

A Food Recommendation System Based on Graph Neural Networks

張碩文
NCKU CSIE

p76134692@gs.ncku.edu.tw

蘇祐蓁
NCKU AIM

ne6131021@gs.ncku.edu.tw

陳子輝
NCKU CSIE

p76135062@gs.ncku.edu.tw

許漢權
NCKU IMI

q56135019@gs.ncku.edu.tw

Abstract

隨著近年線上外送平台如 *Foodpanda*、*UberEat* 逐漸成為主流，餐點推薦系統的重要性日益增加。有效的推薦系統不僅能夠精確地向使用者推薦合適的餐點，還可以幫助平台最大化收益並提升用戶體驗。本研究將提出一種基於圖神經網路 (Graph Neural Networks; GNN) 的推薦系統模型，通過分析平台上用戶與餐點之間的關聯性，從中萃取出有價值的資訊，進而提供更精準的餐點推薦。此模型將針對使用者的偏好與歷史行為進行學習，結合圖結構的數據特徵，實現更個性化且高效的推薦效果。最後，我們將對該模型進行實驗與評估，並與現有的推薦系統進行比較。

1. Introduction

隨著線上外送平台如 *Foodpanda* 和 *UberEat* 的迅速發展，餐飲產業的數位化轉型逐漸加快。這些平台不僅縮短了顧客與餐廳之間的距離，還利用大數據分析技術來優化用戶體驗並提升平台的營收效率。在此背景下，餐廳推薦系統扮演了關鍵角色。它通過分析顧客的歷史消費行為、偏好，以及當前的流行趨勢，實現個性化推薦，進一步提高了推薦的準確性和顧客的滿意度。傳統的推薦系統方法，如基於內容的過濾 (content-based filtering) 和協同過濾 (collaborative filtering)，能夠在一定程度上提供有效的推薦，仍難以捕捉到顧客與餐廳之間更深層的關聯結構。隨著圖神經網路 (Graph Neural Networks, GNN) 的應用擴展，人們發現可以將顧客與餐廳之間的互動關係表示為圖結構，以便從更高層次上捕捉其複雜的互動模式。這種關聯結構可以使用二分圖 (bipartite graph) 來表示。

如圖 1 所示，節點可以分為兩類：使用者節點和餐廳節點，兩者節點類型的不同使得該圖同時為異質圖 (attributed heterogeneous graph)。本研究使用 Zhang 等人提出的 NIE-GCN 方法 [2]，從該資料集的二分異質

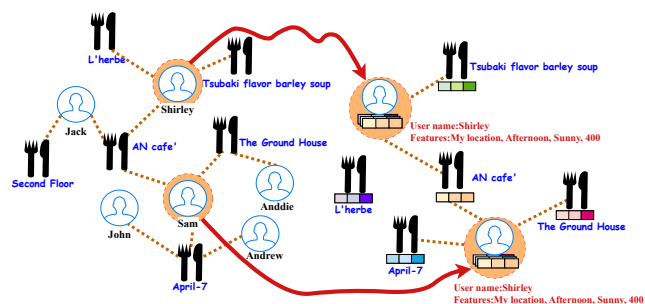


Figure 1. 推薦系統中的二分圖示意 [1]

圖中提取出更深層的連結信息，以預測顧客節點最感興趣的前 k 個餐廳節點，進一步優化推薦精度。最終以歸一化折扣累積增益 (Normalized Discounted Cumulative Gain, NDCG) 和召回率 (Recall) 作為評估推薦系統表現的指標，為平台提供精確的推薦評估依據。本

