

國立中央大學

資訊工程學系
碩士論文

融合多模型排序之點擊預測模型

Learning to ensemble ranking model for sequential click
prediction

研究生：賴品菁

指導教授：陳弘軒 博士

中華民國一百零九年六月

國立中央大學圖書館學位論文授權書

填單日期：109/7/17

2019.9 版

授權人姓名	賴品菁	學號	107522044
系所名稱	資訊工程所 資訊工程所	學位類別	<input checked="" type="checkbox"/> 碩士 <input type="checkbox"/> 博士
論文名稱	融合多模型排序之點擊預測模型	指導教授	陳弘軒

學位論文網路公開授權

授權本人撰寫之學位論文全文電子檔：

- 在「國立中央大學圖書館博碩士論文系統」

(☒)同意立即網路公開

()同意 於西元_____年_____月_____日網路公開

()不同意網路公開，原因是：_____

- 在國家圖書館「臺灣博碩士論文知識加值系統」

(☒)同意立即網路公開

()同意 於西元_____年_____月_____日網路公開

()不同意網路公開，原因是：_____

依著作權法規定，非專屬、無償授權國立中央大學、台灣聯合大學系統與國家圖書館，不限地域、時間與次數，以文件、錄影帶、錄音帶、光碟、微縮、數位化或其他方式將上列授權標的基於非營利目的進行重製。

學位論文紙本延後公開申請 (紙本學位論文立即公開者此欄免填)

本人撰寫之學位論文紙本因以下原因將延後公開

- 延後原因

()已申請專利並檢附證明，專利申請案號：

()準備以上列論文投稿期刊

()涉國家機密

()依法不得提供，請說明：_____

- 公開日期：西元_____年_____月_____日

※繳交教務處註冊組之紙本論文(送繳國家圖書館)若不立即公開，請加填「國家圖書館學位論文延後公開申請書」

研究生簽名：賴品菁

指導教授簽名：陳弘軒

*本授權書請完整填寫並親筆簽名後，裝訂於論文封面之次頁。

國立中央大學碩士班研究生
論文指導教授推薦書

資訊工程學系碩士班 學系/研究所 賴品菁 研究生
所提之論文 融合多模型排序之點擊預測模型
係由本人指導撰述，同意提付審查。

指導教授

洪弘毅 (簽章)

109 年 7 月 14 日

國立中央大學碩士班研究生
論文口試委員審定書

資訊工程學系碩士班 學系/研究所 賴品菁 研究生
所提之論文 融合多模型排序之點擊預測模型
經由委員會審議，認定符合碩士資格標準。

學位考試委員會召集人



委

員

陳弘毅

鄭世詠

許長峰

中 華 民 國

109 年 7 月 7 日

融合多模型排序之點擊預測模型

摘要

現今因為網路的蓬勃發展，網路與人們的生活密不可分，在網站的需求上，不管是購物網站或是音樂、影片網站，越來越多網站有個人化的服務及推薦。若能有效的推薦商品給使用者，不但能增加使用者對此網站的滿意度，還能讓這個網站的創造者獲得收益，造就雙贏的局面。

本篇論文利用兩種不同大小的電商資料，對使用者下一個可能會點擊的商品做排序，推薦給使用者。為了將真正被點擊過的商品排在更前面的推薦序列中，我們使用了四種不同類型的模型來做排序。由於最近深度學習在各個領域表現卓越，推薦系統也開始使用深度學習架構來訓練，但我們發現，在不同的電商規模下，使用深度學習架構不一定會表現得比淺層模型好。

在發現了這個現象之後，我們使用這四個不同的模型排序過後的序列排名，來當作我們的訓練資料，並重新排序，最後我們得到了更符合使用者歷史點擊的排名，將真正被點擊過的商品的排名往前排，並獲得明顯的進步。不管在哪個資料集上，都表現的比四個基準模型還要好，並在實驗過程中發現了商品排名與商品出現次數的相關性。

關鍵字： Top-N 排序、推薦系統、點擊率預測、機器學習、深度學習

Learning to ensemble ranking model for sequential click prediction

Abstract

With the vigorous development of the Internet, it becomes inseparable from people's lives. Various websites, such as online retailers and online music/video streaming websites, provide personalized recommendations to show the RIGHT products to users. These recommendations may increase users' satisfaction with these services and create revenues for the service providers.

This paper aims to predict a users' next clicking item based on two e-commerce datasets. We compared four ranking models and claimed a model performs better if the model ranks the next clicking item at a former position among the four models. Despite recent researches showed excellent performance on deep learning-based ranking models, we found that this is not always the case. Notably, we may need to consider the size of the service providers as a reference to decide to apply a deep or a shallow learning model.

Since different models may work under various scenarios, we developed an ensemble model that learns to rank the items based on the rankings returned by the four ranking models. Experimental results show that the new model outperforms the four baselines: on both datasets, the new model tends to put the next clicking item at a former position.

Keywords: Top-N recommendation, recommendation system, prediction of CTR, machine learning, deep learning

目錄

	頁次
摘要	ix
Abstract	xi
目錄	xiii
一、緒論	1
1.1 研究動機	1
1.2 研究目標	2
1.3 研究貢獻	2
1.4 論文架構	3
二、背景及相關論文	5
2.1 推薦系統 Top-N 名候選商品排名	5
2.2 簡易模型與深度學習於推薦系統之應用	7
2.3 深度學習模型是否確實提升推薦系統效果	9
三、實驗發想與方法	11
3.1 候選商品召回與商品向量訓練	11
3.2 多層神經網路 (Multilayer perceptron)	12
3.3 阿里巴巴集團提出之深度學習模型	13
3.3.1 Deep Interest Network (DIN)	14
3.3.2 Deep Interest Evolution Network (DIEN)	16

3.4	融合多模型排序結果	17
四、	實驗設置與結果	19
4.1	資料集介紹	19
4.1.1	電商 A 資料集	20
4.1.2	阿里巴巴集團旗下淘寶資料集	20
4.2	實驗資料配置與前處理	21
4.2.1	資料集介紹與前處理	21
4.2.2	四個基礎模型實驗介紹	22
4.3	評量指標	24
4.4	實驗結果	25
4.4.1	基礎模型在對應資料集之排序結果	26
4.4.2	參考基礎模型排序資訊之重新排序候選商品結果	30
五、	結論與未來展望	35
5.1	結論	35
5.2	未來展望	36
	參考文獻	37

圖目錄

	頁次
2.1 Youtube 推薦系統流程圖	6
2.2 item-based 模型示意圖 [14]	7
2.3 DIEN 模型示意圖 [24]	8
3.1 Gensim 訓練詞向量的流程圖	12
3.2 MLP 模型架構內容	13
3.3 DIN[23] 模型架構	15
3.4 DIEN[24] 模型架構	16
3.5 測試資料處理後格式	18
4.1 資料前處理範例	22
4.2 MLP 訓練過程	23
4.3 資料切割說明	25
4.4 淘寶平均排名與商品出現次數之關係圖	28
4.5 電商 A 平均排名與商品出現次數之關係圖	28
4.6 淘寶在 KNN 和 MLP 模型下平均排名與商品出現次數之 關係圖	29
4.7 電商 A 在 KNN 和 MLP 模型下平均排名與商品出現次數 之關係圖	29
4.8 淘寶在利用決策樹融合模型下平均排名與商品出現次數之 關係圖	32

4.9 淘寶在利用 XGBoost 融合模型下平均排名與商品出現次數之關係圖	32
4.10 淘寶在利用邏輯式迴歸融合模型下平均排名與商品出現次數之關係圖	32
4.11 電商 A 在利用決策樹融合模型下平均排名與商品出現次數之關係圖	33
4.12 電商 A 在利用 XGBoost 融合模型下平均排名與商品出現次數之關係圖	33
4.13 電商 A 在利用邏輯式迴歸融合模型下平均排名與商品出現次數之關係圖	33

表目錄

	頁次
3.1 淘寶和 Amazon 商品數量與會員數量	14
3.2 訓練資料特徵資訊	17
4.1 資料集內容介紹	19
4.2 電商 A 資料資訊欄位	20
4.3 淘寶資料資訊欄位	21
4.4 淘寶之平均排名	27
4.5 電商 A 資料集之平均排名	27
4.6 淘寶在各模型 top-n 排名命中率比重	27
4.7 電商 A 在各模型 top-n 排名命中率比重	27
4.8 淘寶在融合模型預測平均排名與標準差	30
4.9 電商 A 在融合模型預測平均排名與標準差	30
4.10 淘寶在融合模型 top-n 排名命中率比重	30
4.11 電商 A 在融合模型 top-n 排名命中率比重	30

一、緒論

1.1 研究動機

因為網路的快速發展，現今越來越多人使用電子產品並在網路上購買物品或是看影片、聽音樂。由於此類網站被使用的非常頻繁，相對的每個使用者可以選擇的同類型網站也非常多，所以相較之下，若要能真正留住使用者，除了有高品質的商品外，推薦系統也是不可或缺的一環。

因為最近幾年深度學習的卓越成果，讓各個領域的人都相繼使用深度學習模型來實作各種任務。但在比較過程中，往往忘記最簡單最根本 (e.g.K-近鄰演算法) 的模型，這篇論文中就有提到，[1] 論文中就有提到。有越來越多人使用深度學習模型來改良推薦系統的架構，但隨著越來越多不同的深度學習推薦系統出現，衆多學者都只專注於深度學習模型之間的比較，卻沒有人拿深度學習模型與一般簡單架構模型做比較。在這篇論文的實驗下，用最簡單純粹的模型有時也能達到超過深度學習模型的成果，甚至單純的推薦最受歡迎的商品有時也能超過深度學習推薦系統。

這樣的發現也讓我們意識到，或許在不同規模大小的網站上，可能適用不同的推薦演算法。不同的網站因為規模大小不同，商品數量和使用者數量差距也會相當大。因此網路巨頭 (e.g. 淘寶、Amazon) 們所採用的推薦演算法不見得適用於所有不同規模大小的網站上，因為商品數量和使用者數量的差距，可能會導致複雜度較高的模型不一定會做的比簡單的模型還要好。

1.2 研究目標

本論文的目的是建立一個推薦系統中候選商品的新排序，讓使用者的點擊率增加，並探討資料量大小、商品出現的頻率與訓練模型間的關係。由於現今店商數量日益增加，規模都有所不同，若將大規模的電商推薦系統 (ex. 淘寶網站) 套用在小規模的電商網站，那效果不一定會有很好的表現。

我們使用真實的電商使用者和商品互動的資料，將資料中使用者有點擊過的商品蒐集起來，組成使用者一天中的點擊序列 (點擊流)，利用淺層機器學習模型和深度學習模型來預測商品會被點擊的機率，並生成一個新的候選商品推薦順序。所以我們希望能透過不一樣的模型來學習候選商品的排序順序，並讓不同資料量的電商有選擇不同模型排序結果的機會。

1.3 研究貢獻

本論文利用四個不同的基準模型，對候選商品做排序，並利用融合模型來訓練出一個新的候選商品排序結果，在接下來的章節會有實驗結果說明我們的研究在候選商品排序上，獲得不錯的結果，並在研究過程中，發現當商品在訓練資料的出現次數不同的時候，在不同的基準模型的預測排名也會有所不同，而且我們也找到了商品出現次數和排名之間的相關性。

本論文的研究可運用於各電商網站候選商品的排序上。電商網站每天日新月異，數量也越來越多，除了商品的品質之外，推薦系統的好壞也會影響使用者使用這個網站的頻率，但若每個網站都使用相同的推薦系統，那就無法藉由不同網站的特性去做改變，所以推薦系統應該要隨著網站不同而有所不同，其中我們研究發現的資料出現次數和排名的相關性，就可以為不同的網站帶來不同的推薦系統，讓每個網站都能達到

比較好的效果。

1.4 論文架構

本篇論文共分五個章節，其架構如下：

- 1) 第一章：說明本篇論文之研究動機、研究目標和研究貢獻
- 2) 第二章：介紹本篇論文之相關主題研究與其研究成
- 3) 第三章：說明本篇論文所使用的方法及模型架構
- 4) 第四章：呈現本篇論文的實驗結果和不同方法模型之間的比較結果
- 5) 第五章：本篇論文之結論與未來展望

