ADL Final

黃秉茂 r11944024 黃子瑋 r10944045 高涵毅 r11922a11 吳祐任 r11922170

Abstract

本次Final project裡我們利用了ALS、Bert、Sentence transformers、T5、ckip-transformers、multi-label classification與XGBoost做預測，也嘗試實作了基於統計的推薦模式，並分別在不同task上進行實驗，最後進行結果的分析與討論。

Introduction

我們的Final Project選擇參加Kaggle競賽，題目與資料皆由Hahow線上學習平台所提供，目標是記住不同課程間的相關性，並預測用戶未來可能會購買的課程，此次競賽針對預測對象分成課程(course)跟子類別(subgroup)兩項，每個項目再根據預測用戶是否存在於訓練資料切成seen與unseen兩份，四個子競賽的評分指標皆為MAP@50。

問題的解法除了參考助教提示研究Alternating Least Square與BM25，並結合本學期課程使用多種自然語言處理技巧，例如Bert, T5, ckip-transformers，此外我們將題目視作multi-label classification運用XGBoost做預測，最後額外嘗試使用新穎的推薦系統GNN，成員的工作分配列在下方：

- 黃子瑋: Seen/Unseen Course Prediction, ALS, Matrix Factorization, Heuristic Approach實作/實驗、報告撰寫/錄投影片
- 吳祐任: Seen/Unseen course/topic prediction，bert multi label，T5，XGBoost，statistics Approach、報告撰寫/錄投影片
- 黃秉茂: Seen/Unseen course/topic prediction，BM25，Text preprocess + Okapi BM25、statistics Approach、報告撰寫/錄投影片
- 高涵毅: Seen course prediction，Sentence Similiarity, GNN-based CF、報告撰寫/錄投影片

Related work

Alternating Least Squares

又稱交替最小平方法，時常使用在推薦系統中的演算法。該演算法透過觀察輸入的稀疏矩陣中使用者對於物品的評分來推斷相似使用者感興趣的物品並給予推薦。

Bert

Bert(Bidirectional Encoder Representations from Transformers)是一種基於transformer的架構，它使用一種稱為掩蔽語言研究建構的技術進行訓練，模型需要預測被掩蔽的輸入的原始值。這個訓練過程使Bert能夠學習句子中的單詞之間的上下關係。
一但經過訓練，Bert就能針對特定的NLP任務進行fine tune，例如文章分類、問答、語意探索等等。
Bert的主要優勢之一是它在非常的的數據及上進行預訓練，使它能夠學習通用的語言特徵，這讓它成為廣泛NLP任務中的佼佼者，performance超越了當時許多模型成為AI領域中的霸主。

Sentence transformers

Sentence transformers[1] 是一個用於自然語言處裡(NLP)任務的python framework，透過使用pre-trained model把句子轉化成embeddings。然後這些embeddings 可用於各種NLP任務，如句子分類、相似性比較或分群。該框架是基於pyTorch與Transformers之上，提供了大量pre-trained modeld可供使用者下載並在自己的任務上進行fine tune，將模型應用於廣泛的NLP任務之中。

t5(Text-to-text)

T5 (Text-To-Text Transfer Transformer)是google開發的一種基於transformer的大規模語言模型。它是google在bert發表後一年所開發出來的預訓練模型，在眾多的NLP任務中取的最先進的成果，包括語言翻譯與summary與QA。
T5在龐大的數據上訓練，作者們從網路蒐了極大量的訓練數據用於訓練模型，通過把所用問題都轉化成(text to text)的方式訓練。
T5與其它模型不同的優勢在於，它可以執行多種NLP任務，而不需要為每種任務都單獨訓練一個模型。這讓T5成為一時的熱門主題。

Bert multi-labels

這是基於Bert的一種模型應用，它把原先使用於分類的bert model的輸出通過sigmod輸出機率，藉此完成了bert的多標籤分類應用。

XGboost

XGboost是一個gradient boosting trees的開源library。XGBoost是由gradient boosting 演算法延伸而來的，gradient boosting algorithm是一種利用弱預測模型集成的機器學習技術。
相比與深度學習模型，XGBoost訓練速度更快，在Kaggle或其他數據競賽中經常被用來快速建立baseline。
XGBoost有許多的超參數，通過我們需要透過grid serach和交叉驗證來進行參數的調整以取得較佳的表現。

