0.12
AUC	0.61	0.54	0.55

架構四(預測是否合作及合作影片類型)

方法



1.特徵向量:性別、個人風格、和其他 YouTuber 之間的關係、每種影片類型的平均觀看量/1000萬、訂閱量/1000萬
2.餵入模型的向量:兩兩youtuber特徵向量+影片類別向量
3.神經網路:

```
class BinaryClassificationModel(nn.Module):
    def __init__(self):
        super(BinaryClassificationModel, self).__init__()
        self.fc1 = nn.Linear(2*featuredim+len(cat), 50)
        self.fc2 = nn.Linear(50, 20)
        self.fc3 = nn.Linear(20, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        x = self.sigmoid(x)
        return x
```

影片類型的建議值(video type k) ={ [(YouTuber i的訂閱量)/(YouTuber i的訂閱量 + YouTuber j的訂閱量)] x [(1/在YouTuber j的頻道中影片類別為k的有出現YouTuber i的影片數量)] x [$\sum_{YouTuber\ i\ Appear\ in\ video\ type\ k\ of\ YouTuber\ j}$ (在YouTuber j的頻道中該影片(屬於影片類別k)的觀看量/在YouTuber j 的頻道中屬於該影片年度影片類別k的平均觀看量)]}

+ {[(YouTuber j的訂閱量)/(YouTuber i的訂閱量 + YouTuber j的訂閱量)] x [(1/在YouTuber i的頻道中影片類別為k的有出現YouTuber j的影片數量)] x [$\sum_{YouTuber\ j\ Appear\ in\ video\ type\ k\ of\ YouTuber\ i}$ (在YouTuber i的頻道中該影片(屬於影片類別k)的觀看量/屬於該影片年度YouTuber i 頻道中該影片類別的平均觀看量)]}

註:如果兩者的合作影片僅在其中一個YouTuber的頻道中出現，則不需考慮訂閱量的加權權重。

標記為1:建議值>1
標記為0:建議值<1、未曾合作該影片類型

Loss function:(考量data imbalance)

$$l_{n,c} = -w_{n,c}\left[p_c y_{n,c} \cdot \log \sigma(x_{n,c}) + (1 - y_{n,c}) \cdot \log(1 - \sigma(x_{n,c}))\right]$$

架構四(預測是否合作及合作影片類型)

實驗結果

	兩兩是否合作某影片類型	兩兩是否合作
Accuracy	0.6619656429858091	0.14551206010280743
Recall	0.649402390438247	0.8823529411764706
AUC	0.655754060612282	0.4873824900880233
F1 score	0.7865206851236046	0.16479494262666672

架構五(預測是否合作及合作後的成功度)

方法

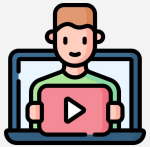
任務1: 預測是否合作
(link prediction)
0 / 1

任務3: 合作後的成功度
(weight prediction)
大於 0 的數值

任務1
Logistic Regression,
LightGBM

Model

任務3
Linear Regression,
LightGBM,
XGBoost



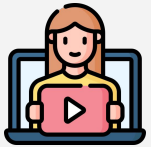
Node2Vec embed | Attributes

Youtuber a

add or concat

Node2Vec embed | Attributes

Youtuber b



架構五（預測是否合作及合作後的成功度）

方法 - weight 的定義

我們假設 Youtuber 最在意的是**影片觀看數**，因此選擇使用兩個 Youtuber 的**「所有合作影片之平均觀看量」**當作 weight，用來代表該合作的成功度。

$$\text{edge}(a, b) = \sum_{i \in \text{所有 a 與 b 的合作影片}} \text{影片 } i \text{ 的觀看數}$$

任務3: 預測合作後的成功度 (weight prediction)

給定 a, b → 預測 edge(a, b)

架構五(預測是否合作及合作後的成功度)

方法 - weight 的定義

考量到隨著時間增加，觀看量會隨著訂閱數的上升以及曝光時間長度增加而有所成長。因此我們須將原始觀看量作scaling。

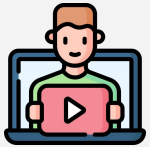
每部影片的正規化平均觀看量(viewCountAvgInYear)：

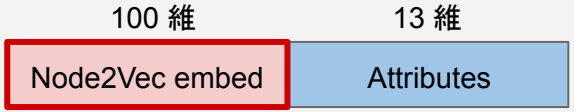
viewCountAvgInYear = 原始影片觀看量 ÷ 該影片所屬年度的該 Youtuber 所有影片平均觀看量

若 **viewCountAvgInYear > 1**，則代表該影片的觀看量在所屬年度是**高於平均**的，大於 1 越多代表觀看數越高、**影片成效越好**；反之，若 **viewCountAvgInYear < 1**，則代表該影片的觀看量在所屬年度是**低於平均的**，小於 1 越多則觀看數越低、**影片成效越差**。

架構五(預測是否合作及合作後的成功度)