一、緒論

二、 背景及相關論文

本章節中，我們首先介紹推薦系統中的 Top-N 排序任務，並將重點放在排序上，因為我們的論文主要探討的主題與排序有關。再來我們會介紹簡易模型和深度學習模型對推薦系統的應用與差異，透過比較兩篇不同的相關論文，來顯示簡易模型與深度學習模型之間的差別。最後我們透過對論文 [1] 中的實驗來探討，深度學習模型是否真的完全優於簡易模型。

2.1 推薦系統 Top-N 名候選商品排名

由於受到螢幕大小或使用者會瀏覽的範圍限制，系統一次只能展示極少數量的商品，於是 Top-N 推薦就成為推薦系統中相當有名的問題 [2]–[10]。從所有商品中，選出 N 個使用者最可能會點擊的商品後，再透過不同的推薦系統做排序，最後呈現給使用者。使用者對於這個推薦系統的評分，取決於使用者對這個推薦序列點擊了幾個項目，並且這些項目都被排在第幾名。利用這樣的資訊一再的修正 Top-N 推薦系統，讓使用者的點擊率 (click through rate, 簡稱 CTR) 上升。

Top-N 推薦系統主要分成兩部分：(1) 產生候選商品和 (2) 排序。在 [11] 論文中就清楚的展現了 Youtube 推薦系統流程，從如何篩選候選商品到排序候選商品，如圖2.1所示，從圖片左邊開始，在所有商品中先挑選使用者可能點擊的商品，之後再透過使用者的歷史資訊去對候選商品做排序，最後將排序結果的前 N 個商品推薦給使用者，如圖2.1右邊的最

後推薦序列。可以看到圖片左邊原本所有影片的種類很多樣，但最後推薦的序列，類別比較單一，因為是參考了使用者的歷史資訊所做的推薦。

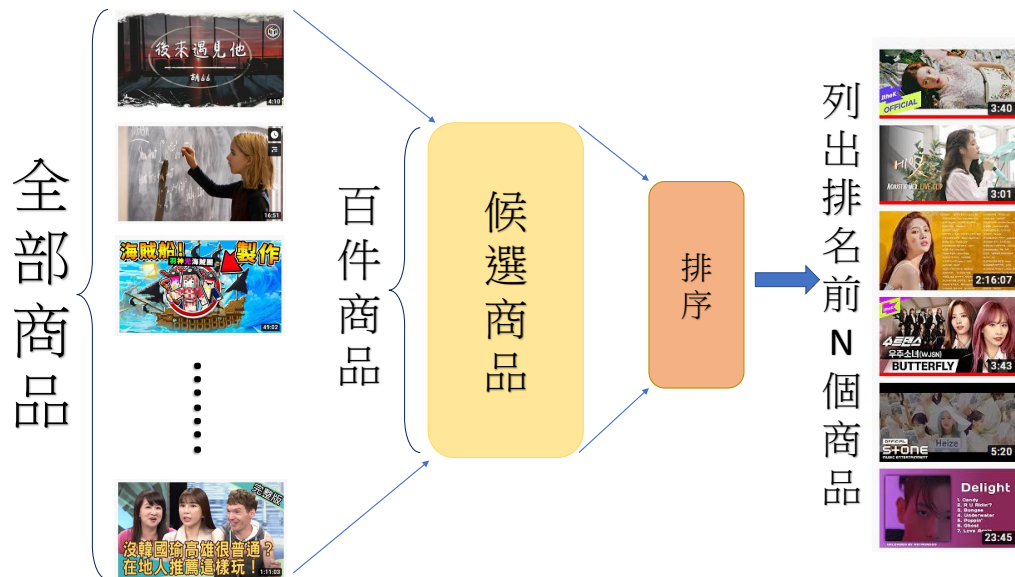


圖 2.1: Youtube 推薦系統流程圖

在 Top-N 排序的問題中，有非常多方法能對商品做排序，像是非常有名的方法—矩陣分解 (Matrix Factorization) 應用在 Top-N 排序問題上，在 [12] 中就是利用矩陣分解的方法來做排序；還有著名的協同過濾 (Collaborative filtering, 簡稱 CF) 方法，透過比較相似的使用者，間接獲得不同使用者對相同物品的評價。除了利用簡單的矩陣運算或相似性計算的方式來處理排序問題，近幾年也越來越多人將深度學習應用在推薦系統上，由於深度學習模型可以透過大量的使用者資訊，來學習使用者的興趣特徵及商品之間的相關性，所以在 Top-N 排序問題上，理應能獲得較好的成效。

本篇論文主要研究的部分是排序候選商品的部分，由於有眾多不一樣的方法來訓練排序問題的模型，但大部分的模型都是以預測某個商品會不會被點擊來當作訓練目標，模型會輸出目前要預測的商品被點擊的機率，然後再與真正被點擊的機率做損失計算，然後更新模型。本篇論文也是利用相同的訓練手法，套用在不同的模型來做排序，探討簡易模型和深度學習模型對排序問題效果的差異。

2.2 簡易模型與深度學習於推薦系統之應用

從最早的傳統機器學習到現今的深度學習，都能應用在推薦系統上。在 [12][13][14][15][16][17][18] 中使用得方法，都是以簡易模型來實現推薦系統，如圖2.2所示，是以 item-based 為主的推薦系統示意圖，利用使用者的歷史行為來學習出使用者的興趣特徵，並從全部商品中獲得與這些特徵有相關的商品，並推薦給使用者，就算現在深度學習已經被廣泛的應用，但這些簡易模型還是能有相當不錯的效果。

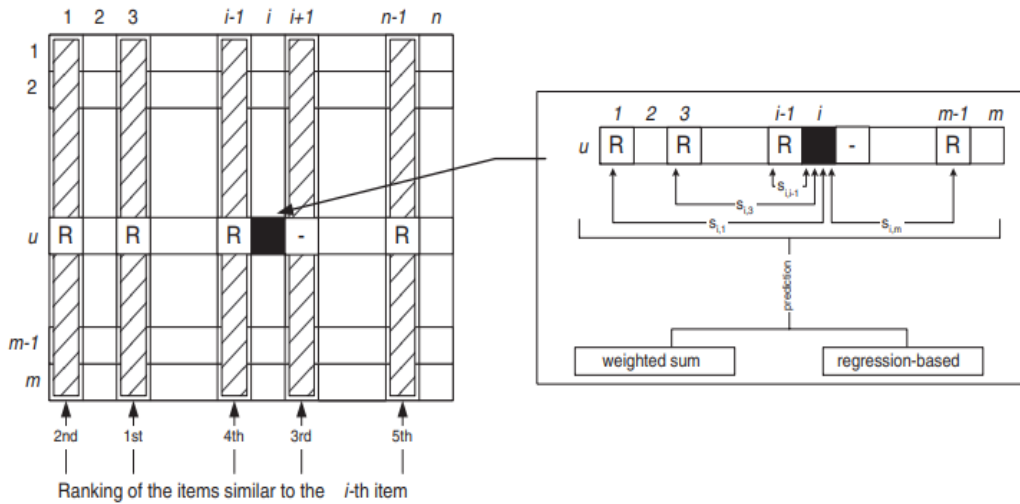


圖 2.2: item-based 模型示意圖 [14]

除了上述這些簡易模型，深度學習模型也被廣泛的應用於推薦系統上，在 [19][20][21][22] 這些論文中，都著重於深度學習模型的訓練，其中以阿里巴巴推出的模型為例，像 Deep Interest Network(DIN)[23] 模型和 Deep Interest Evolution Network(DIEN)[24] 模型都有利用到 2017 年 Google 提出的注意力機制 (Attention)[25] 架構，來捕捉使用者的歷史行為與要推薦的商品之間的關係，最後再利用全連接層來訓練此商品會被點擊的機率，藉此來判斷此商品該不該推薦給這個使用者。之後在 2019 年阿里巴巴集團又提出了透過時間段 (session) 來分割使用者在不同時期的歷史行為的模型 [26]，並透過這樣的資料分割來加以訓練，並得到比前面兩個模型還要好的結果。圖2.3展示 DIEN 模型的架構，在深度學習模

型中，與簡易模型較不同的是通常使用的模型都較為複雜並且模型的神經網路層都較多，並且使用的特徵都較多樣，主要就是為了能在這些眾多的特徵中找到使用者與推薦商品的關聯性。

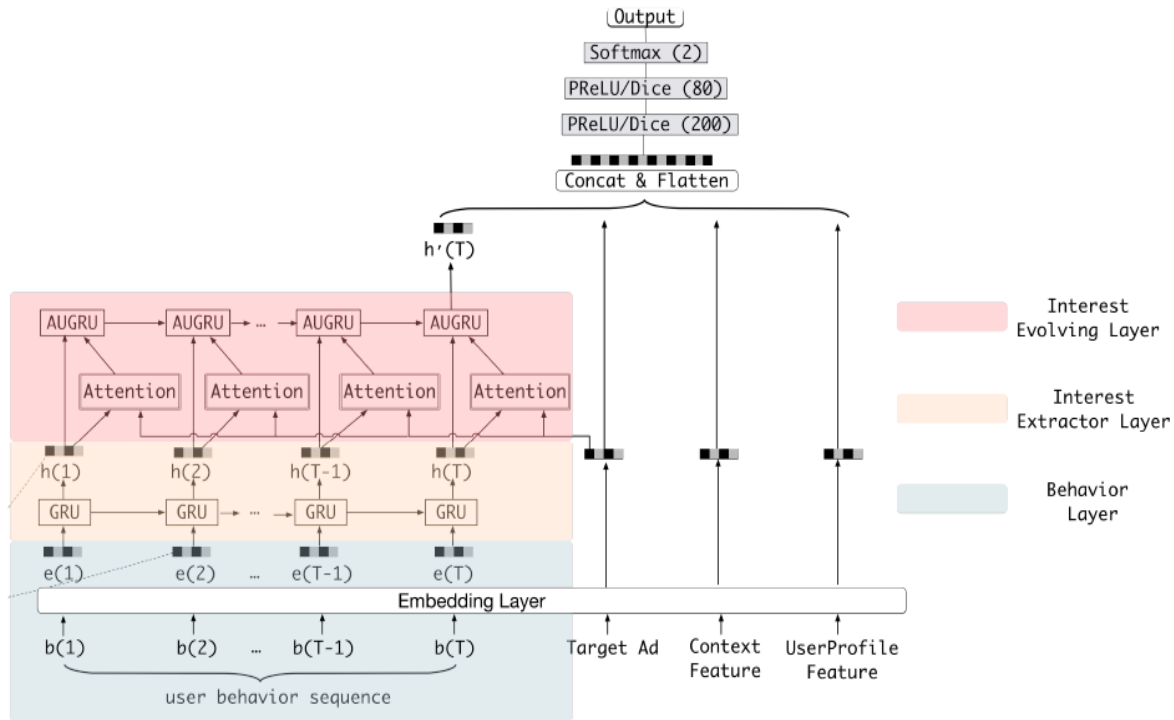


圖 2.3: DIEN 模型示意圖 [24]

我們以圖2.2和圖2.3為例，來比較簡易模型和深度模型。在圖2.2中，將所有使用者對所有商品的評分建構成一個使用者對商品的矩陣 (如圖片左邊矩陣)，但這個矩陣式一個稀疏矩陣，因為使用者不可能真的對所有商品都做評分，所以大部分的欄位會是空白。假設我們現在要預測使用者 u 對商品 i 的評分，則我們先將使用者 u 評分過的商品對商品 i 做相似度計算，在利用計算出來的相似度來當作這個商品分數的權重，並將每個有被評分過的商品分數乘上相對應的權重 (與商品 i 的相似度)，最後再相加，則就成為使用者 u 對商品 i 的預測評分。