BM25

BM25(Best Match 25)[2]，是一種在訊息檢索中的一種演算法，給定搜索查詢的文件，對該文件的相關性進行評分。它是從TF-IDF[3]的改進，被廣泛用在搜尋引擎或其它訊息檢索系統中。

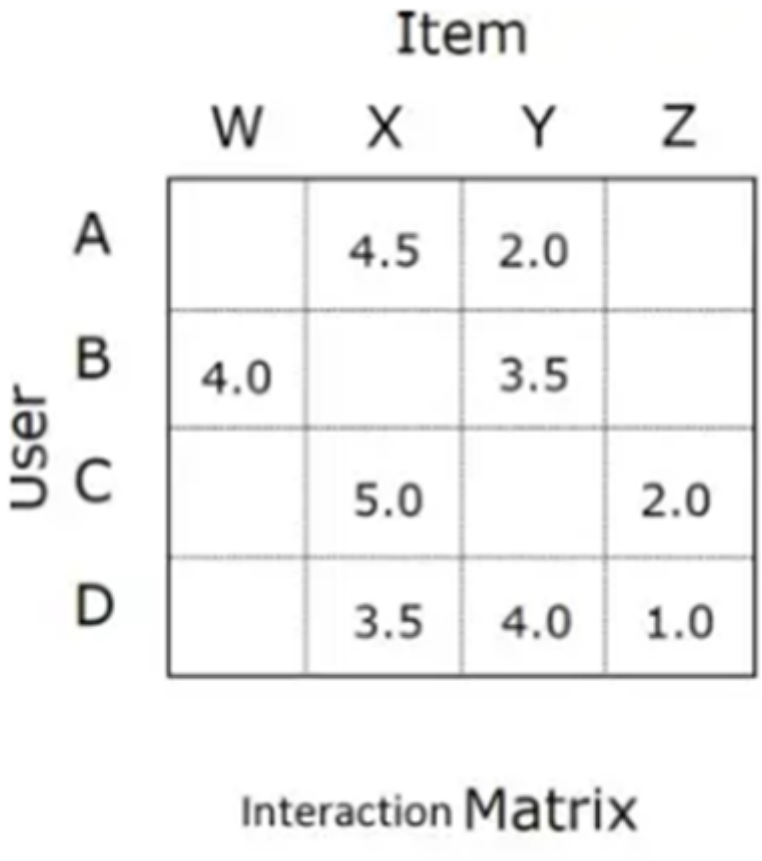
NN-based CF

Collaborative Filtering (CF) 是一種在傳統推薦系統上常用的演算法，粗略地分成Memory-based, Model-based, Hybrid三種方法。Memory-based又可以細分成User-based與Item-based的方法，而Model-based過去會使用Bayesian network或latent semantic models等資料探勘的方式，隨著深度學習火紅發展，結合圖(graph)的神經網路被認為是解決非結構化資料的利器，著名的方法如GCN[4], GAT[5], GraphSage[6]。

Approach

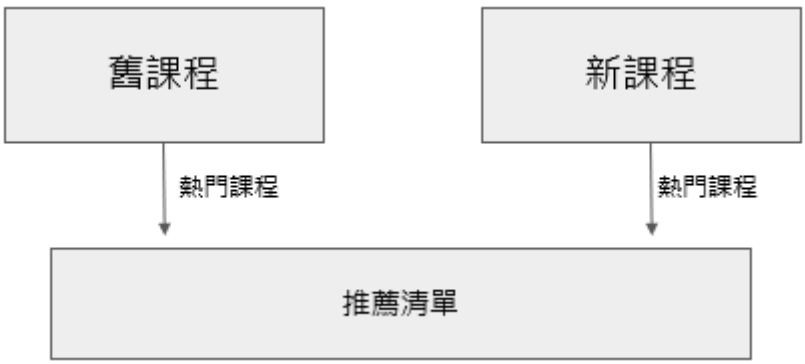
Alternating Least Squares

- 輸入為稀疏的Interaction Matrix，每個Row為使用者Column為Item，矩陣中的Element為使用者對於Item的喜好程度。透過使用者的購買紀錄分析使用者的喜好相似性進行推薦。