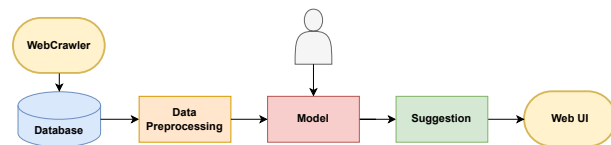


Figure 2. 推薦系統架構圖

研究的架構如圖 2 所示。首先，透過爬蟲技術從 *Foodpanda* 和 *Google Maps* 上獲取每個餐廳的資訊，並將這些資料儲存到資料庫中。接著進行基本的資料前處理，為模型的輸入做好準備。在使用者節點上，所包含的資訊如表 1 所列。這些資訊經處理後會被輸入到模型中，最終生成推薦結果，並透過 *Web UI* 進行呈現。透過這些節點資訊，本研究可以針對不同使用情境生成個性化推薦，讓系統在多維度考量下更精確地滿足使用者需求。

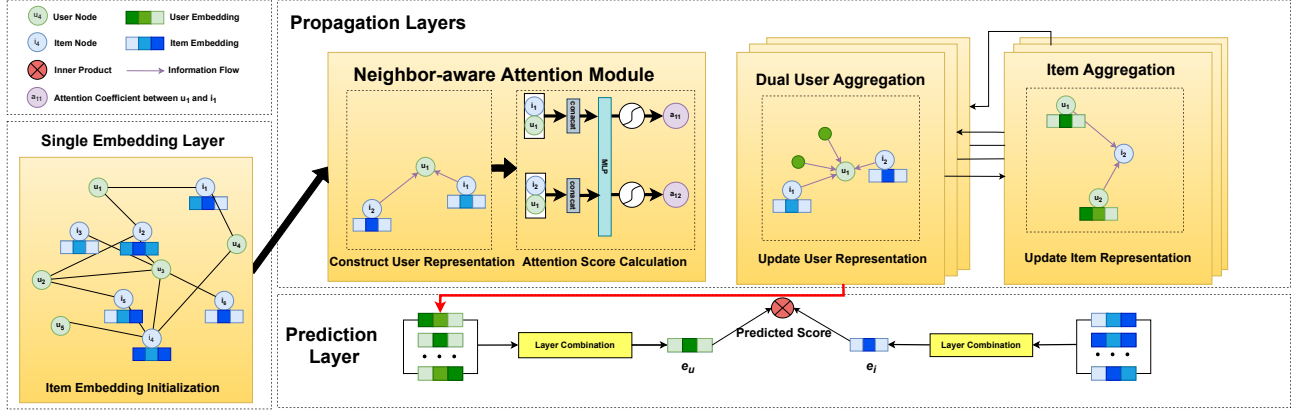


Figure 3. 模型架構圖

節點類型	節點內容
使用者	當下位置、使用時間、天氣、預算
推薦餐廳	餐廳名稱、價位、評分、評論

Table 1. 使用者與推薦餐廳的節點資訊

2. Proposed

2.1. Single Embedding Layer

本研究所採用的 NIE-GCN[2] 模型可拆解成幾個主要步驟如圖 3，首先是對餐廳節點進行向量嵌入。第一步會針對每個餐廳節點 i 生成一個嵌入向量 e_i ，此向量位於 d 維空間中，即 $e_i \in \mathbb{R}^d$ 。在嵌入過程中，所有餐廳節點的嵌入向量會被收集成一組矩陣 $E_I^{(0)}$ ，此矩陣表達了所有餐廳節點的初始嵌入狀態，如下式所示：

$$E_I^{(0)} = [e_{i_1}^{(0)}, e_{i_2}^{(0)}, \dots, e_{i_N}^{(0)}] \in \mathbb{R}^{N \times d}, \quad (1)$$

其中， N 代表餐廳節點的總數，而 d 則是嵌入向量的維度；上標 0 表示第 0 層傳播的初始狀態。藉由得到每個餐廳節點的嵌入向量後，模型便可利用這些向量作為基礎，來進一步推斷每位使用者的偏好。因為每個使用者節點 u 的鄰居必定為餐廳節點，故模型會透過與該使用者互動過的餐廳節點來推敲其個人喜好。

在完成餐廳節點的嵌入後，第二步則是建構每位使用者節點的初始嵌入向量 $e_u^{(0)}$ 。此過程需要依賴與使用者節點 u 相鄰的餐廳節點 i 的嵌入向量 $e_i^{(0)}$ 。第 0 層的嵌入構造方式如下：

$$e_u^{(0)} = \sigma \left(\sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} e_i^{(0)} \right), \quad (2)$$