在圖2.3中，模型架構主要分成三部分：行為層 (behavior layer)、興趣提取層 (interest extractor layer)、興趣演化層 (interest evolving layer)。行為層將使用者過去的歷史興趣輸入到嵌入層並得到每個興趣各自 (商品) 的向量。興趣提取層是將行為層輸出的向量透過 GRU 對時序性資料

的特性，將每個歷史興趣前面的興趣的特徵提取出來，傳遞給下一個歷史興趣，並透過 GRU 來決定要傳遞多少的資訊，最後再輸出新的歷史興趣向量。最後的興趣演化層將興趣提取層的輸出向量，與要預測的商品做注意力機制運算，輸出一個注意力分數，在與原本興趣提取層的向量做相乘然後輸入到 AUGRU 中，而歷史興趣的最後一個商品從 AUGRU 的輸出向量，則用來代表使用者所有歷史資訊。最後再將輸出的向量和其他相關資訊的向量合併並輸入全連接層做訓練，獲得一個機率值來判斷要預測的商品會不會被這個使用者點擊。

以近幾年學者們發表的論文來說，大部分的論文都只專注於探討深度學習模型的優化與創新，但就實際狀況來說，不見得每個網站的推薦系統都適合使用如此複雜的模型來做推薦，可能要透過不同網站的不同需求和不同的資訊量，來決定應該要使用怎麼樣模型才能達到更好的效益。接下來我們會探討深度學習模型是不是效果真的比簡易模型來的好。

2.3 深度學習模型是否確實提升推薦系統效果

雖然現在大部分的推薦系統論文，都在比較各種深度學習模型的效果，模型的架構也越來越複雜，但深度學習模型一定會比較好嗎？在 [1] 這篇論文中，就在探討深度學習模型是否真的完全優於一般的簡易模型或傳統的機器學習模型。論文中蒐集了 18 篇與 Top-N 排序問題相關的論文，並且都是使用深度學習模型，實驗過程利用每篇論文中描述的作法，重現所有深度學習模型的訓練過程，但由於某些論文因為資料集的問題，或是模型細節架構問題，而無法完全按照論文中描述的樣子重現，所以最後 [1] 的作者選了 7 篇可以重現並能完全複製實驗過程的論文來比較。

在 [1] 論文中選出了幾種簡易 Top-N 排序的方法 (ex.itemKNN 模型 [14] 和簡易隨機漫步 (random walks)[27] 等等) 來跟深度學習模型做比較，其中也包括單純推薦熱門商品這個方法，而深度學習模型方法包括

[20][21][22][28][29][30]。在 [1] 這篇論文的實驗中，實驗結果都以命中率 (hit rate) 和 NDCG 來衡量每篇論文中方法的優劣。在實驗結果中，大部分論文的深度學習方法，效果至少會在一個資料集下被簡易的模型超越，甚至有可能發生，只要推薦最熱門的商品，就能得到最大的效益，根本不用使用到過於複雜的模型來推薦。而且幾乎所有有使用的論文中的深度學習模型都可以被簡單的模型超越，以此可以證實，就算是簡單的排序模型，也能達到與深度學習模型相同的效果。此篇論文也提到，現今的研究中，只要實驗架構與深度學習相關的論文，很多都使用在某些資料集表現不好的模型來比較，或是只跟深度學習模型做比較，不與最簡單但卻最有用的模型做比較，以至於大家都只在乎模型的架構要多複雜、神經網路要多深等等，卻忽略最根本的方法其實也可以做得很好。

在緒論的研究動機中也有提到，因為這樣的實驗結果與發想，我們使用在亞洲地區最受歡迎的電商網站—淘寶—的資料集和淘寶的公司—阿里巴巴集團提出的 DIN[23] 和 DIEN[24] 模型來當作我們探討這個議題的深度學習模型，我們認為以中國的市場，淘寶的資料集可以說是亞洲地區數一數二豐富的，再使用我們蒐集的台灣某電商資料來做比較，可以看出不同電商在使用不同模型之間的差異。

三、實驗發想與方法

基於前一章所說，有越來越多的推薦系統利用深度學習模型來做推薦，大部分的研究者只專注於增進深度學習模型的效能。我們認為在使用者數量與商品數量多寡不同時，對於推薦系統模型的要求會有所不同。本章節會先介紹如何召回候選商品和商品向量如何取得，之後再介紹我們使用的四種複雜程度不同的模型，最後會介紹如何將前述的四種模型做結合，並訓練出更好的排序。

3.1 候選商品召回與商品向量訓練

我們使用的商品向量，是透過事先的訓練產生的。我們使用的資料集中，每一筆資料的呈現方式都是以使用者為單位，一個使用者在同一天中點擊過的所有商品為歷史點擊序列，如圖3.1所示，將使用者 A 的歷史點擊序列 (圖中 item B、C、D 序列) 當作一個句子，序列中的每個商品當作一個一個的字，然後利用訓練詞向量 (word embedding) 非常有名的 Gensim 套件¹來訓練我們的商品向量，Gensim 套件的訓練方式如圖中下方所示，由於上下文對於字詞的訓練很有幫助，所以 Gensim 在訓練的過程，會決定要看前後文的多少個字，來當作訓練的依據，這就是所謂的 window size。假設 window size 設為 1，則模型會參考當前要訓練的詞的前後 1 個詞的資訊，來訓練當前詞的詞向量，所以若我們當前要訓練的是 item C 的商品向量，則我們會參考前後兩個商品 (B 和 D) 資訊，圖

¹<https://radimrehurek.com/gensim/models/word2vec.html>

中的藍色框框代表 window 包含的範圍。

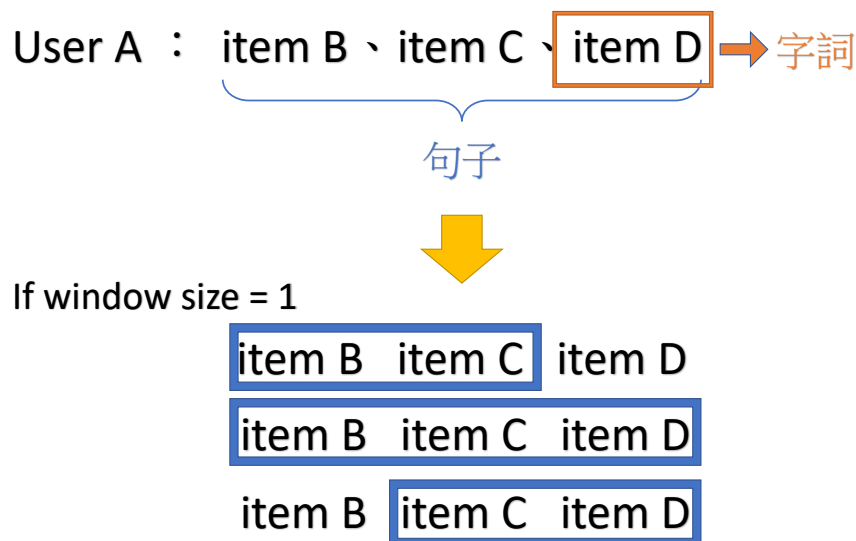


圖 3.1: Gensim 訓練詞向量的流程圖

透過上面描述的訓練過程得到的商品向量，成為我們召回候選商品的資料集。我們利用餘弦相似性 (cosine similarity) 計算商品與商品之間的相關性，並將這些相關性結果，按照相關性大到小的順序重新排序，召回每個商品最相似的九十九個候選商品。而這些召回的九十九個商品，就成為我們的第一個基準模型的結果。此模型是利用餘弦相關性直接計算最相關的九十九個商品來當作候選商品並做排序，最後得出的排序就是排序過後的結果，我們稱這個模型為 KNN 模型。

3.2 多層神經網路 (Multilayer perceptron)

我們為了分析深度學習模型與一般簡易監督式模型，對 Top-N 排序問題的影響有多大差異，我們建構了複雜程度介於餘弦相似性模型與 DIEN 模型 (相關論文中提到的深度學習模型) 之間的多層神經網路 (Multilayer perceptron, 簡稱 MLP)。

此 MLP 模型架構內容如圖3.2，此模型架構的輸入，是目前使用者點擊的商品或正在瀏覽的商品 (簡稱當前商品) 和下一個要預測會不會被

點擊的商品 (簡稱目標商品) 的商品向量 (item embedding)。商品向量是能表現出此商品特徵的一個等長維度的向量。將這兩個事先訓練好的商品向量當作 MLP 模型的輸入，輸出則是點擊當前商品後，接著點擊目標商品的機率。利用每個當前商品對應到的候選九十九個商品的機率，再去重新做排序。

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 512)	131584
leaky_re_lu_1 (LeakyReLU)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
leaky_re_lu_2 (LeakyReLU)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
leaky_re_lu_3 (LeakyReLU)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
leaky_re_lu_4 (LeakyReLU)	(None, 64)	0
dense_5 (Dense)	(None, 1)	65

```

Total params: 304,129
Trainable params: 304,129
Non-trainable params: 0

```

圖 3.2: MLP 模型架構內容

3.3 阿里巴巴集團提出之深度學習模型

阿里巴巴集團旗下有兩個規模最大的電商網站—淘寶和天貓。阿里巴巴的電商規模可說是在亞洲地區，少數可以超越 Amazon 網站的購物網站。不管是使用者數量或是商品數量，都非常龐大，且每位使用者對於商品的交互作用 (例如：瀏覽商品、評分商品、購買商品、點擊商品等等)，都非常頻繁。我們以淘寶網站為例，資料顯示，淘寶每天有超過 6000 萬的固定訪客，平均每分鐘出售 4.8 萬件商品。表3.1顯示 Amazon 電商網站與淘寶網站的商品與使用者數量。

表 3.1: 淘寶和 Amazon 商品數量與會員數量

淘寶	商品數量達到 10 億件，註冊會員近 5 億
Amazon	商品數量大約 3 億件，註冊會員近 1 億 5 千萬

由於使用者對於商品的喜好，變化性大且廣泛，若要透過使用者以往的所有紀錄去預測使用者接下來會點擊什麼樣的商品，並推薦，是非常困難的一件事，所以阿里巴巴集團分別在 2017 年和 2019 年月相繼提出了兩個推薦系統的架構——Deep Interest Network (DIN)[23] 和 Deep Interest Evolution Network[24] (DIEN)——來提升點擊率 (click-through rate, CTR)，以下會介紹兩種模型架構和差異。

3.3.1 Deep Interest Network (DIN)

如前面所說，阿里巴巴從使用者的豐富歷史行為中捕捉使用者的興趣時，發現使用者興趣有兩個主要特點：

- 1) **興趣多樣性**：使用者對商品的興趣非常廣泛。例如一位學生可能會購買文具用品，也會購買遊戲商品。
- 2) **局部商品興趣聚焦**：使用者是否會點擊某一件商品，往往是基於使用者以前點擊的部分商品，而不是所有商品。例如使用者是否會點擊我們推薦給他的商品—羽球，與他之前購買過羽球拍這個行為相關，而不會與他之前購買過冰淇淋相關。

由於這些特點，和受到自然語言處理 (NLP) 中提出的注意力 (Attention) 機制 [25] 的啟發，阿里巴巴提出了 Deep Interest Network (DIN)[23] 這個模型架構。DIN 利用類似注意力機制的方式來實現對使用者興趣的捕捉。DIN 針對當前目標商品，局部的激活使用者的歷史興趣，給予和目標商品相關度最高的歷史興趣 (商品) 較高的權重，藉此實現局部商品興趣聚焦的特性，而權重隨著目標商品不同也會有所不同，這個展現了使用者興趣多樣性的特點。

DIN 模型架構如圖3.3。此模型的輸入包含使用者資訊、歷史興趣和候選 (目標) 商品資訊，這些資訊皆會經過嵌入層 (Embedding Layer)，產出每個資訊所代表的向量。擁有這些向量之後，將這名使用者的所有歷史興趣向量，跟要推薦的候選商品，兩兩之間做局部機活並產生一個激活權重 (Activation Weight)，如圖3.3最右邊的激活單元 (Activation Unit)，這個激活權重就代表此歷史興趣與目標商品的相關性權重。經過這個激活單元產出激活權重後，將這些權重當作加權系數與相對應的歷史興趣的向量做加和池化 (sum pooling)，得到一個新的向量。最後將使用者向量、加和池化後的向量和目標商品向量合併，當作全連接層 (fully connected layers) 的輸入去訓練這個網路，最終得到一個介於 $[0,1]$ 之間的值，這個值就代表使用者會點擊我們推薦的這個候選商品的機率。