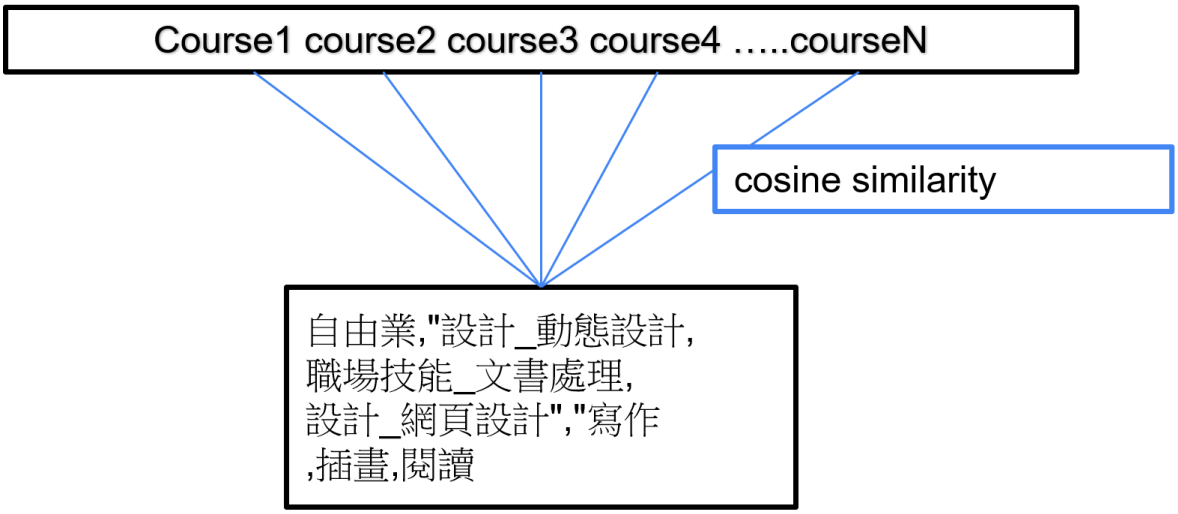
Heuristic Approach

- Validation Data統計觀察結論:
  - 消費者比較傾向於**購買較新的課程**
  - 消費者較傾向於**購買熱門課程**(購買次數最多)
- 實作方法
  - 新舊課程為Threshold為自由調整的參數，例如本報告使用課程編號600以後為新課程，以前為新課程
  - 分別從新舊課程中選擇出熱門的課程並結合出推薦List，其中舊課程選出23個，新課程選擇27個
  - 推薦List以購買次數做排序推薦給所有使用者
  - Kaggle: 0.14735



Sentence Similarity(cosine similarity)

- Model
  - shibing624/text2vec-base-Chinese
  - distiluse-base-multilingual-cased-v2
- Input
  - occupation_titles
  - interests
  - recreation_names

