

Steam Video Games Recommendation

第七組

M11202133 邱立楷

M11252004 謝奇容

M11252028 陳威



目錄

- 動機與目的
- 資料集介紹
- 方法介紹
- 實作結果
- 應用與未來展望

動機與目的

- Steam是全球最大的數位遊戲發行平台之一，擁有數以千計的遊戲和數以億計的玩家，我們也是玩家之一，因此在如此龐大的遊戲庫中，如何幫助玩家找到他們可能感興趣的遊戲，我們對此非常有興趣。
- 我們的目的是利用Link prediction方法，根據玩家的歷史行為，預測他們可能感興趣的遊戲，從而提高遊戲推薦的準確性。
- 除此之外，利用社群辨識的方法，幫玩家建立社群，使玩家可以與購買行為相似的玩家交流，互相介紹與推薦遊戲，提高遊戲的能見度。

資料集介紹

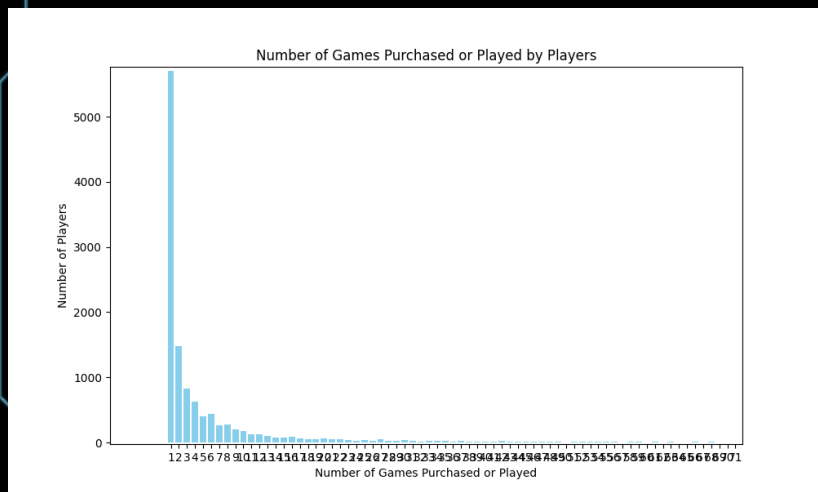
- 資料集採用來自Kaggle的Steam Video Games。
- Steam-200k.csv包含:
 - ◆ User ID
 - ◆ Name of the game
 - ◆ Behavior(purchase or play)
 - ◆ Play time(hours)

方法介紹

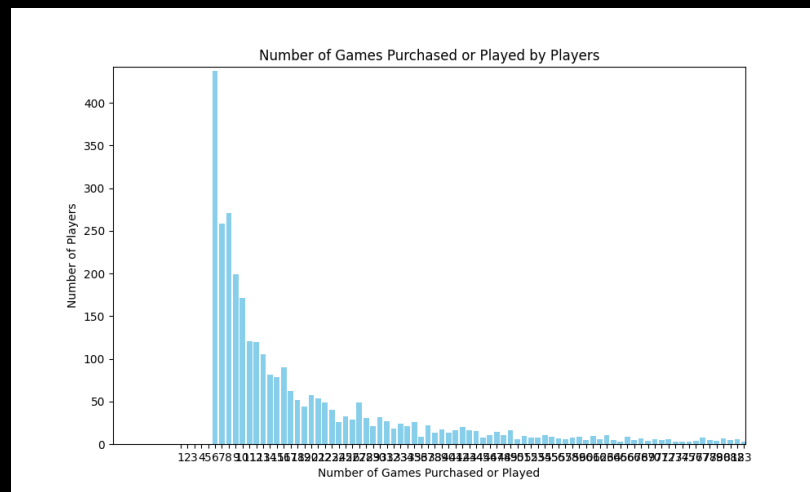
- 前處理
 - ◆ 資料刪減
 - ◆ 資料合併
 - ◆ 資料切割
 - ◆ 取得遊戲id
 - ◆ 特徵提取