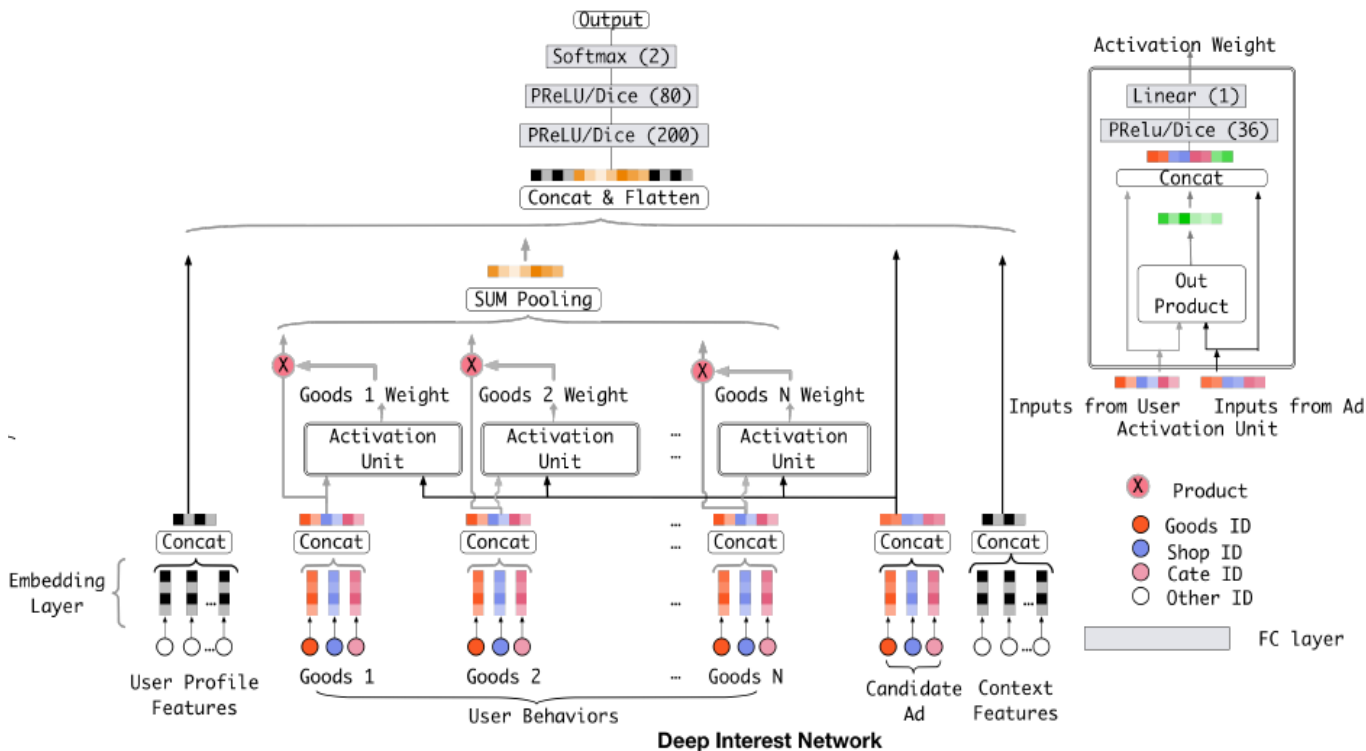


圖 3.3: DIN[23] 模型架構

3.3.2 Deep Interest Evolution Network (DIEN)

前一小節提到的 DIN 模型，雖然可以捕捉候選商品與歷史興趣的相關性，卻沒有考慮到使用者興趣的動態變化，使用者的興趣會隨著時間的不同而產生改變，可以從歷史的行為中去捕捉。因此阿里巴巴集團又提出了一個新的模型架構——Deep Interest Evolution Network(DIEN)。此模型架構跟 DIN 模型主要的差異在兩個部分：興趣提取層 (interest extractor layer) 和興趣演化層 (interest evolving layer)，模型架構如圖3.4。

興趣提取層主要目的，是提取按時間排序的歷史興趣中，歷史興趣之間的依賴性和相關性。將每個歷史興趣向量和要預測的候選商品向量當作輸入，利用 GRU 對時間序列資料的特性，來保留更早期歷史興趣的資訊對後期歷史興趣之間的相關性和演化。興趣演化層的部分，透過注意力機制，來計算歷史興趣跟要預測的候選商品之間的相關性，並透過作者提出的注意力更新機制 GRU(AUGRU) 來對相關性高的歷史興趣增加影響程度，讓相關性低的歷史興趣減弱其影響力。

最後再將這些提取出來的向量與其他資訊的向量合併，並利用深度學習網路訓練，最後輸出是使用者會點擊這個預測候選商品的機率。

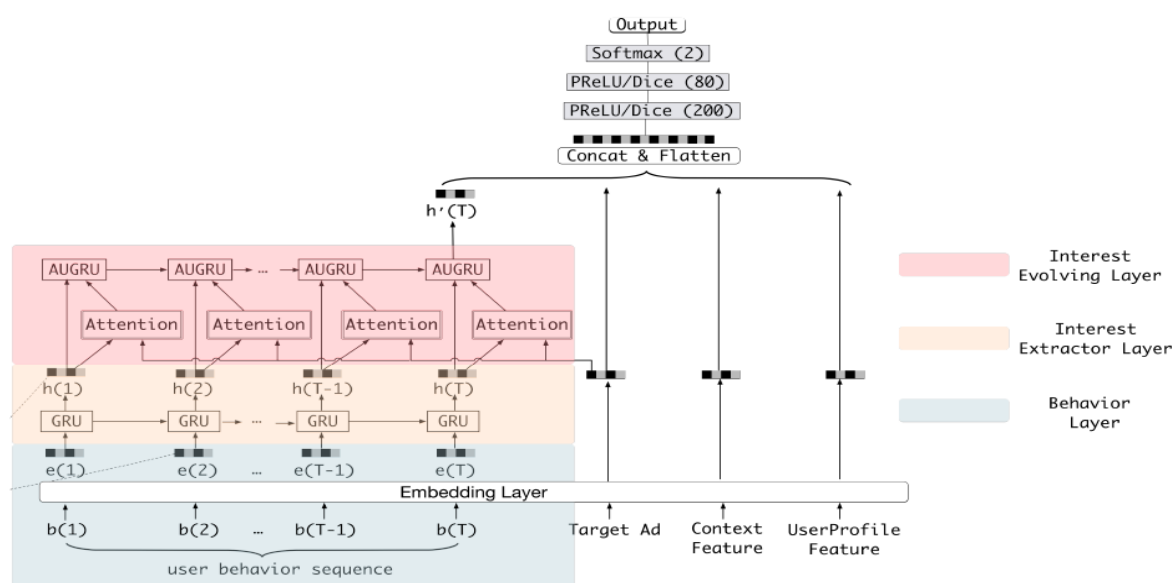


圖 3.4: DIEN[24] 模型架構

3.4 融合多模型排序結果

經過前述的四個模型的訓練與預測，我們已知的資料內容增加了四個模型重排過後，目標商品在當前商品候選九十九位中排名的名次。有了這些資訊後，再加上目標商品和當前商品在訓練資料集中出現的次數，我們再去訓練一個新的模型，來學習在不同排名和不同出現次數的狀況下，目標商品應該要排在當前商品的候選商品中的哪個位子。

在前面四個模型測試結果中我們發現，當前商品出現次數變多的時候，原本表現不好的模型會變好，預測目標商品排名的準確率會跟著提高；相反的，原本表現好的模型就會輸給原本表現不好的模型。我們透過這樣的發現，將四個模型的測試資料排名結果與商品的出現次數，來當作這部分實驗的訓練資料，如表3.2。

表3.2中，每個當前商品對應真正有被點擊過的目標商品，為一個正樣本資料，會標記為 1；則在從當前商品的候選商品中，隨機挑選一個沒有被點擊過的商品來當作負樣本，標記為 0。以這樣的資料去訓練模型，讓模型學習在出現次數和排名的影響下，應該要讓目標商品排序在推薦序列的前面，然後將沒被點擊過的候選商品排序在後面一點的位子，最後獲得一個目標商品的點擊機率。

表 3.2: 訓練資料特徵資訊

當前商品出現次數	目標商品出現次數	knn 排名	MLP 排名	DIN 排名	DIEN 排名
4	580	2	1	5	46
55	505	73	36	54	11
2	5	20	96	64	31

我們嘗試了許多機器學習的方法。我們將訓練的過程當作是分類問題：在點擊當前商品之後，接著會點擊目標商品的話，標準答案會標為 1；若點擊當前商品後，不會點擊目標商品，則答案標為 0。我們使用的方法如下：

- 1) 決策樹 (Decision Tree)
- 2) 梯度提升決策樹 (XGBoost)
- 3) 邏輯式迴歸 (logistic regression)

訓練完模型後，將我們每筆測試資料處理成如下圖3.5的樣子，將每筆當前商品的候選九十九個商品召回，並將每個當前商品與候選商品當作一筆測試資料，輸入到模型中並獲得一個候選商品被點擊的機率，最後再透過這九十九個機率做重新排序，得到一個新的推薦序列。

		當前商品 出現次數	目標商品 出現次數	KNN 排名	MLP 排名	DIN 排名	DIEN 排名
商品A · 商品B		4	580	2	1	5	46
99 個 候 選 商 品	商品A · 商品B	4	580	2	1	5	46
	商品A · 商品C	4	5	20	96	64	31
	商品A · 商品G	4	505	73	36	54	11
	⋮						

圖 3.5: 測試資料處理後格式

四、實驗設置與結果

此章節說明我們如何前處理資料，並說明為何選擇這些特徵來當我們的訓練模型的依據，最後將評估我們模型，進一步比較我們提出的新排序方法及其他基準模型的排序方法。透過預測實驗，我們最後的融合模型方法重新排序候選九十九個商品後，相較於其他基準模型，真正被點擊過的目標商品的排名有明顯幅度的往前。接下來依序介紹我們使用的資料集、實驗的設置、對照的模型方法及實驗結果。

4.1 資料集介紹

本篇論文使用的資料集有兩個：天池上公開的阿里巴巴集團旗下淘寶資料集¹和台灣某電商資料 (簡稱電商 A 資料集)。

如表4.1所示，淘寶資料集總共有九天的使用者使用資訊，而電商 A 資料集總共有四個月的使用者使用資訊。可以看到資料量是電商 A 的數量較多，但不同商品個數則是淘寶資料集較多。

表 4.1: 資料集內容介紹

	淘寶資料集	電商 A 資料集
資料涵蓋年份月份日期	2017/11/25-12/03	2019/03/01-06/30
資料量	大約 10 萬	大約 75 萬
不同商品個數	大約 150 萬	大約 35 萬

¹<https://tianchi.aliyun.com/dataset/dataDetail?dataId=649>

4.1.1 電商 A 資料集

電商 A 資料集資料時間涵蓋範圍從 2019 年 3 月到 6 月，總共四個月的使用者行為紀錄。內容包含使用者編號、使用者點擊過的商品編號、使用的裝置編號、時間戳記和行為類別，如表4.2所示。

表 4.2: 電商 A 資料資訊欄位

	例子
使用者編號	47b1ea66b86711e999fdacde48001122
裝置編號	fd7c495b-865f-4381-8c67-64dea1d86c4a
行為類別	pageload
瀏覽畫面類型	product-detail
商品編號	e33b661eb86611e98276acde48001122
時間戳記	2019-03-06T17:22:53.206

資料集中，除了使用者的行為紀錄外，還包括了商品的商品向量和商品類別編號。商品向量是電商 A 利用更多以前的使用者行為，事先訓練好的商品向量，我們認為這樣的向量能清楚的表達商品的特徵，所以在訓練 KNN 模型和 MLP 模型的過程中，我們所使用的都是電商 A 提供的商品向量。