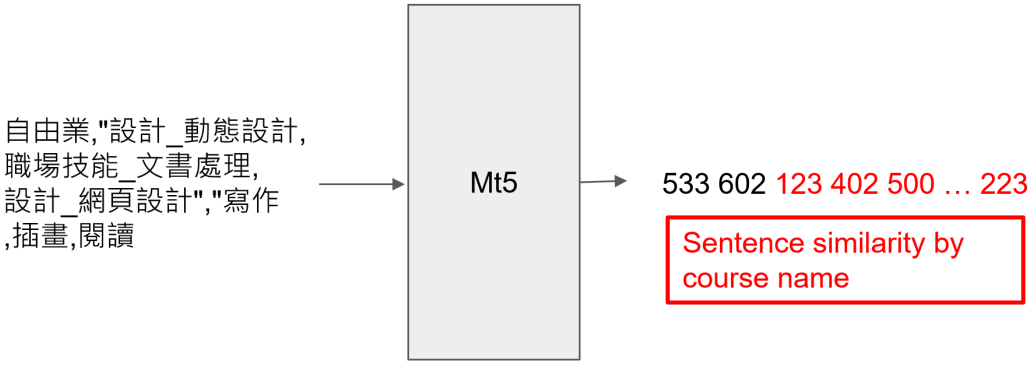


- N 可以用熱門課程替換縮小推薦範圍
- 計算input跟課程名字的cosine similarity
- 取分數最高的前五十名作為結果

Seq2Seq

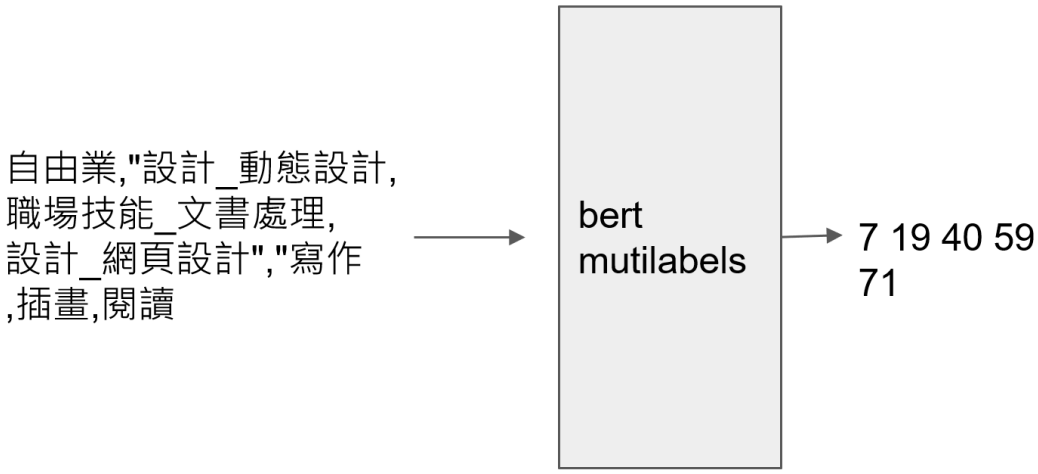
- Model
  - google/mt5-small
- x
  - occupation_titles+interests+recreation_names
- y
  - course labels
- loss fuction
  - cross entropy loss
- 模型自己決定要輸出多少
- 原始輸出不足五十時用Sentence Similarity從熱門課程中補到50個
- model config：
  - epoch 20
  - batch size 4
  - learning_rate=3e-4
  - gradient_accumulation_steps = 8

- weight_decay=0.01



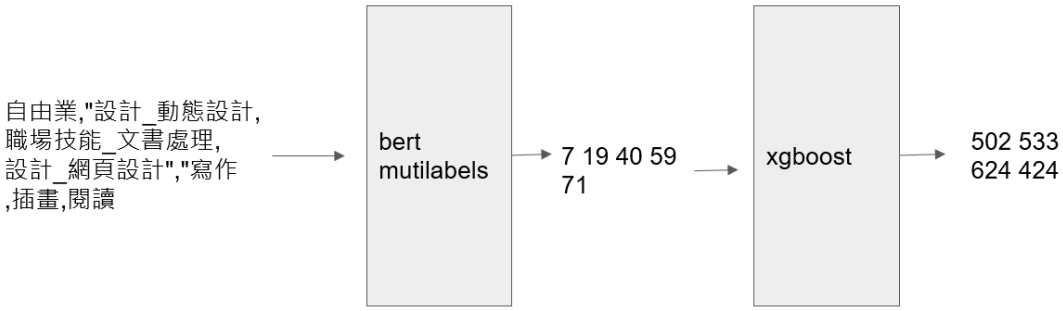
Bert-multi label

- Model
  - hfl/chinese-roberta-wwm-ext
- x
  - occupation_titles
  - interests
  - recreation_names
- y
  - 728課程one hot encoding labels
- loss fuction
  - BCELoss
- 模型輸出通過sigmoid以高到低作為預測課程
- model config：
  - epoch 20
  - batch size 24
  - learning_rate=3e-5
  - gradient_accumulation_steps = 2
  - weight_decay=0.01