方法介紹

- 資料刪減
 - ◆ 首先刪除資料集有一列全為0的無用欄位
 - ◆ 接著我們過濾掉遊戲數量小於(<)6的玩家資料
 - ◆ 資料數量為113829筆



圖(1)Raw data



圖(2)Filtered data

	Number of Games Purchased or Played,Number of Players
1	1,5700
2	2,1476
3	3,827
4	4,633
5	5,403
6	6,437

圖(3)遊戲數量小於6
的玩家人數

方法介紹

- 資料合併

- ◆ 我們將同個玩家的同個遊戲，原本分成兩列的資料，整合成同一列

```
user-id,game-title,behavior-name,value  
151603712,The Elder Scrolls V Skyrim,purchase,1  
151603712,The Elder Scrolls V Skyrim,play,273
```

圖(4)Raw data

```
user-id,game-title,behavior-name,value,play_time  
151603712,The Elder Scrolls V Skyrim,purchase,1.0,273.0
```

圖(5)Combined data

- 資料切割

- ◆ 使用sklearn.model_selection 的train_test_split做資料切割
- ◆ `train_user_ids, test_user_ids = train_test_split(user_ids, test_size=0.2, random_state=42)`

方法介紹

- 取得遊戲id
 - ◆ 資料集內只有提供遊戲名稱，在繪圖與分析時，以名稱作為節點較難處理，因此我們手動為各遊戲產生id
 - ◆ 先讀取資料集，並使用ndarray的unique()方法，取出不重複的每個遊戲，再依index為每個遊戲賦予一個id，最後將id寫回資料集內

```
user-id,game-title,behavior-name,value,play_time,game_id
76767,Banished,purchase,1.0,24.0,164
76767,Arma 2 Operation Arrowhead Beta (Obsolete),purchase,1.0,,294
76767,Arma 2 Operation Arrowhead,purchase,1.0,,162
```

圖(6) 加入game_id的data

方法介紹

- 繪圖
 - ◆ 我們讀取了train_data內的每筆資料，以play_time為權重，繪製user to game的有向權重圖
 - ◆ 我們還讀取了test_data內的每個user的一半資料，繪製於圖上，用以當作預測鏈結時的特徵

方法介紹

- 特徵提取
 - ◆ 我們根據作業1的經驗，使用了3個特徵作為鏈結預測的依據
 - ◆ 共同鄰居數量：計算user與game的共同鄰居數量，呈現兩者的潛在關聯性
 - ◆ user購買的遊戲數量：計算user的直接後繼節點的數量
 - ◆ game被購買的數量：計算game的直接前驅節點的數量

方法介紹

- 隨機森林
 - ◆ 我們使用隨機森林判斷是否有鏈結
 - ◆ 使用GridSearchCV來尋找最適合此資料的模型參數

```
param_dist = {"n_estimators" : [i * 5 for i in range(17, 26)],  
              "max_depth" : list(range(4,10)),  
              "min_samples_split" : [i * 5 for i in range(18, 30)],  
              "min_samples_leaf" : [i * 5 for i in range(1, 5)]}  
  
clf = RandomForestClassifier(random_state=25, n_jobs=-1)  
rf_random = GridSearchCV(estimator=clf, param_grid=param_dist, cv=5, n_jobs=-1, verbose=2)  
  
rf_random.fit(x_train, y_train)
```