4.1.2 阿里巴巴集團旗下淘寶資料集

由於我們參考的基準模型 DIN[23] 有使用到阿里巴巴資料集，但在論文中並沒有提到使用的是淘寶資料集或是天貓資料集，所以我們在阿里巴巴集團旗下的阿里雲網站裡的天池數據集²中，找到資料內容說明較為清楚的淘寶資料集。

此資料集包含 2017 年 11 月 25 號到 12 月 3 號之間這 9 天使用者對商品的各種行為。每一筆資料中，包括了如表4.3中的資訊。

²<https://tianchi.aliyun.com/>

表 4.3: 淘寶資料資訊欄位

	例子
使用者編號	1
商品編號	2268318
商品類別編號	2520377
行為類別	pv
時間戳記	1511544070

因為淘寶資料集中的資訊只有使用者對商品的互動行為，沒有代表商品特徵的商品向量，所以我們使用 Gensim 套件³和淘寶一天的資料來訓練淘寶商品的商品向量。Gensim 是從原始的非結構化的文本中，無監督式地學習文本中隱藏的主題向量表示，我們將這樣的想法套用到我們的商品序列中，並產生商品的向量表達式。

4.2 實驗資料配置與前處理

4.2.1 資料集介紹與前處理

在實驗開始之前，我們花了很多時間在資料處理的部分，為了讓兩份資料在我們的實驗需求下有一樣的條件，我們事先做了一些過濾和重新排序，將資料整理過後才進行實驗的部分。

- 1) 電商 A 資料集：我們將資料中，行為類型為 page load 和瀏覽畫面類型為 product detail 的互動行為留下，最後再以一天為單位，每天中的相同裝置編號所點擊過或互動過的商品，我們將這些商品按照時間戳記順序串在一起，成為一條相同裝置使用者所點擊過的點擊流 (click stream)，這樣一條一條的資料就是我們實驗的輸入。
- 2) 淘寶資料集：淘寶資料集中，每個使用者點過的一個商品，會被當作一個行為，我們將同一天中，行為類型為 page view(pv) 的保留，將這

³<https://radimrehurek.com/gensim/models/word2vec.html>

些保留下來的行為，同樣的以一天為單位，將同個使用者在同一天中瀏覽過的商品按照時間順序串接起來，成為淘寶資料集中的點擊流。

如圖4.1所示，箭頭左邊是使用者的點擊紀錄，圖中有兩位使用者在同一天不同時間的點擊紀錄，我們將同個使用者點擊過的商品，按照時間順序串接在一起，形成圖中箭頭右邊的樣子，同個使用者在同一天的紀錄最後只會有一條點擊序列。

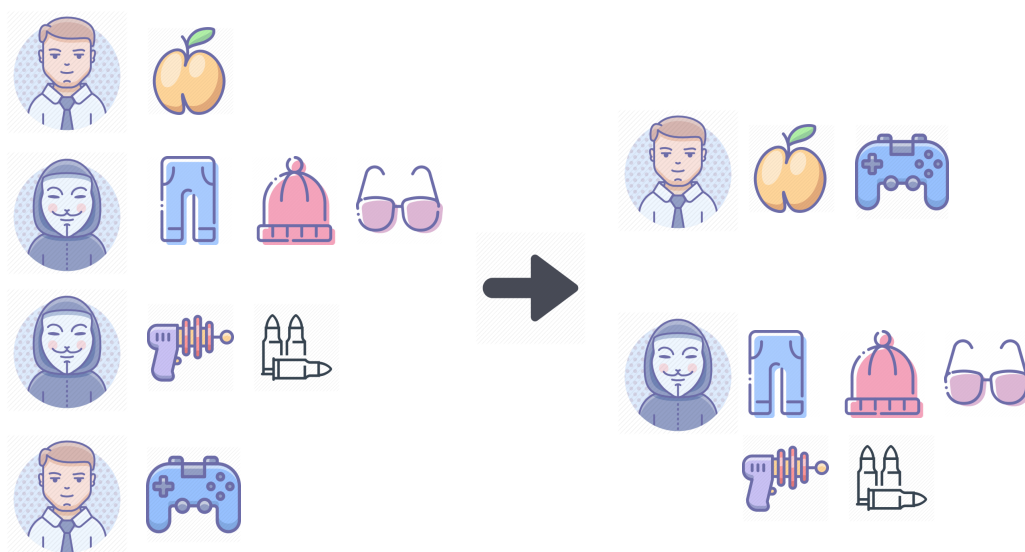


圖 4.1: 資料前處理範例

4.2.2 四個基礎模型實驗介紹

我們實驗方法總共有四個基準模型：KNN 模型、MLP 模型、DIN 模型和 DIEN 模型。四個模型中又分為兩類，KNN 模型和 MLP 模型的訓練方式較為相同；DIN 模型和 DIEN 模型訓練方式較為相同。KNN 模型的輸入，是當前商品和其餘的全部商品，兩兩成對，經過兩個商品向量的餘弦相似性計算，最後輸出這兩個商品之間的相關程度 (介於 $[0,1]$ 之間的值)。藉由 KNN 模型，可以找與當前商品最相似候選九十九個商品，之後其他模型所使用的候選九十九個商品，都是按照 KNN 模型所算出來的這些相似商品為代表。

得到候選商品後，MLP 模型的輸入就變成：當前商品向量和目標商品 (候選商品其中之一) 的向量，如圖4.2所示，將這兩個向量輸入到模型中，最後輸出點擊當前商品後會點擊目標商品的機率，藉由這個分數，將九十九個候選商品做排序，得出新的排名。

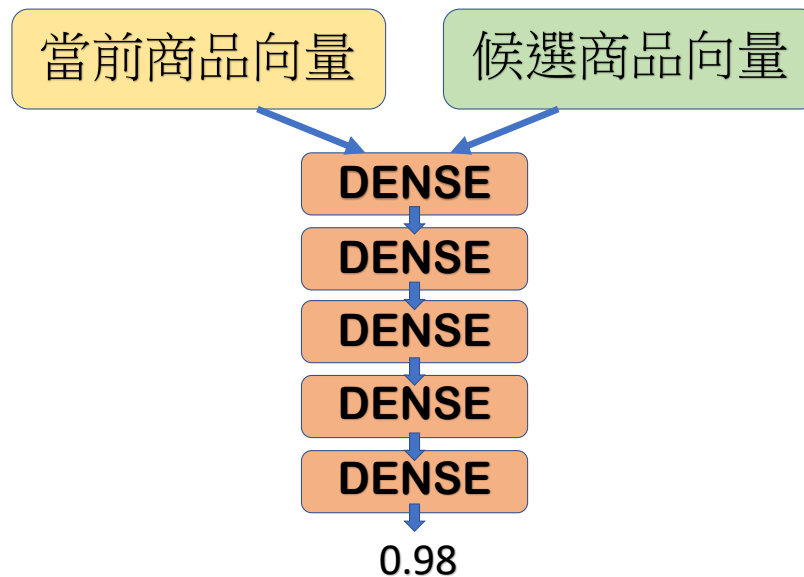


圖 4.2: MLP 訓練過程

由上面敘述可以發現，KNN 模型和 MLP 模型都是單純的透過目前點擊的商品向量和要預測會不會點擊的商品向量之間的關係，來判斷目標商品應該安排在候選商品中的第幾名，沒有其他使用者資訊或是歷史行為來訓練模型，單純的使用點擊對 (clicked pair) 來學習點擊某個商品後，點擊其他商品的機率會是多少。如圖4.2的輸入只有兩個商品的向量，透過 MLP 模型之後得到一個機率，而輸入就是前面所說的點擊對。

而 MLP 模型和 DIN、DIEN 模型主要的差異在於，DIN 和 DIEN 模型訓練的時候，輸入有包含使用者的歷史資訊 (點擊流)，但 MLP 模型沒有，雖然皆為深度學習模型，但在輸入的時候 MLP 模型選擇與 KNN 模型相同，DIN 模型和 DIEN 模型相同，藉由模型輸入的不同，間接比較在同樣深度學習模型中，輸入的不同會對結果造成什麼樣的影響。以下介紹 DIN 模型和 DIEN 模型的訓練方式。

經過 KNN 模型算出來的候選九十九個商品，會做為我們後面模型訓練的依據。在訓練 DIN 模型和 DIEN 模型時，我們所使用的九十九個候選商品，就是 KNN 模型算出來的。

DIN 模型和 DIEN 模型的輸入皆是相同的。輸入的一筆資料中，包含使用者編號、互動過的商品編號的序列、商品類別編號的序列、目標商品的編號和最後目標商品會不會被點擊的答案 (1 或 0)。將這些使用者特徵和目標商品特徵輸入 DIN 模型或 DIEN 模型後，會經由嵌入層得到一個代表各個特徵資訊的向量，再接著讓目標商品對歷史序列做注意力機制的訓練，如圖3.3，得到各個特徵新的向量後，再合併輸入到全連接層做訓練，最終得到使用者會不會點擊目標商品的機率。

經過這樣的訓練結果，我們將所有當前商品對應的九十九個候選商品，都去做預測並得到會不會被點擊的機率，再藉由這些機率對候選商品排序，排序後的序列就是這個模型對點擊此當前商品的使用者的推薦序列。

4.3 評量指標

我們使用「平均排名」來當作我們評量模型的指標。定義如下：

每一筆測試資料中，都會有當前商品和目標商品，每一筆資料經過這些模型預測後，會產出一個目標商品在當前商品候選九十九個商品中的新排名。將這些同個模型預測出來的新排名，全部加總之後平均，就是我們的評量指標——平均排名，如式子4.1所示。

$$Average_K = \frac{\sum_i^N rank_i}{N} \quad (4.1)$$

式子4.1中 K 包含 KNN、MLP、DIN、DIEN, N 表示 K 這個模型在測試資料中的所有排名的數量， $rank_i$ 表示排名的名次。最後 $Average_K$ 表示 K 這個模型在測試資料的平均排名。

4.4 實驗結果

實驗結果分兩階段：第一階段是由四個基準模型，對淘寶資料集第一天的資料和電商 A 資料集前 14 天的資料中使用者的歷史序列來做訓練，透過單純的使用者資訊、商品資訊和類別資訊等等，來對當前商品的候選九十九個商品做排序，最後再將真正有被點擊的目標商品的排名取出，並將同個模型的排名們做平均，最後得出此模型在這個資料集上的平均排名。在第一階段訓練完後，會利用淘寶的後 8 天資料和電商 A 後 108 天資料做預測，並將有名次的目標商品資料記錄下當作下一階段要訓練的資料。第二階段在前述四個基準模型訓練完畢並有測試資料的平均排名之後，我們將前一階段的前半部測試結果當作訓練資料，後半部當作測試資料，如圖4.3，利用這些資料中的平均排名和商品出現次數，來做第二個模型的訓練，讓第二個模型學到第一階段模型的排名與商品出現次數相關性，透過這樣的訓練，再將候選商品重新排序，獲得一個平均排名更好的模型。

第一階段



第二階段



灰色的部分為第一階段使用的訓練資料
第二階段不會使用到這個資料

圖 4.3: 資料切割說明

4.4.1 基礎模型在對應資料集之排序結果

實驗結果如表4.4和表4.5所示。可以清楚的看到，不管在哪個資料集上，我們所建構的 MLP 模型，排名都比 DIN 模型和 DIEN 模型好，由此可證明，在不一樣的資料集下，不一定複雜的深度學習模型會表現得比較好。

透過上面的表4.4和表4.5可以看到 KNN 模型和 MLP 模型的排名較為接近，DIN 模型和 DIEN 模型較為接近。KNN 模型和 MLP 模型輸入只有商品的向量，卻能在排名的時候做得比其他兩個較複雜的深度學習模型好，我們認為是因為我們資料集不管是在資料量上，或是特徵數量上都不夠大量，以致於當 DIN 和 DIEN 模型在訓練時，還無法更準確地抓到資料集的特性。