Bert-multi label+XGboost

- Bert-multi label for topic and XGboost for course
  - 使用Bert-multi label預測出user可能會購買的topic，在利用topic與課程訓練xgboost分類器
- input：
  - occupation_titles
  - interests
  - recreation_names



Sentence Similarity + feature filter

- 針對前550課程，取最熱門的10堂課
  - train購買紀錄對比較新的課程不友善，如果一起排序可能會把答案排除
- 551~728利用課程日期、價格、出現次數綜合排序
  - 因為最後面的課程(最新)在validation set上購買次數也都幾乎是0，無法簡單利用新課程加購買次數及價格判斷時，就以資訊不足，刪除該課程不進入推薦課程名單中。
- 加入免費課程
- 用以上幾種規則選出候選的課程集套用Sentence Similarity

Text preprocess + Okapi BM25

- Text preprocess：
  - processing
    - Lexicon Normalization
      - Stemming
      - Lemmatization
    - Tokenization
    - Remove Stop Words
    - Concat all course infos
  - package
    - jieba, nltk
- Okapi BM25
  - bag-of-words retrieval function

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{f(q_i,D) * (k_1+1)}{f(q_i,D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})}$$

  - D: documents
  - Q: query

- IDF: inverse of documents freq
- f: freq of terms
- k, b: parameters

LightGCN

為較輕型的GCN模型，屏除過去GCN使用的特徵轉換非線性的activation函數，單純使用線性轉換做message-passing，每個user與item的embedding可以用以下兩個式子描述：

$$e_u^{(k+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(k)}$$

$$e_i^{(k+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_u|}} e_u^{(k)}$$

Experiments

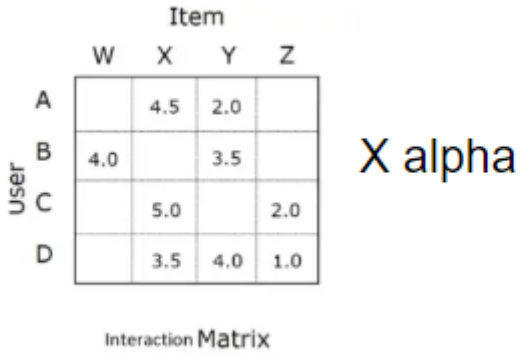
Alternating Least Squares

- Model Config
  - factors = 500
  - iterations = 3
  - alpha = 120
- Result
  - Validation Accuracy: 0.080
  - Kaggle Accuracy: 0.05686
- Factor對於Performance的影響
  - Factor可以理解成模型的複雜程度或是模型的大小
  - 過大的Factor可能會導致Overfitting

Factor	Validation Accuracy
100	0.062
200	0.073
500	0.062
100	0.080
1000	0.078

- 將Alpha放大對於效果提升的影響:

- Alpha可以理解為對於消費者的喜好進行加強，為一個乘上Interaction Matrix的常數



- Alpha放大對於Performance效果顯著

Alpha	Validation Accuracy
1	0.043
5	0.065
10	0.07
40	0.077
80	0.079
120	0.080

- Alpha對於Logistic Matrix Factorization Performance影響
  - 將Alpha放大使用在Logistic Matrix Factorization上並不如ALS有優異表現

Alpha	Validation Accuracy
1	0.057
2	0.054
5	0.053
10	0.051

- Iteration對於Performance的影響
  - 過大的Iteration對於Performance無益

Alpha	Validation Accuracy
2	0.078
3	0.080
5	0.076
10	0.072

Sentence Similarity(cosine similarity)

- Improvement
  - 使用熱門課程縮小候選課程集合可以大幅提升結果
  - 分別取train set上的前300、150、100、50與所有課程比較在validation set上的表現

	728	**300**	**150**	**100**	**50**
shibing624/text2vec-base-chinese	0.0298	0.0280	0.0316	0.0392	0.0460
distiluse-base-multilingual-cased-v2	0.0174	0.0236	0.0332	0.042	0.0542

Seq2Seq

- result：
  - unseen course(純粹使用seq2seq不加額外預測)
    - validation set：0.05412
    - kaggle：0.06238
  - unseen course(使用seq2seq加上sentance similarity預測)
    - validation set：0.08
    - kaggle：0.07487