圖(7) 隨機森林訓練參數

方法介紹

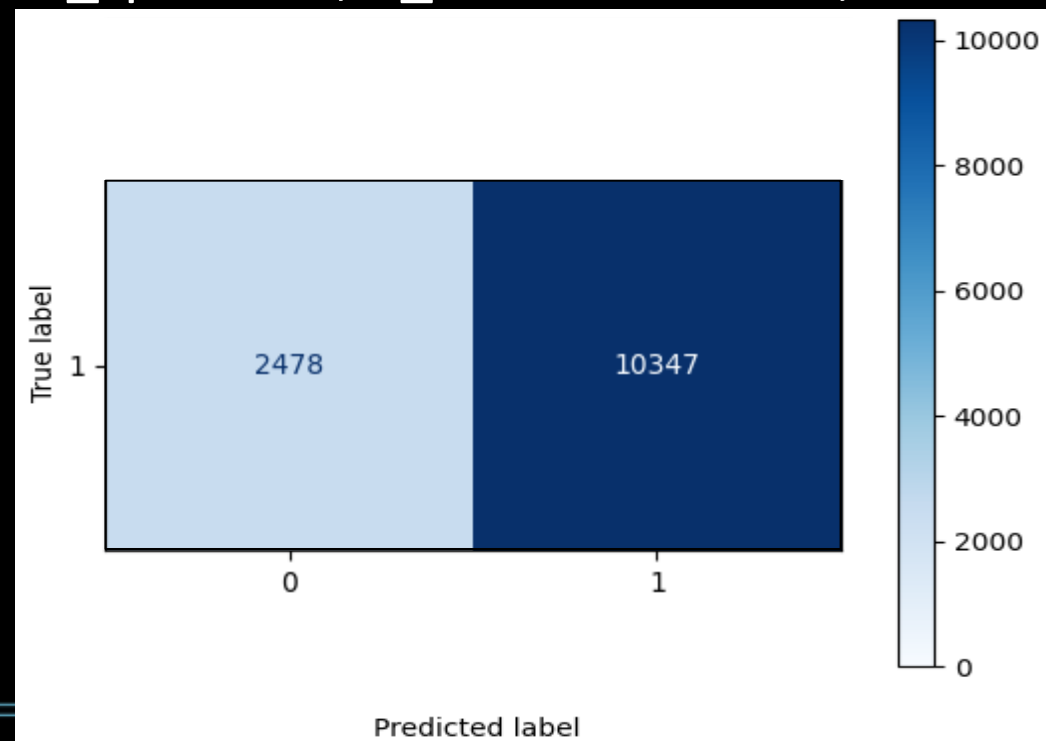
- 遊戲推薦系統
 - ◆ 我們使用訓練出來的模型，預測測試資料集的所有使用者，在未有鏈結的節點上，鏈結的可能性
 - ◆ 並根據該使用者的測試遊戲筆數，提取可能性最高的兩倍數量的遊戲，作為我們的推薦結果