表4.6和表4.7呈現不同模型對目標商品排序的命中率。表中 Top-5 表示模型將目標商品排序在候選商品中前 5 名的位子的個數佔全部資料總數的比重。可以看到兩個資料集在 KNN 模型中的表現都是相當突出的，幾乎有一半的資料結果都將目標商品排序在前 20 名的位子。因為在排序候選商品時，通常不會有太多使用者會看完所有推薦的候選商品，所以我們特別將前 5、10、15 和 20 名的比重呈現出來，來證明透過簡單的 KNN 模型，就能將目標商品排序在很前面的位子，但深度學習模型雖然在前 5 名的時候命中率很低，但在前 20 名的比重也是有 20% 的，等於在考慮不同狀況下的目標商品和當前商品關係時，深度學習模型有時也會有被參考的價值。

下面展示當前商品在訓練資料中出現的次數和最後目標商品的排名之間的相關性。圖4.4和圖4.5展現兩個資料集在四個基準模型上的排名表現和當前商品出現次數的關係圖。可以看到 MLP 模型在當前商品出現次數和目標商品排名之間的關係非常明顯，當商品數量出現較多次時，目標商品就會排名在較前面的位子，此結果可以說明，當前商品出現次數較多的時候通常能預測的較準確，然而 KNN 模型當當前商品出現次

數較多時，目標商品的排名卻會往後掉。而 DIN 模型也和 MLP 模型有一樣的關係趨勢，幅度較小的原因我們認為可能是因為深度學習模型中還無法對們排名關係做很正確的學習，或是所有排名在 DIN 模型中較平均，而在 MLP 模型中排名的分布範圍比較大。而 DIEN 模型也是如此，比起 DIN 模型，趨勢線下降的幅度又更小了。

表 4.4: 淘寶之平均排名

	KNN	MLP	DIN	DIEN
平均值	30.02	31.93	47.80	50.17
標準差	28.04	27.83	27.99	28.83

表 4.5: 電商 A 資料集之平均排名

	KNN	MLP	DIN	DIEN
平均值	20.31	20.23	44.34	46.77
標準差	24.19	22.45	27.14	30.26

表 4.6: 淘寶在各模型 top-n 排名命中率比重

	KNN	MLP	DIN	DIEN
Top-5	24%	20%	5%	6%
Top-10	35%	30%	10%	10%
Top-15	43%	39%	15%	15%
Top-20	50%	46%	21%	20%

表 4.7: 電商 A 在各模型 top-n 排名命中率比重

	KNN	MLP	DIN	DIEN
Top-5	38%	33%	5%	8%
Top-10	52%	48%	12%	17%
Top-15	61%	58%	18%	24%
Top-20	67%	65%	25%	29%

我們將 KNN 和 MLP 這兩個模型的趨勢線放大來看，如圖4.6和圖4.7，可以清楚的看到在淘寶的資料集中，MLP 模型的紅色點點結果明顯的在隨著當前商品出現次數增多的時候，排名有漸漸往前的趨勢，但相較之下 KNN 模型就沒有明顯網上或下趨勢，藍色點點的分布也很分散。

透過這樣的實驗結果，我們可以更明確的知道，雖然在平均排名上，MLP 模型結果不如 KNN 模型，但如果仔細的分析其中的排名與出現次數關係，會發現在當前商品出現較多時，利用 MLP 模型可以將真正被點擊過的目標商品排在較前面的位子，而 KNN 模型反而在這時候會表現得不好。由此實驗結果，我們就發想了一個重新排序的方法：如果我們將這四個基準模型的測試結果當作訓練資料，或許能學到在不同時機能使用不同基準模型的排名法則，讓平均排名能超越這四個基準模型。