Bert-multi label

- result：

	seen_topic	**unseen_topic**	**seen_course**	**unseen_course**
validation		0.3127		0.1093
kaggle	0.28125	0.31208	0.04533	0.1093

Bert-multi label+XGboost

- result：
  - unseen course
    - validation：0.084
    - kaggle：0.08327

Sentence Similarity + feature filter

- result：
  - unseen course：
    - kaggle：0.09947
  - seen course：
    - kaggle：0.1047

Text preprocess + Okapi BM25

seen course	seen subgroup	unseen course	unseen subgroup
0.03120	0.23871	0.05527	0.18876

NN-based CF

- seen course: 0.0804

Discussion

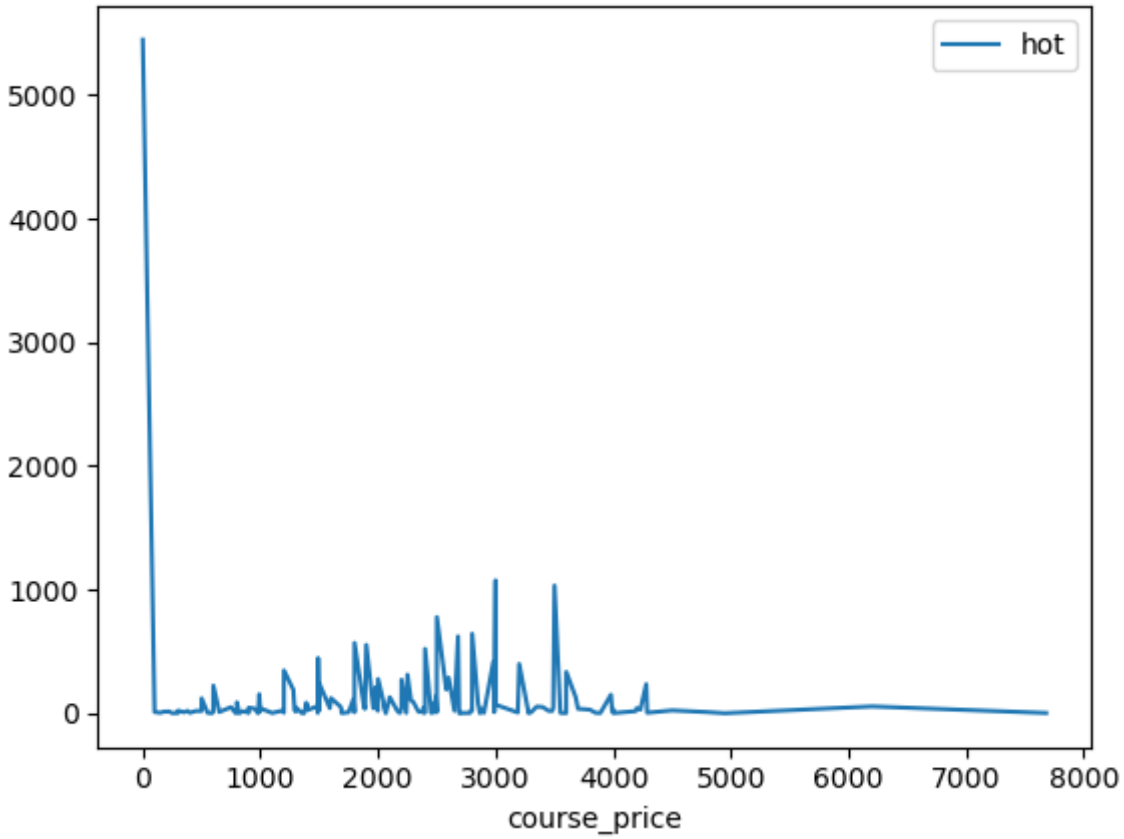
Kaggle Result

method \ task	seen course	seen subgroup	unseen course	unseen subgroup
seq2seq + Sentence Similarity	(no submission)	(no submission)	0.07487	(no submission)
Sentence Similarity + feature filter	0.11776	(no submission)	0.11363	(no submission)
Bert-multi label	0.04533	**0.31208**	0.08227	**0.28125**
Bert-multi label + XGboost	(no submission)	(no submission)	0.08327	(no submission)
Text preprocess + Okapi BM25	0.03120	0.23871	0.05527	0.18876
Sentence Similarity + statist	**0.14735**	(no submission)	**0.16253**	(no submission)