方法介紹

- 使用者分群
 - ◆ 我們根據作業2的經驗，使用networkx內的greedy_modularity_communities為使用者分群

實作結果

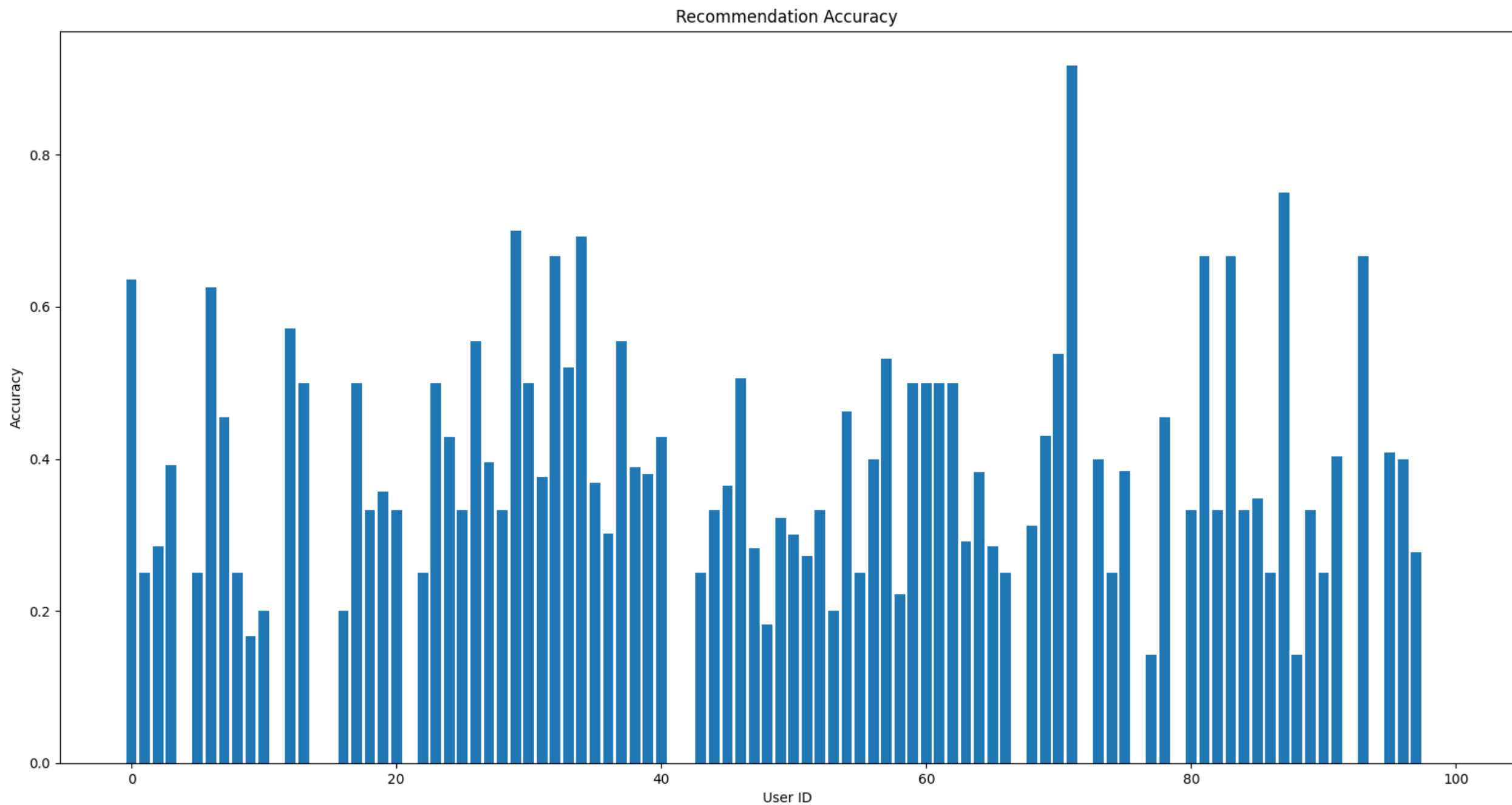
- 隨機森林結果
 - ◆ best estimator: RandomForestClassifier(max_depth=6, min_samples_leaf=20, min_samples_split=120, n_estimators=120, n_jobs=-1, random_state=25)
 - ◆ score: 0.81
 - ◆ f1 score: 0.89



圖(8) 測試資料的混淆矩陣

實作結果

- 遊戲推薦系統
 - ◆ 最後為100個使用者推薦了遊戲
 - ◆ 準確率的評估標準是已購買遊戲中，有在推薦清單的比例
 - ◆ 平均有34%購買的遊戲，有出現在推薦清單中