四、實驗設置與結果

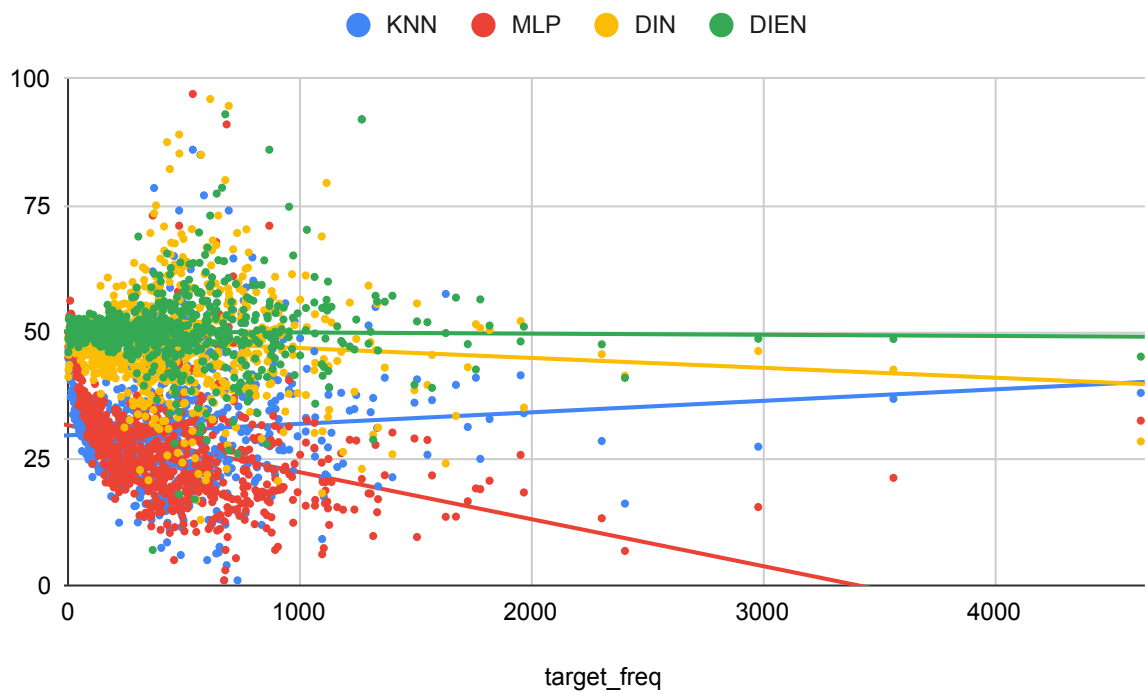


圖 4.4: 淘寶平均排名與商品出現次數之關係圖

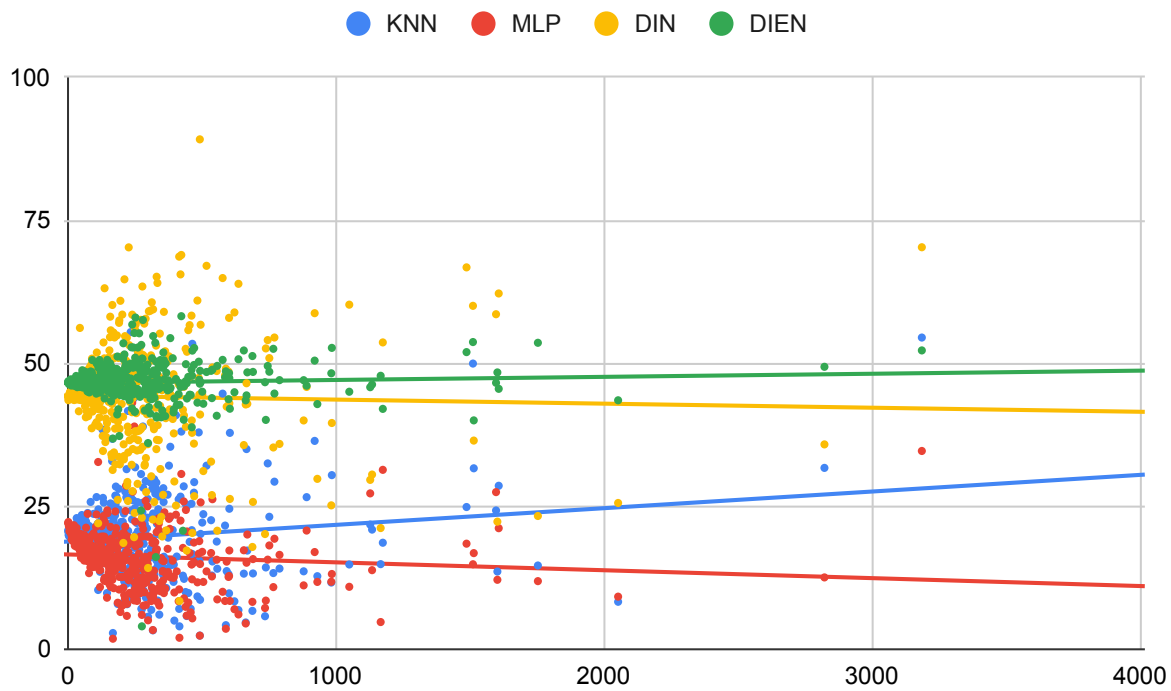


圖 4.5: 電商 A 平均排名與商品出現次數之關係圖

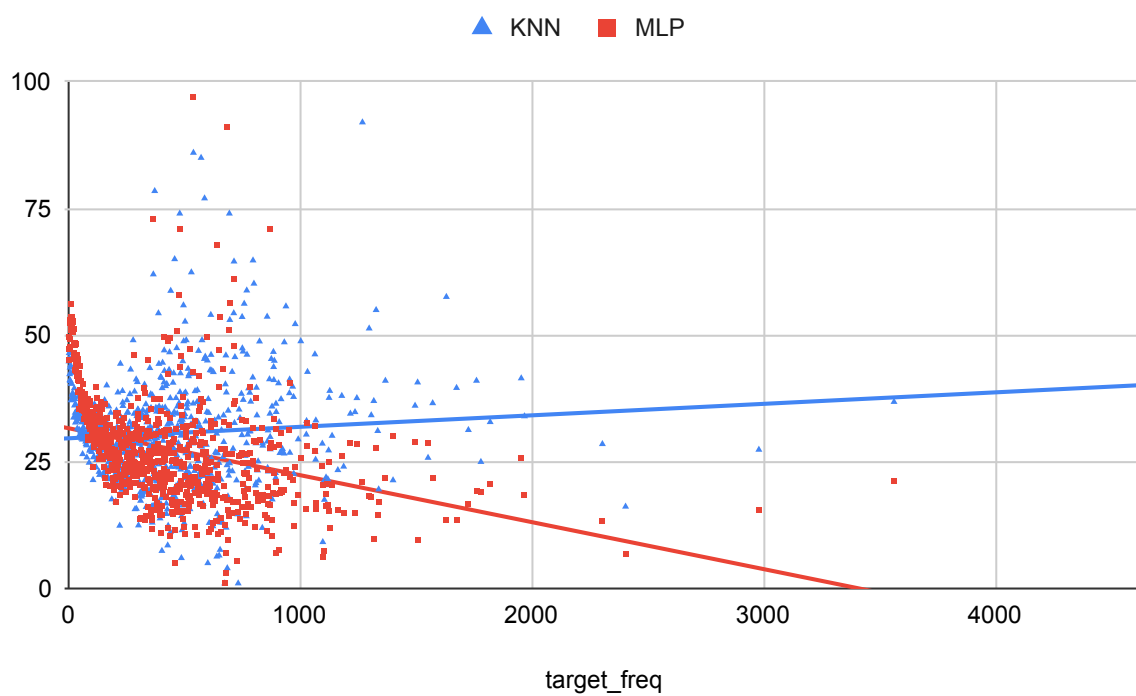


圖 4.6: 淘寶在 KNN 和 MLP 模型下平均排名與商品出現次數之關係圖

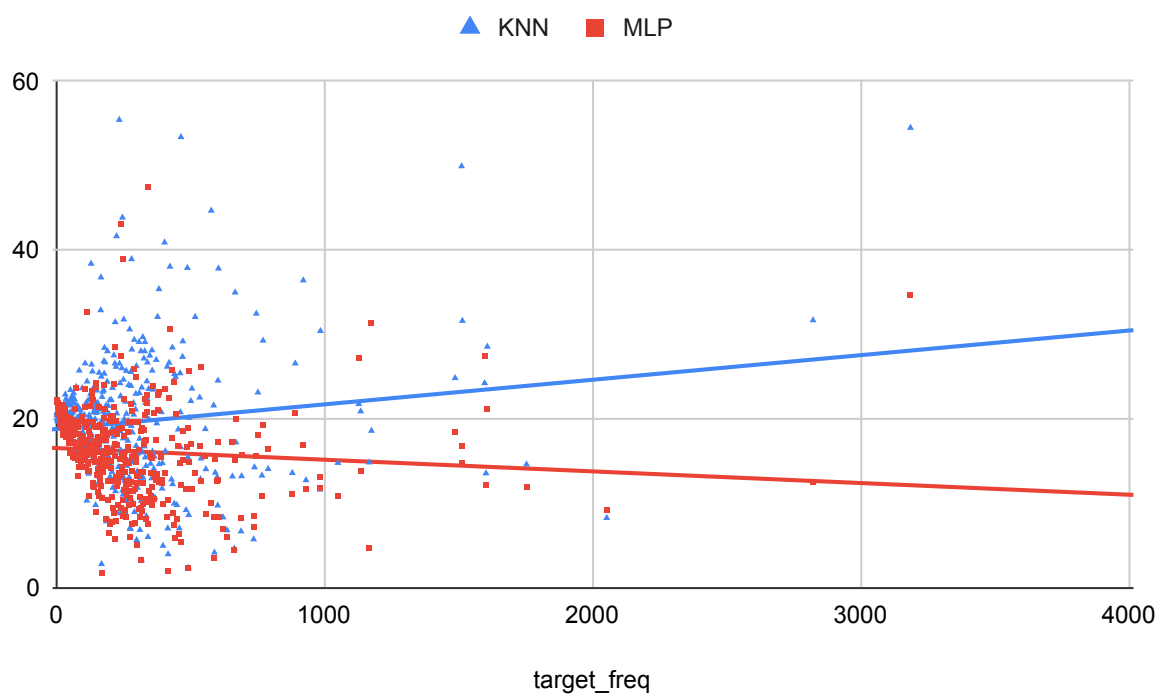


圖 4.7: 電商 A 在 KNN 和 MLP 模型下平均排名與商品出現次數之關係圖

4.4.2 參考基礎模型排序資訊之重新排序候選商品結果

透過前面四個基準模型的實驗，我們得到兩個資料集在四個模型中的排序，藉由這些排序，我們再去做一次新的模型訓練，藉此得到更好的排名。最後的實驗結果如下表4.8和表4.9。可以看到在平均排名方面，融合模型的結果比四個基準模型都還要更有效的將真正被點擊過的目標商品，排在九十九個候選商品中較前面的位子，透過學習四個模型在不同狀況下的表現，來判斷怎麼樣的狀況下，哪種基準模型的排名應該更為重要、更有參考價值，並透過出現次數的特徵，來決定應該將排名排在哪個位子。

表 4.8: 淘寶在融合模型預測平均排名與標準差

	決策樹	XGBoost	邏輯式迴歸
平均值	25.44	25.32	23.04
標準差	24.43	24.80	24.10

表 4.9: 電商 A 在融合模型預測平均排名與標準差

	決策樹	XGBoost	邏輯式迴歸
平均值	24.76	21.61	18.24
標準差	28.01	23.12	19.68

表 4.10: 淘寶在融合模型 top-n 排名命中率比重

	決策樹	XGBoost	邏輯式迴歸
Top-5	23%	25%	30%
Top-10	39%	39%	44%
Top-15	49%	48%	54%
Top-20	57%	57%	61%

表 4.11: 電商 A 在融合模型 top-n 排名命中率比重

	決策樹	XGBoost	邏輯式迴歸
Top-5	28%	29%	31%
Top-10	49%	49%	48%
Top-15	60%	60%	60%
Top-20	65%	65%	69%

以整體的平均排名來說，透過融合模型訓練後的重新排序，不管商品的出現次數或原始排名狀況怎麼樣，排名明顯都有往前的趨勢，可以從表4.8和表4.9看到，標準差的範圍也比四個基準模型的範圍還要小，所以融合模型的目標商品排名都較為前面且分布稍微集中。

表4.10和表4.11呈現不同模型對預測目標商品的排名的命中率。Top-5 表示將目標商品排序在前 5 名的位子的個數佔全部測試資料的比重，

其餘的以此類推。可以看到不管是哪個資料集，邏輯式迴歸模型對目標商品排名的預測都相當不錯，將近 60% 的預測排名都在前 20 名，且都比四個基準模型預測的還要好。

淘寶資料集的預測結果分布如圖4.8、圖4.9和圖4.10所示，電商 A 資料集預測結果分布如圖4.11、4.12和4.13。不管在哪種模型的融合下，重排的結果都有呈現以下趨勢：當前商品在訓練資料中出現次數越多，目標商品排名就越往前的趨勢，尤其以邏輯式迴歸的排名更為前面，重排過後目標商品的名次幾乎都在前 20 名左右。

在第二階段的實驗結果中，我們認為，當前商品和目標商品在訓練資料出現次數越多的時候，越能讓目標商品的排名往前，而且只要數量多到一個程度，模型就會直接將目標商品排名在第一名的位子，這個結果顯示，若當前商品和目標商品在訓練資料中出現很多次，則會像推薦熱門商品一樣，將目標商品排在較前面的位子；但若當前商品或目標商品出現的次數不算多的時候，就會利用四個基準模型的排名與商品出現次數作分析，學習在這樣的出現次數下，應該更相信哪個模型的排名，並透過商品出現次數來對這四個基準模型的名次做更新。