我們嘗試了learning base、rule base與統計學上的方法，我們可以發現最好的就是統計學的方法，在我們嘗試的眾多方法中，對於那些沒有使用到統計資訊的model往往我們只要在預測中加入熱門課程，或是改變搜尋空間到熱門選項中就能發現結果會提升不少，統計熱門課程讓我們的performance提升不少。

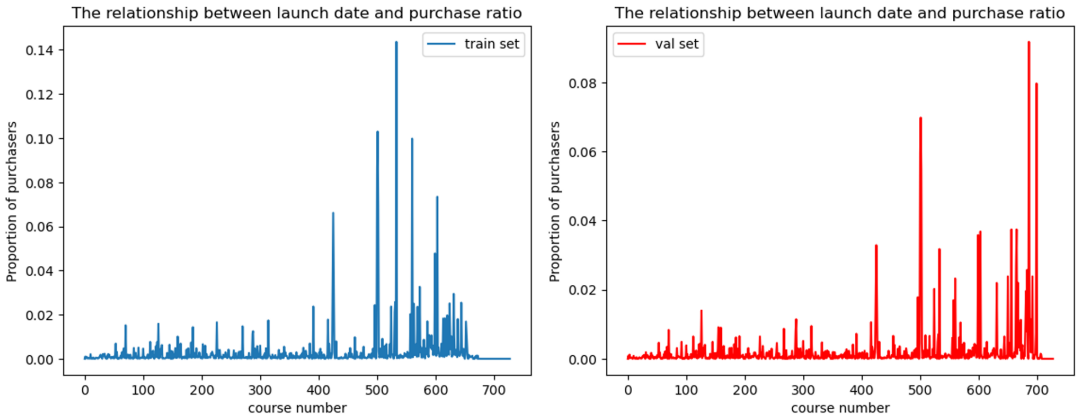
一個很明顯的原因是由於這個評分所採用的mAP@50，如果我們能夠知道哪堂課程在整體資料出現最多次，那我們只要簡單的預測所有人同一個課程並且以出現次數排序即可獲得很高的分數。

除了計算公式上的特性，我們可以發現顧客就是會比較傾向購買熱門課程，這可能與該網站上的廣告或者本身內容足夠吸引人有關，熱門課程就是會有它的原因，因此我們可以在推薦時優先推薦這些課程。