其中 N_u 表示使用者節點 u 的鄰居集合，而 N_i 則代表餐廳節點 i 的鄰居集合； $\sigma(\cdot)$ 是激活函數，選擇使用雙曲正切函數 (tanh)。透過這一層的加權平均，可以有效地融合使用者與其鄰近餐廳節點的特徵信息，以更準確地反映該使用者的行為特徵。

2.2. Propagation Layers

完成初始嵌入後，下一步是計算每個使用者節點 u 與其相鄰的餐廳節點 i 之間的注意力分數 $\rho(u, i)$ ，藉此衡量不同鄰居節點的重要性。該分數計算方式如下：

$$\rho(u, i) = Q^T \sigma(W(e_u^{(0)} || (e_i^{(0)} + b))), \quad (3)$$

其中 $W \in \mathbb{R}^{2d \times d}$ 、 $Q \in \mathbb{R}^{1 \times d}$ 、 $b \in \mathbb{R}^{1 \times d}$ ，這三個參數分別為注意力機制中的權重矩陣與偏置項。此處的 $\sigma(\cdot)$ 同樣為雙曲正切函數 (tanh)，而 $||$ 則表示向量的串接操作。 $\rho(u, i)$ 的分數越高，代表使用者 u 與餐廳 i 之間的關聯性越強。因此，透過該分數可以量化每個餐廳節點對於預測使用者偏好的貢獻度。

為將注意力分數限制在 $[0, 1]$ 的範圍內，利用 Softmax 函數對這些分數進行範圍歸一化處理，得到最終的注意力值 $\alpha(u, i)$ 。公式如下：

$$\alpha(u, i) = \frac{\exp(\rho(u, i))}{\left(\sum_{j \in N_u} \exp(\rho(u, j)) \right)^\beta}, \quad (4)$$

其中 $\alpha(u, i)$ 代表了使用者節點 u 與餐廳節點 i 之間的最終注意力值。透過 $\alpha(u, i)$ ，每個使用者節點均能依據其鄰居的特徵，綜合考量注意力分數來完成向量更新。

在初始嵌入狀態完成後，為進行更深層的特徵傳播，使用者節點 u 的嵌入向量 $e_u^{(k)}$ 將在每一層根據其鄰居節點的資訊進行更新。使用者的嵌入向量 $e_u^{(k)}$ 的更新方式會依據先前計算出的注意力權重 $\alpha(u, i)$ ，並利用相鄰餐廳節點 i 在上一層的嵌入向量 $e_i^{(k-1)}$ 進行加法聚合 (Add Aggregation)。此聚合方式可以有效融合鄰居的特徵信息，更新的計算公式如下：

$$e_u^{(k)} = \sigma \left(\sum_{i \in N_u} \alpha(u, i) e_i^{(k-1)} \right), \quad (5)$$

其中， $\sigma(\cdot)$ 為雙曲正切函數 (tanh)， N_u 表示使用者節點 u 的鄰居集合。透過加權聚合，使用者節點可以自

適應地調整對不同鄰居特徵的重視程度，進一步增強模型對於不同使用者偏好的捕捉能力。

同樣地，初始狀態後的餐廳節點的嵌入向量 $e_i^{(k)}$ 也會根據其相鄰使用者節點的嵌入進行更新。這個更新方式與公式 (2) 類似，餐廳節點 i 的嵌入向量會根據其相鄰的使用者節點 u 的嵌入向量 $e_u^{(k)}$ 進行加權平均，計算方式如下：

$$e_i^{(k)} = \sigma \left(\sum_{u \in N_i} \frac{1}{\sqrt{|N_u||N_i|}} e_u^{(k)} \right), \quad (6)$$