四、實驗設置與結果

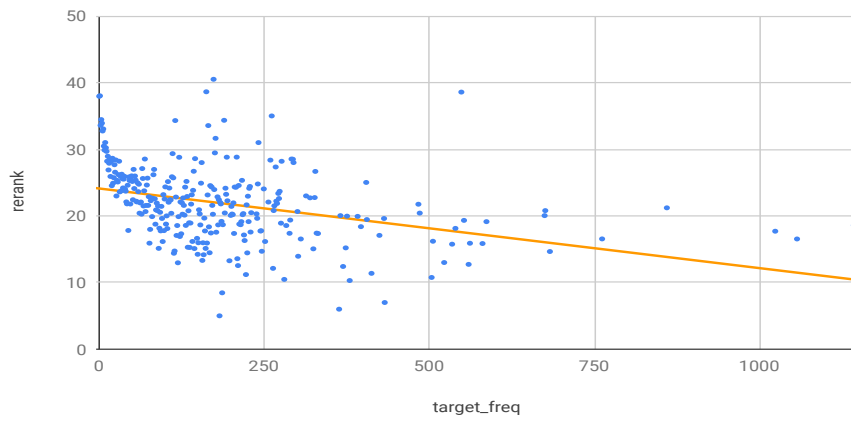


圖 4.8: 淘寶在利用決策樹融合模型下平均排名與商品出現次數之關係圖

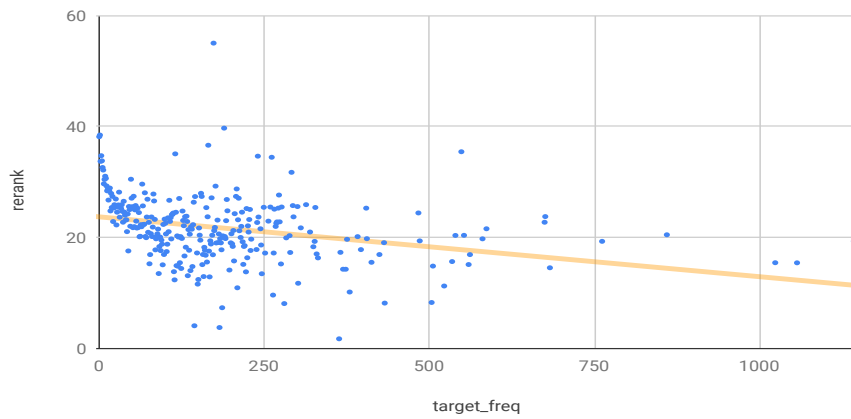


圖 4.9: 淘寶在利用 XGBoost 融合模型下平均排名與商品出現次數之關係圖

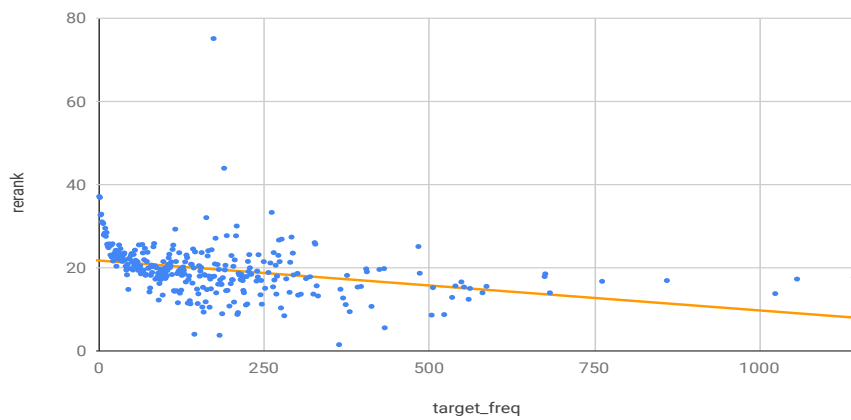


圖 4.10: 淘寶在利用邏輯式迴歸融合模型下平均排名與商品出現次數之關係圖

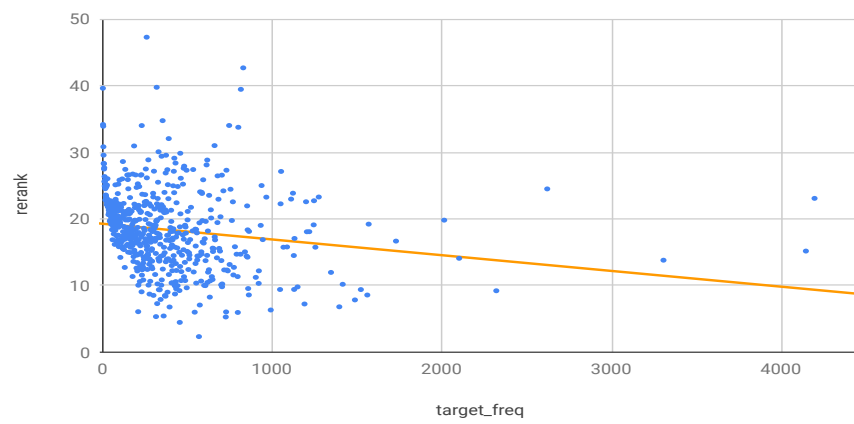


圖 4.11: 電商 A 在利用決策樹融合模型下平均排名與商品出現次數之關係圖

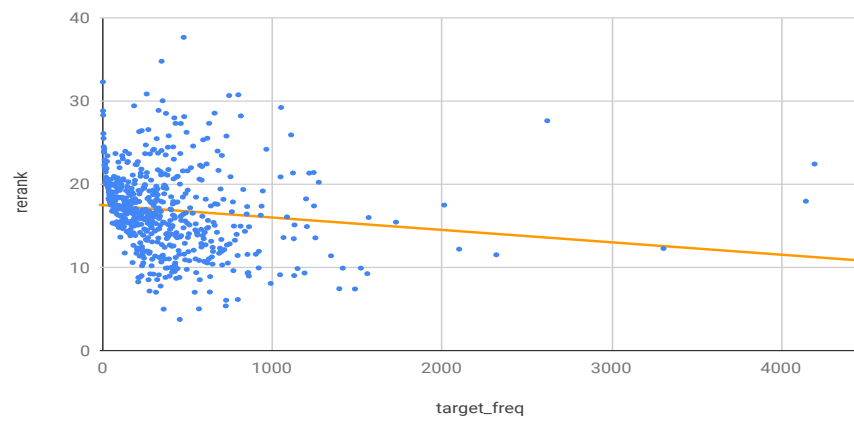


圖 4.12: 電商 A 在利用 XGBoost 融合模型下平均排名與商品出現次數之關係圖

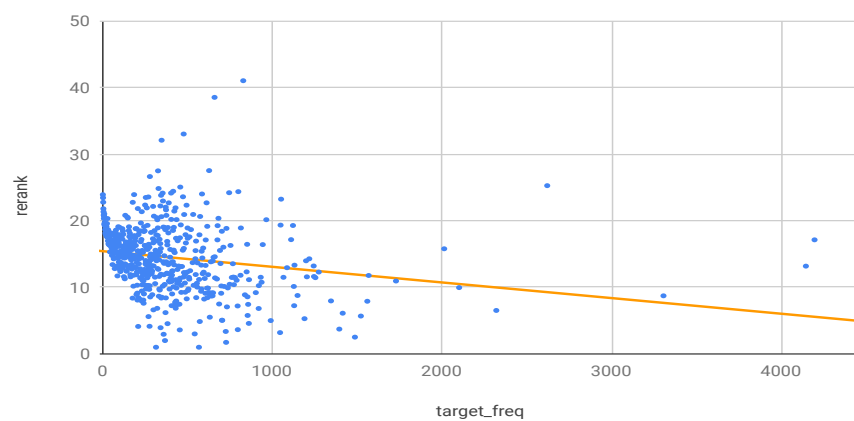


圖 4.13: 電商 A 在利用邏輯式迴歸融合模型下平均排名與商品出現次數之關係圖

四、實驗設置與結果

五、 結論與未來展望

5.1 結論

從我們的實驗結果，可以看出我們提出的想法跟我們的實驗結果是相符合的。在不同的電商資料集下，不同的基準模型下，目標商品在當前商品的候選九十九位中排名的平均名次都不太相同，又以 KNN 模型和 MLP 模型的平均排名較靠近；DIN 模型和 DIEN 模型平均排名較靠近。在四個基準模型的預測之下，目標商品的排名會隨著當前商品在訓練資料中出現的次數變多，而往前，在實驗結果的部分，也有透過趨勢線來呈現出現次數和目標商品排名的關聯性。

在以往的 Top-N 排序問題上，因為眾多資料龐大的資料集都是公開的，大部分的推薦系統模型都越來越複雜，想藉由這些知名電商網站的資料集來建立更有效的推薦系統，但卻在我們實驗中證明了不是所有電商網站都適合如此複雜的模型，或許簡單的模型就能達到很好的效果。

因為這樣的發現，我們決定將這些不同的基準模型的結果融合，透過這些基準模型的結果，建立一個參考這些基準模型的新排名。我們的實驗結果也有大幅度的進步。透過這些基準模型的資訊，與訓練資料中的資訊，我們不管在哪個資料集中，都有使平均排名往前，而且還超越基準模型中最好的平均排名。

5.2 未來展望

在四個基準模型的實驗中，我們使用兩個不同規模大小的電商資料集來做訓練，發現在訓練集資料數越龐大時，對於目標商品的排序會做得更好，但以四個不同模型在兩個不同資料集的表現來說，不管是哪個資料集，都以簡單模型 (KNN 模型和 MLP 模型) 表現的最好，複雜的深度學習模型 (DIN 模型和 DIEN 模型) 雖然應該在商品出現次數越多時，表現越好，但在我們的實驗中，在 Top-N 排序問題上平均排名都落後簡單模型。我們認為，由於我們使用的並非全部的淘寶資料，以至於在深度模型和簡單模型的差異和使用時機並沒有明顯得區別，所以若能將淘寶資料集的數量蒐集的更完整，或是使用更龐大的資料集，或許就能明顯的展現不同規模的電商資料集適用的推薦系統模型是有區別的，而且就算是在同個資料集下，商品出現的次數不同，也會影響應該使用的推薦模型。

在最後的融合多模型排序之模型訓練實驗中，我們發現當前商品出現的次數越多時，目標商品的排名也能有效的往前排 (排名更前面)，在這樣的實驗結果中，我們主要是使用當前商品出現次數、目標商品出現次數和目標商品實際在四個基準模型下的排序名次來訓練模型，在最後的訓練結果中，我們發現在最後的融合模型排序的訓練中，模型主要學到的是如何將目標商品出現次數多的商品排序在更前面的位子，雖然主要是捕捉到這樣的資訊，但也有發現到如果當前商品的次數出現的更多，對於目標商品排名的預測會越準確，但這樣的效果相比目標商品出現次數的影響，稍微小了一些，若能將模型做的更完善，更能捕捉到目標商品排序和當前商品之間的關係，或許能達到更好的效果。

參考文獻

- [1] M. F. Dacrema, P. Cremonesi, and D. Jannach, “Are we really making much progress? A worrying analysis of recent neural recommendation approaches,” *CoRR*, vol. abs/1907.06902, 2019. arXiv: 1907.06902. [Online]. Available: <http://arxiv.org/abs/1907.06902>.
- [2] A. M. Elkahky, Y. Song, and X. He, “A multi-view deep learning approach for cross domain user modeling in recommendation systems,” in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW ’ 15, Florence, Italy: International World Wide Web Conferences Steering Committee, 2015, pp. 278–288, ISBN: 9781450334693. DOI: 10.1145/2736277.2741667. [Online]. Available: <https://doi.org/10.1145/2736277.2741667>.
- [3] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recommendation,” in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys ’ 16, Boston, Massachusetts, USA: Association for Computing Machinery, 2016, pp. 233–240, ISBN: 9781450340359. DOI: 10.1145/2959100.2959165. [Online]. Available: <https://doi.org/10.1145/2959100.2959165>.
- [4] J. Manotumruksa, C. Macdonald, and I. Ounis, “A contextual attention recurrent architecture for context-aware venue recommendation,” in *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, ser. SIGIR ’ 18, Ann Arbor, MI, USA: Association for Computing Machinery, 2018, pp. 555–564, ISBN: 9781450356572. DOI: 10.1145/3209978.3210042. [Online]. Available: <https://doi.org/10.1145/3209978.3210042>.
- [5] N. Sachdeva, K. Gupta, and V. Pudi, “Attentive neural architecture incorporating song features for music recommendation,” *CoRR*, vol. abs/1811.08203, 2018. arXiv: 1811.08203. [Online]. Available: <http://arxiv.org/abs/1811.08203>.
- [6] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, “Recurrent knowledge graph embedding for effective recommendation,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, ser. RecSys ’ 18, Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, pp. 297–305, ISBN: 9781450359016. DOI: 10.1145/3240323.3240361. [Online]. Available: <https://doi.org/10.1145/3240323.3240361>.

- [7] Y. Tay, A. T. Luu, and S. C. Hui, “Translational recommender networks,” *CoRR*, vol. abs/1707.05176, 2017. arXiv: 1707.05176. [Online]. Available: <http://arxiv.org/abs/1707.05176>.
- [8] Y. Tay, L. A. Tuan, and S. C. Hui, “Multi-pointer co-attention networks for recommendation,” *CoRR*, vol. abs/1801.09251, 2018. arXiv: 1801.09251. [Online]. Available: <http://arxiv.org/abs/1801.09251>.
- [9] T. X. Tuan and T. M. Phuong, “3d convolutional networks for session-based recommendation with content features,” in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ser. RecSys ’17, Como, Italy: Association for Computing Machinery, 2017, pp. 138–146, ISBN: 9781450346528. DOI: 10.1145/3109859.3109900. [Online]. Available: <https://doi.org/10.1145/3109859.3109900>.
- [10] F. Vasile, E. Smirnova, and A. Conneau, “Meta-prod2vec - product embeddings using side-information for recommendation,” *CoRR*, vol. abs/1607.07326, 2016. arXiv: 1607.07326. [Online]. Available: <http://arxiv.org/abs/1607.07326>.
- [11] P. Covington, J. Adams, and E. Sargin, “Deep neural networks for youtube recommendations,” in *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016.
- [12] S. Kabbur and G. Karypis, “Nlmf: Nonlinear matrix factorization methods for top-n recommender systems,” vol. 2015, Dec. 2014. DOI: 10.1109/ICDMW.2014.108.
- [13] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, Jan. 2004, ISSN: 1046-8188. DOI: 10.1145/963770.963776. [Online]. Available: <https://doi.org/10.1145/963770.963776>.
- [14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th International Conference on World Wide Web*, ser. WWW ’01, Hong Kong, Hong Kong: Association for Computing Machinery, 2001, pp. 285–295, ISBN: 1581133480. DOI: 10.1145/371920.372071. [Online]. Available: <https://doi.org/10.1145/371920.372071>.
- [15] A. Tewari, J. Singh, and A. Barman, “Generating top-n items recommendation set using collaborative, content based filtering and rating variance,” *Procedia Computer Science*, vol. 132, pp. 1678–1684, Jan. 2018. DOI: 10.1016/j.procs.2018.05.139.
- [16] C. Cooper, S. Lee, T. Radzik, and Y. Siantos, “Random walks in recommender systems: Exact computation and simulations,” Apr. 2014, pp. 811–816. DOI: 10.1145/2567948.2579244.
- [17] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in. Jan. 2011, pp. 73–105. DOI: 10.1007/978-0-387-85820-3_3.

- [18] B. Paudel, F. Christoffel, C. Newell, and A. Bernstein, “Updatable, accurate, diverse, and scalable recommendations for interactive applications,” *ACM Trans. Interact. Intell. Syst.*, vol. 7, no. 1, Dec. 2016, ISSN: 2160-6455. DOI: 10.1145/2955101. [Online]. Available: <https://doi.org/10.1145/2955101>.
- [19] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Ou, “Behavior sequence transformer for e-commerce recommendation in alibaba,” *CoRR*, vol. abs/1905.06874, 2019. arXiv: 1905.06874. [Online]. Available: <http://arxiv.org/abs/1905.06874>.
- [20] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, “Variational autoencoders for collaborative filtering,” in *WWW*, P.-A. Champin, F. L. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds., ACM, 2018, pp. 689–698. [Online]. Available: <http://dblp.uni-trier.de/db/conf/www/www2018.html#LiangKHJ18>.
- [21] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, “Neural collaborative filtering,” *CoRR*, vol. abs/1708.05031, 2017. arXiv: 1708.05031. [Online]. Available: <http://arxiv.org/abs/1708.05031>.
- [22] H. Wang, N. Wang, and D. Yeung, “Collaborative deep learning for recommender systems,” *CoRR*, vol. abs/1409.2944, 2014. arXiv: 1409.2944. [Online]. Available: <http://arxiv.org/abs/1409.2944>.
- [23] G. Zhou, C. Song, X. Zhu, X. Ma, Y. Yan, X. Dai, H. Zhu, J. Jin, H. Li, and K. Gai, “Deep interest network for click-through rate prediction,” 2017.
- [24] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, “Deep interest evolution network for click-through rate prediction,” *ArXiv*, vol. abs/1809.03672, 2019.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: <http://arxiv.org/abs/1706.03762>.
- [26] Y. Feng, F. Lv, W. Shen, M. Wang, F. Sun, Y. Zhu, and K. Yang, “Deep session interest network for click-through rate prediction,” *CoRR*, vol. abs/1905.06482, 2019. arXiv: 1905.06482. [Online]. Available: <http://arxiv.org/abs/1905.06482>.
- [27] C. Cooper, S. Lee, T. Radzik, and Y. Siantos, “Random walks in recommender systems: Exact computation and simulations,” Apr. 2014, pp. 811–816. DOI: 10.1145/2567948.2579244.
- [28] T. Ebesu, B. Shen, and Y. Fang, “Collaborative memory network for recommendation systems,” *CoRR*, vol. abs/1804.10862, 2018. arXiv: 1804.10862. [Online]. Available: <http://arxiv.org/abs/1804.10862>.
- [29] B. Hu, C. Shi, W. Zhao, and P. Yu, “Leveraging meta-path based context for top-n recommendation with a neural co-attention model,” Aug. 2018. DOI: 10.1145/3219819.3219965.

- [30] X. Li and J. She, “Collaborative variational autoencoder for recommender systems,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’ 17, Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 305–314, ISBN: 9781450348874. DOI: 10.1145/3097983.3098077. [Online]. Available: <https://doi.org/10.1145/3097983.3098077>.