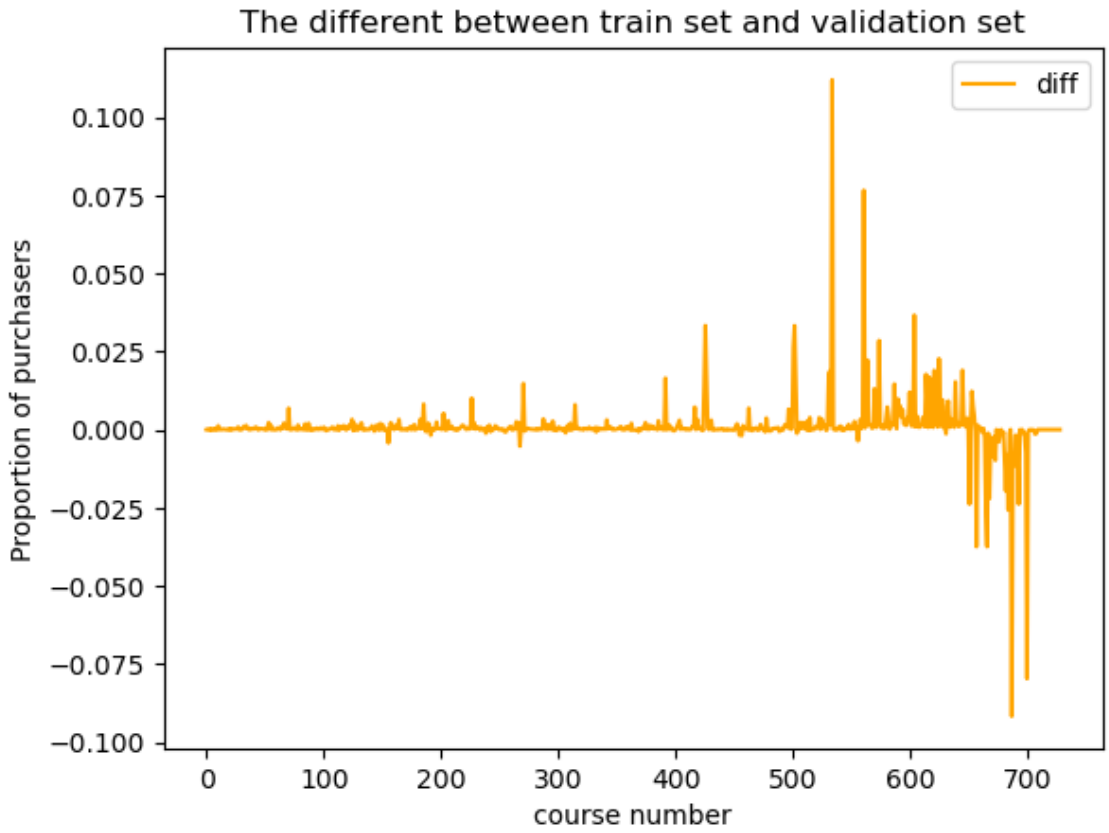


從價格上討論，免費課程明顯是壓倒性的熱門，但價格不是免費，價格與熱門程度就沒有較明顯的關係，這表示用戶對商品的選擇依然會回到注重內容、自身興趣等等。

- 課程購買比率，x軸為課程編號，y軸是該課程在該訓練集被購買比率



- 課程購買比率在訓練集與驗證集的差異(使用訓練集比率減去驗證集比率)



以課程推出時間來討論，透過以上兩張圖片我們可以發現，在大約編號550以前，訓練集與驗證集中的熱門課程差異並不大，但在550到630之間訓練集上就有兩門是非常熱門但在驗證集卻表現普通，而在630以後在驗證集的熱門課程卻在訓練集中完全沒有。
造成以上現象的原因在於train set的資料是用戶到8/31的購買紀錄，但編號630以後的課程卻是9/1才推出，這代表我們訓練model時會直接忽略後面的課程。驗證集是9/1到10/31的購買紀錄，從圖上可以看出驗證集中的熱門課程有相當多集中在編號630以後，所以只讓模型學會train data是不夠的，同理，這個現象也會影響在最後的分數上，因此我們利用課程推出時間、課程價格、熱門度等多項指標來選出我們的推薦候選課程。

結合以上討論，當我們在無法獲得更多資訊(用戶與課程的互動:點擊、評論等等)，我們可以利用統計資訊來快速建立一個推薦系統，這是一個簡單又有效的方法，我們也很容易在生活中發現推薦熱門項目的各種應用場景，而若能獲得更多特徵，比如用戶興趣、職業，就能以深度學習模型加上統計資訊的方式讓推薦系統更加準確。

Conclusion

本次我們嘗試了多種方法來實作該資料集的推薦系統，我們認為learn base(Bert)的表現會比rule base(BM25)的要好。
資料的分布在推薦系統上是非常重要的資訊，在缺少更多情報的狀況下，可以專門推薦熱門課程，這是快速有效的策略。
我們也能從本次實驗中知道，推薦系統並不是GNN的專利，善用transformer也能在推薦系統上取得不錯的效果。

reference

1. https://www.sbert.net/index.html ↩
2. https://en.wikipedia.org/wiki/Okapi_BM25 ↩
3. https://en.wikipedia.org/wiki/Tf%E2%80%93idf ↩

4. https://arxiv.org/abs/1609.02907 ↩
5. https://arxiv.org/abs/1710.10903 ↩
6. https://arxiv.org/abs/1706.02216 ↩