其中， N_u 和 N_i 分別表示使用者節點 u 和餐廳節點 i 的鄰居集合，並透過加權平均來控制每個鄰居對於嵌入向量的貢獻。此公式 (6) 與公式 (2) 中的加權項 $\frac{1}{\sqrt{|N_u||N_i|}}$ 可以平衡不同鄰居數量對嵌入更新的影响，從而避免由於鄰居數量不均而引起的偏差。藉由此加權項，每個餐廳節點的嵌入向量都將根據與其相鄰使用者的特徵進行有效更新，從而更準確地反映餐廳與使用者間的潛在關係。

2.3. Prediction Layers

在完成每一層使用者節點的嵌入向量 $e_u^{(k)}$ 和餐廳節點的嵌入向量 $e_i^{(k)}$ 的計算後，接下來的步驟是將這些嵌入向量進行聚合，以獲得最終的嵌入表達，在本研究將所有層的使用者嵌入向量和餐廳嵌入向量分別相加，計算公式如下：

$$e_u^* = \sum_{k=1}^L e_u^{(k)}, \quad e_i^* = \sum_{k=1}^L e_i^{(k)}, \quad (7)$$

其中 L 表示嵌入層的總數。這樣計算得到的 e_u^* 和 e_i^* 分別代表了使用者節點和餐廳節點的最終嵌入向量，這兩個向量綜合了多層的信息，能夠更全面地捕捉使用者與餐廳之間的潛在關聯。為了進一步評估使用者 u 和餐廳 i 之間的關聯性，將這兩個最終嵌入向量進行內積操作，為了將內積值限制在 0 到 1 的範圍內，引入了 S 型函數 (sigmoid function)，計算公式如下：

$$\hat{y}_{ui} = f(u, i) = \sigma(e_u^{*T} e_i^*), \quad (8)$$

其中， \hat{y}_{ui} 表示模型對於使用者 u 和餐廳 i 之間關聯性的預測結果。S 型函數 $\sigma(\cdot)$ 將內積的結果映射到 $[0, 1]$ 的範圍，使得模型的輸出可以被解釋為一個機率值，表達使用者 u 對餐廳 i 的喜好程度。

2.4. Model Optimization

本研究模型的目標函數採用貝氏個性化排序 (Bayesian Personalized Ranking, BPR)，以實現使用者偏好排序的優化，具體公式如下：

$$\mathcal{L} = \sum_{(u,i) \in \mathcal{O}^+, (u,j) \in \mathcal{O}^-} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|_2^2, \quad (9)$$

其中 \mathcal{O}^+ 和 \mathcal{O}^- 分別表示正樣本與負樣本的集合；正樣本為符合使用者偏好的項目，而負樣本為非偏好項目。 $\sigma(\cdot)$ 是 S 型函數 (sigmoid function)，可將輸出的範圍限制在 $[0, 1]$ 之間，用以衡量正樣本和負樣本之間的預測差距。此目標函數的設計旨在使正樣本的預測值 \hat{y}_{ui} 高於負樣本的預測值 \hat{y}_{uj} ，即期望 \hat{y}_{ui} 越高越好，而 \hat{y}_{uj} 越低越好，從而增強模型對正負樣本的區分能力。

該目標函數包含了一項 L2 正則項，用於減少模型過擬合的風險。 λ 是 L2 正則化的係數， $\Theta = \{E, Q, W, b\}$ 表示模型中所有需要學習的參數集。通過引入正則項，模型的權重得以控制在合理範圍內，避免權重過大導致的模型複雜性增加。

在計算過程中，預測差距 $\hat{y}_{ui} - \hat{y}_{uj}$ 的值範圍在 $[-1, 1]$ ，經過 S 型函數 $\sigma(\cdot)$ 映射後，範圍被壓縮到 $[0, 1]$ 。為了使目標函數在最大化的情況下實現梯度下降 (gradient descent) 訓練，對公式添加負號，這樣在取對數 \ln 後，所有值為正，因此相當於最小化正的損失值，使得模型能夠通過梯度下降的方式進行有效優化。