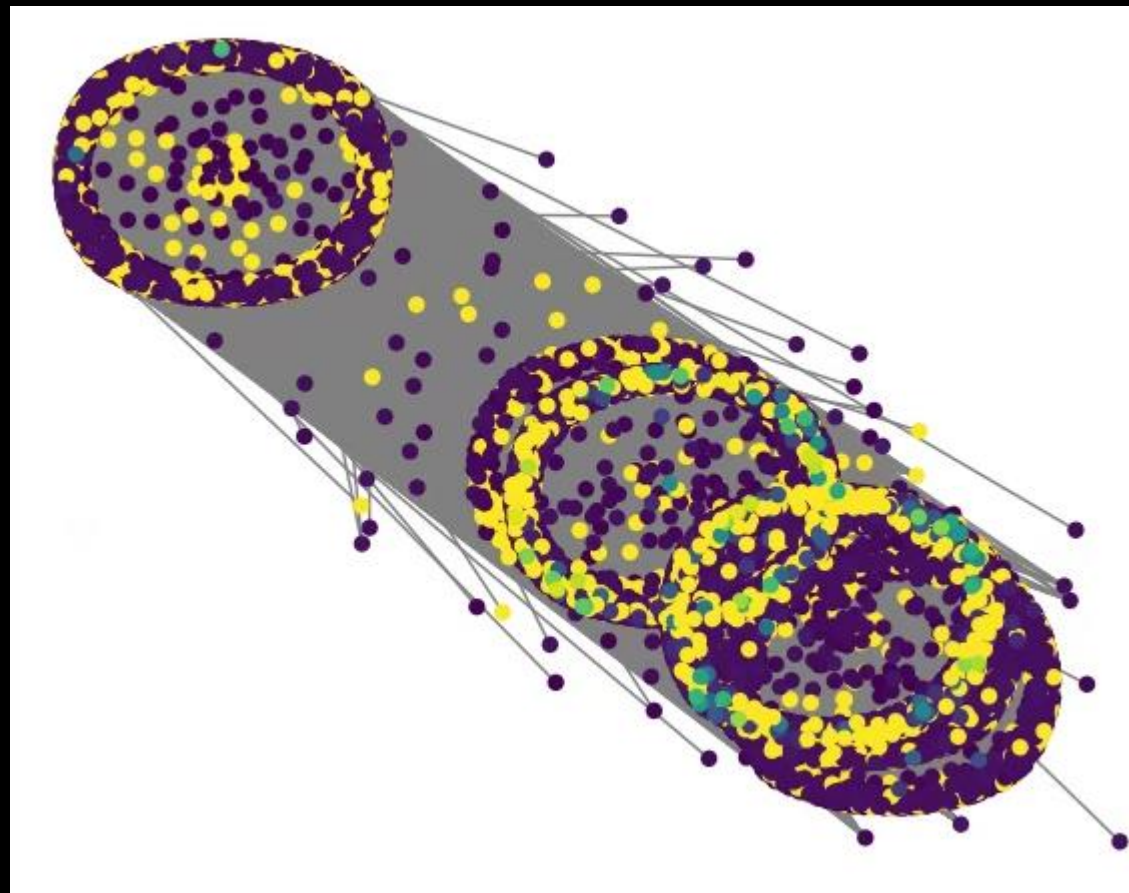
圖(9) 推薦系統對100名使用者的準確率

實作結果

- 使用者分群
 - ◆ 所有使用者分成82個社群

```
community count: 82  
Total nodes: 17548  
Total edges: 128804
```

圖(10) 社群分析結果



圖(11) 社群分析圖

應用與未來展望

- 我們的模型在鏈結預測上有高達89%的準確率，雖推薦系統準確率不佳，但我們相信只要能增加更多的特徵，一定能讓準確率提升。且在部分使用者身上，我們也有看到高準確率的推薦，所以我們認為此方法不只可以用在Steam的遊戲推薦，也能用在相似的平台，例如Epic、Nintendo eshop、PlayStation Store等各大遊戲平台，甚至能使用於影音平台的推薦系統

應用與未來展望

- 我們還依據使用者購買的遊戲，提供了使用者的分群，若能在遊戲購買平台上提供交流區，供各社群的成員互相分享、推薦遊戲，能讓使用者更容易找到自己有興趣的遊戲
- 未來，若能為購買遊戲的鏈結增加更多特徵，將能使鏈結預測更準確，也較能推薦更符合使用者期望的遊戲，增加使用者的購買遊戲的數量，也提高使用者對平台的期望值，讓使用者能購買、遊玩到真的符合自己興趣的遊戲，也讓遊戲發行商能增加銷量，創造雙贏的局面

分工

學號與姓名	工作內容
M11202133 邱立楷	<ul style="list-style-type: none">• 資料前處理• PPT製作
M11252004 謝奇容	<ul style="list-style-type: none">• 隨機森林、遊戲推薦系統• 使用者分群• PPT製作
M11252028 陳威	<ul style="list-style-type: none">• 報告錄影