方法 - Node2Vec embedding



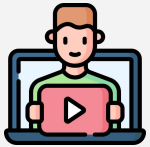


- 核心思想是透過**隨機游走(random walk)**
  來捕捉節點在圖中的結構特徵，並將這些特徵轉換為低維向量表示。Node2Vec 不考慮節點本身的標記或特徵，**只考慮整張圖中節點的結構**，因此可以將使用 Node2Vec 取得的 node embedding 用於各種下游任務中，而不用為每個任務重新設計與計算一次

- 使用訓練資料建構出 graph 後，訓練 Node2Vec 模型，設定每個節點得到 **100 維**的 node embedding

架構五(預測是否合作及合作後的成功度)

方法 - Attributes



- 訂閱數:原訂閱數 / 1000 萬
- 性別:男性 → 0, 女性 → 1, 團體 → 2
- 星座:0(摩羯座)～11(射手座)
- 年齡:(實際年齡 - 30) / 10
- 各影片類別的平均觀看量(共9 項, travel, unbox, challenge, culture, game, personal, talk, education, music)

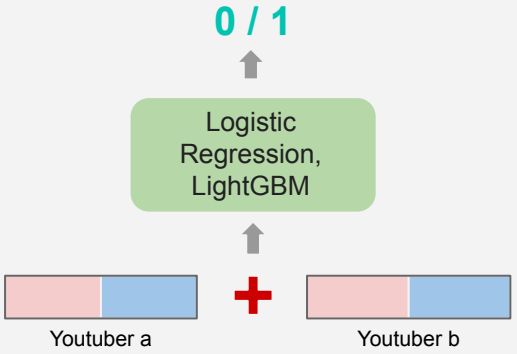
類別 k 的正規化平均觀看量 = $\frac{\sum_{video\ i\ \epsilon\ type\ k} viewCountAvgYear(i)}{|N|}$,

|N| = 屬於類別 k 的影片總數

架構五(預測是否合作及合作後的成功度)

實驗結果 - 預測是否合作(任務一)

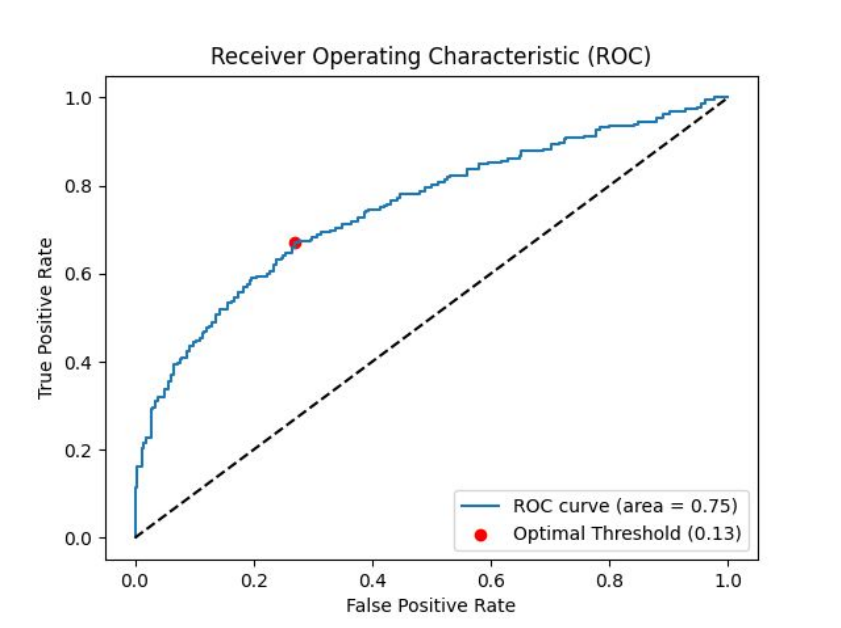
- LightGBM 最穩定
- 最佳組合: Mix (Add) + LightGBM
- 只使用 Node2Vec 或只用 Attributes 差不多, 但**兩者皆用**可以更提升預測準確度

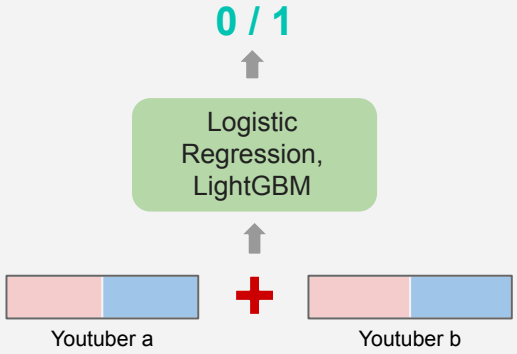


評估指標: ROC-AUC	1. Node2Vec (Add)	2. Node2Vec (Concat)	3. Attributes (Add)	4. Attributes (Concat)	5. Mix (Add)	6. Mix (Concat)
Logistic Regression	0.633	0.534	0.659	0.605	0.682	0.583
LightGBM	0.720	0.648	0.700	0.675	**0.749**	0.656

架構五(預測是否合作及合作後的成功度)

實驗結果 - 預測是否合作(任務一)