3. Experiment

3.1. Data Preprocessing

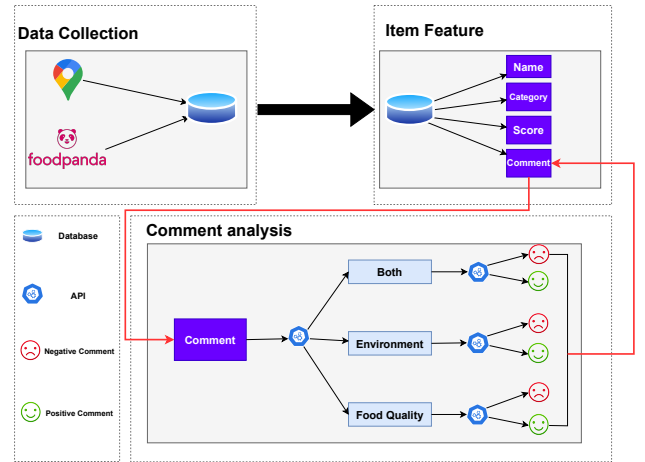


Figure 4. 資料前處理架構圖

本研究的資料前處理架構如圖 4 所示，透過爬蟲技術從 Foodpanda 與 Google Maps 收集餐廳相關資料。每間餐廳的資料包含餐廳名稱、類型、評分以及顧客評論。由於餐廳通常會累積大量評論，這些評論對於描述店家的服務品質、餐點特色與顧客體驗具有高度相關性。然而，若將所有評論直接整合後丟入 GPT-4 進行分析，評論主題的多樣性可能導致資訊雜亂，進而影響模型微調 (fine-tuning) 的效果。當不同主題的評論混合訓練時，模型可能難以準確識別每類特徵的意圖，進一步降低結果的精確性。

為解決上述問題並提升模型的針對性，本研究先將評論依主題分為三類：餐點品質、店內環境氛圍，以及綜合評價。針對這三類評論，分別進行獨立的 LLM 微調，以使每個模型專注於特定主題特徵，避免不同主題資訊的相互干擾。對於餐點品質的評論，模型學習重點為食物的口感、品質及外觀呈現；對於店內環境氛圍的評論，模型則專注於用餐環境、設計風格及舒適度的相關描述；而綜合評價的模型則著重於顧客的整體體驗。透過分主題進行微調，模型得以精準掌握各主題的特徵表現，進一步增強評論處理的針對性。完成主題分類及模型微調後，本研究針對各主題的評論進行情感分析，以判斷評論的情感取向（正面或負面）。主題分類與情感分類的結果分別以 one-hot encoding 表示。其中，主題分類的向量表示為：

$$\mathbf{h}_c \in \mathbb{R}^3, \quad (10)$$

其三個維度分別對應於餐點品質 (c_1)、店內環境氛圍 (c_2) 以及綜合評價 (c_3)。情感分類的向量表示為：

$$\mathbf{h}_s \in \mathbb{R}^2, \quad (11)$$

其中兩個維度分別對應正面情感 (P) 與負面情感 (N)。將主題分類與情感分類的 one-hot encoding 串接後，得到評論的最終向量表示：

$$\mathbf{h}_{i,j} = [\mathbf{h}_c; \mathbf{h}_s] \in \mathbb{R}^5, \quad (12)$$

其中 $[\cdot; \cdot]$ 表示向量的串接操作， $\mathbf{h}_{i,j}$ 代表第 i 家餐廳中第 j 條評論的綜合向量表示。

假設某店家 i 有 n_i 條評論，且每條評論 j 的嵌入向量為 $\mathbf{h}_{i,j}$ ，本研究為評論賦予基於主題分類的權重 $w(c_{i,j})$ ，其設定如下：

$$w(c) = \begin{cases} w_1, & \text{若 } c = c_1, \\ w_2, & \text{若 } c = c_2, \\ w_3, & \text{若 } c = c_3, \end{cases} \quad (13)$$

最終，將所有評論的嵌入向量進行加權平均，以生成餐廳級別的嵌入向量：

$$\mathbf{h}_i = \frac{\sum_{j=1}^{n_i} w(c_{i,j}) \cdot \mathbf{h}_{i,j}}{n_i}, \quad (14)$$

其中 \mathbf{e}_i 表示店家 i 的最終評論嵌入向量，涵蓋該店所有評論的綜合信息，以上方法有效結合主題分類與情感分析，通過嵌入向量表示提升了餐廳評論處理的精確性與針對性，最後再將該向量乘上店家的評分，得到最終的餐廳嵌入向量，如下式所示：

$$\mathbf{e}_i = \mathbf{h}_i \times S_i, \quad (15)$$

其中 S_i 代表店家 i 的評分， \mathbf{h}_i 為店家 i 的最終嵌入向量。

Table 2.
資料集的統計分析

Dataset Name	User	Item	Interaction	Density
Ratio 0.6	47	174	820	0.10026
Ratio 0.7	47	174	957	0.11702
Ratio 0.8	47	174	1115	0.13634
Ratio 0.9	47	174	1259	0.15394
Ratio 1.0	47	174	1389	0.16984

Table 3.
類別對應

Cuisine Category	Subcategories
中式/台式料理	台式, 滷味, 便當, 小吃, 粥, 炒飯, 餃子, 湯品
日韓料理	日式, 壽司, 丼飯/蓋飯, 拉麵, 韓式
東南亞料理	泰式, 越式, 東南亞, 咖哩
歐美料理	牛排, 披薩, 義大利麵, 三明治 / 吐司, 早餐, 歐美
港澳及異國料理	港式, 異國
健康餐與素食	健康餐, 素食
快餐/炸物	漢堡, 炸雞, 鹹酥雞/雞排
甜點與飲料	飲料, 甜點, 蛋糕, 甜甜圈, 豆花, 咖啡
火鍋及燒烤	火鍋, 燒烤, 鐵板燒
麵食類	麵食

3.2. Dataset

本研究整理了來自 Foodpanda 和 Google Map 的餐廳相關資料，分別蒐集了 77 間與 97 間餐廳的資訊。針對每間餐廳，設定了多項特徵，包括餐廳名稱、價位範疇及食物類別。其中，食物類別的分類主要參考 Foodpanda 平台的既有分類方式，最終共定義約 40 種食物類別，例如滷味、便當、牛排、披薩和壽司等。然而，由於 Google Map 獲取的餐廳資料缺乏食物類別分類，本研究以人工方式為該來源的餐廳標註相應的食物類別標籤。為簡化分析，研究將這 40 種細分類別整合為 20 類，並進一步合併為 10 類，其具體對應方式如表格 4 所示。本研究假設用戶喜歡某一食物類別時，會將該類別下的所有相關餐廳與用戶建立互動連結，進而形成用戶與餐廳之間的交互行為資料集。

基於用戶與餐廳的交互行為，進一步進行實驗，模擬不同密度的交互情況。本研究隨機從 47 位用戶與 174 間餐廳之間的交互資料中，挑選不同比例的連結

邊數，包括 0.6、0.7、0.8、0.9 和 1.0。這些比例代表了用戶與餐廳之間的交互密度，並對推薦系統的表現產生影響。如表格 2 所示，當邊數比例為 1.0 時，資料集的密度達到 0.16984，表示用戶與餐廳之間的交互行為非常密集。在此情況下，推薦系統需要處理大量的交互行為，以提供更精準的推薦結果。

3.3. Evaluation Metrics

本研究使用歸一化折扣累積增益（Normalized Discounted Cumulative Gain, NDCG）和召回率（Recall）作為評估推薦系統表現的指標。NDCG 考慮了推薦結果的排序，並對排序靠前的推薦結果給予更高的權重，以反映用戶對於靠前推薦的偏好。召回率則衡量了推薦系統在用戶有興趣的店家中能夠找出多少樣本，進一步評估了推薦系統的全面性。NDCG 和召回率的計算方式如下：

$$\text{NDCG@k} = \frac{1}{Z} \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \quad (16)$$

$$\text{Recall@k} = \frac{1}{Z} \sum_{i=1}^k \frac{rel_i}{\min(k, |R|)}, \quad (17)$$