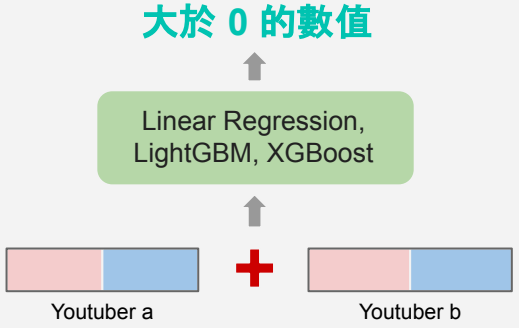


使用 Optimal threshold 的計算結果	
Accuracy	0.699
Recall	0.667

架構五(預測是否合作及合作後的成功度)

實驗結果 - 預測合作後的成功度(任務三)

- LightGBM 最穩定
- 最佳組合 : Mix (Concat) + LightGBM
- **Linear regression 在特徵向量維度比較大的時候會非常不穩定**，RMSE loss 可能會到幾千萬，因此幾乎無法使用，我們在表中標示為 FAILED



評估指標: RMSE loss	1. Node2Vec (Add)	2. Node2Vec (Concat)	3. Attributes (Add)	4. Attributes (Concat)	5. Mix (Add)	6. Mix (Concat)
Linear Regression	1.347	FAILED	1.095	1.086	FAILED	FAILED
XGBoost	1.125	1.091	1.130	1.099	1.116	1.108
LightGBM	1.113	1.078	1.087	1.062	1.094	**1.057**

實驗結果綜合比較 - 預測是否合作

	架構二	架構三	架構四	架構五
Recall	0.51	0.26	**0.88**	0.67
ROC-AUC	0.58	0.61	0.49	**0.75**

- 若考量兩種指標，**架構五**是最好的方法，因為它達到次高的 recall，而且有最高的 AUC，而各指標的分數都算是可以接受
- 若只考量 recall，代表我們希望能成功預測出越多合作組合越好，那就可以選擇 **架構四**，但可能會有較多誤報，因其 AUC 不到 0.5
- 若只考量 AUC，**架構五**是最好的方法，次高為架構三，而架構四的 AUC 顯示其判別正負樣本的能力與隨機猜測差不多

實驗結果綜合比較 - 預測合作影片類型

	架構三	架構四
Recall	0.14	**0.65**
ROC-AUC	0.55	**0.66**

- **架構四**是最好的選擇，不論在哪個指標都取得了最高的分數，而且AUC = 0.66 顯示其有區分正負樣本的能力，不是隨機分類

- 由上表可以看出，**預測合作影片類型相較於只預測合作對象難上許多**，因此分數都沒辦法到太高，但架構四的預測結果仍可提供參考

總結及未來展望

- 資料蒐集
  - 使用星座來代表個人影片風格的方式未盡理想。我們可以定義一些影片風格(例如:可愛型、知識型、搞笑型、宅男型、挑戰型等)，並用人工的方式去瀏覽影片並判斷每個 YouTuber 有哪些個人特質。
  - 影片類型的部分，除了使用 ChatGPT 來判讀影片標題之外，也可以加入影片逐字稿一起當作輸出讓模型標記，以免因為一些易混淆單詞造成標記錯誤。此外，在影片類型的定義上也可以更加精細。
- 實驗架構與模型
  - 在各任務下皆能找到一個較為合適的模型去使用。然而各個架構在不同評估指標下的表現差異顯著，這表示我們的模型仍有很多進步空間。
  - 擴充資料(爬取更多資料、設計 data augmentation 解決資料標記不平衡的問題)
  - 優化模型(擴充使用的 Youtuber 特徵、使用能力更強的深度學習模型)

参考資料

[1] Liu, G. (2021). An ecommerce recommendation algorithm based on link prediction. Alexandria Engineering Journal, 61(1), 905-910. https://doi.org/10.1016/j.aej.2021.04.081
[2] Antonio Ferrara, Lisette Espin-Noboa, Fariba Karimi, and Claudia Wagner. 2022. Link recommendations: Their impact on network structure and minorities. Proceedings of the 14th ACM Web Science Conference 2022 (WebSci '22). Association for Computing Machinery, New York, NY, USA, 228–238. https://doi.org/10.1145/3501247.3531583
[3] N. Kuang, Y. Zuo, Y. Huo, L. Jiao, X. Gong and Y. Yang, "Network Link Connectivity Prediction Based on GCN and Differentiable Pooling Model," 2022 IEEE 14th International Conference on Advanced Infocomm Technology (ICAIT), Chongqing, China, 2022, pp. 1-6, doi: 10.1109/ICAIT56197.2022.9862715.
[4] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 58(7), 1019-1031. https://doi.org/10.1002/asi.20591
[5] Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications, 390(6), 1150-1170. https://doi.org/10.1016/j.physa.2010.11.027
[6] Grover, Aditya; Leskovec, Jure (2016). "Node2vec". Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Vol. 2016. pp. 855–864.

謝謝大家！