其中 rel_i 表示第 i 個推薦結果的相關性， Z 為正規化因子， k 為推薦結果的數量， R 為所有正樣本的集合。本研究將 k 設定為 5，以保證推薦結果的多樣性，並進一步提高推薦系統的全面性。

3.4. Experimental Setup

本研究使用了以下參數設定來評估推薦系統的表現。實驗使用 Python 3.9.19 和 PyTorch 2.3.0 進行實現，CUDA 版本為 11.8。模型訓練過程中，使用了 Adam 優化器，學習率 (learning rate) 設定為 0.001，L2 正則化項設定為 0.0001。訓練過程中，總共進行了 120 個 epoch，每個批次的大小 (batch size) 為 16，測試批次的大小 (test batch size) 為 4。模型的 GCN 層數 (GCNLayer) 設定為 2，聚合方式 (aggregation) 為 "sum"，隱藏層維度 (dim) 為 16。為評估推薦系統的表現，本研究計算了前 5 個推薦結果的 Recall 和 NDCG 值 (topK 設定為 5)。

3.5. Expected Results

Table 4.
各資料集的 Recall 和 NDCG

Dataset Name	Recall@5	NDCG@5
Ratio 0.6	0.402	0.311
Ratio 0.7	0.469	0.399
Ratio 0.8	0.567	0.560
Ratio 0.9	0.647	0.664
Ratio 1.0	0.810	0.917

從實驗結果可觀察到挑選的邊數越多，推薦系統的表現也隨之提升。如表格 4 表所示，當挑選的邊數比例為 1.0 時，推薦系統的 Recall@5 和 NDCG@5 分別達到 0.810 和 0.917，這表示在所有正確的推薦項目中，有 81% 被系統成功推薦給了用戶，表明系統的全面性很高。同時，NDCG@5 的值為 0.917，表示推薦系統不僅能找到正確的項目，還能將它們按用戶的偏好排序，排序的品質非常高，反映了用戶對於推薦結果的滿意度很高。這些數字表明當挑選的邊數為 1.0 時，推薦系統在找到用戶感興趣的項目和排序這些項目方面都表現得非常出色。

4. Conclusion

我們提出了一種新穎的方法，利用 NIE-GCN 模型來解決餐廳推薦系統的挑戰。該模型能有效從二分異質圖中提取更深層的關聯信息，並精確預測顧客節點最感興趣的前 k 個餐廳節點。我們的方法在推薦系統中表現出色，Recall@5 達到 0.810，NDCG@5 則達到 0.917，顯示出該模型在推薦任務中的優越性能。

本研究的主要挑戰在於資料集的建立。由於無法透過爬蟲從 Foodpanda 和 Google Maps 等平台獲取用戶與餐廳的互動資料，我們採用了問卷調查的方式收集資料。但此方法導致資料量不足，且無法完全保證資料的真實性。問卷調查無法直接獲得用戶與餐廳的互動資料，因此我們只能藉由用戶填寫的興趣類別來模擬實際的互動關係。此外，如何將每個店家的評論與評分轉換為適合分析的特徵向量，也是我們面臨的一大挑戰。研究結果顯示，NIE-GCN 模型在提升餐廳推薦系統性能方面具有巨大潛力。未來的研究應著重於建立更大規模且更具真實性的資料集，例如與業界合作，進一步驗證並優化所提出的方法。

References

- [1] Y. Xiao, C. Li, and V. Liu, "Dfm-gcn: A multi-task learning recommendation based on a deep graph neural network," *Mathematics*, vol. 10, no. 5, 2022. 1
- [2] Y. Zhang, Y. Zhang, D. Yan, Q. He, and Y. Yang, "Nie-gcn: Neighbor item embedding-aware graph convolutional network for recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 5, pp. 2810–2821, 2024. 1, 2