

# 推薦系統 API 雛形 — 技術簡介 (Ver. 2.0)

作者： 陳韋辛 Frank Chen

題目： 「以類神經網路演算法和回歸分析實作個性化推薦系統之 API 雛形」

## 對鴻揚之可能貢獻度與實用度：

商品評分與推薦系統日趨普遍，本人也對具有實用性的演算法相當感興趣，因此決定研讀並實作一個即便不懂該演算法也容易使用的推薦系統 API，期望能協助拓展公司未來接案的種類，如購物網站或是旅遊網站等具有推薦功能之專案。此套件不僅能使用在評價推薦系統，只要適當調整輸入格式，也能用來預測使用者在網站中的未來行為與性質，例如未來申請商品退件的可能性以及此顧客的信用等級。

## 功能：

預測所有顧客對未知電影或產品之評價，以便未來對該顧客進行商品推薦，此外，此套件可以動態新增、更新顧客之評價或新增新產品，而不會干擾到原先已處理過的數據導致重新計算，相當符合實際使用情形。

## 程式碼：

- Github: <https://github.com/WeiHsinChen/RecommendationAPI>
- Google Drive: <http://ppt.cc/6iumD>

## 使用環境：

- 以 Python 語言實作
- 使用 SQLite 資料庫
- 須下載 SQLAlchemy 便於資料庫存取以及 numpy 套件以便矩陣運算

## 套件優點：

使用方便簡單，足以滿足最基本推薦系統的需求。輕量化設計，相較於其他開放源程式碼的推薦系統套件(e.g., RecommendationAPI in Drupal)更加簡單易懂，也不用像 Prediction API in Google Cloud Platform 或是 Machine Learning Recommendation API in Microsoft Azure 等需要高額使用費用。此套件也支援動態更新資料，且越新的評價在分析中會佔越高之權重，越舊的評價會佔分析中較低之權重。此外，此套件雛型能夠輕易轉型成其他推薦形式，如將商品評分紀錄改成商品點擊紀錄等，便能計算顧客對該商品的未來點擊數以預測其喜好程度。

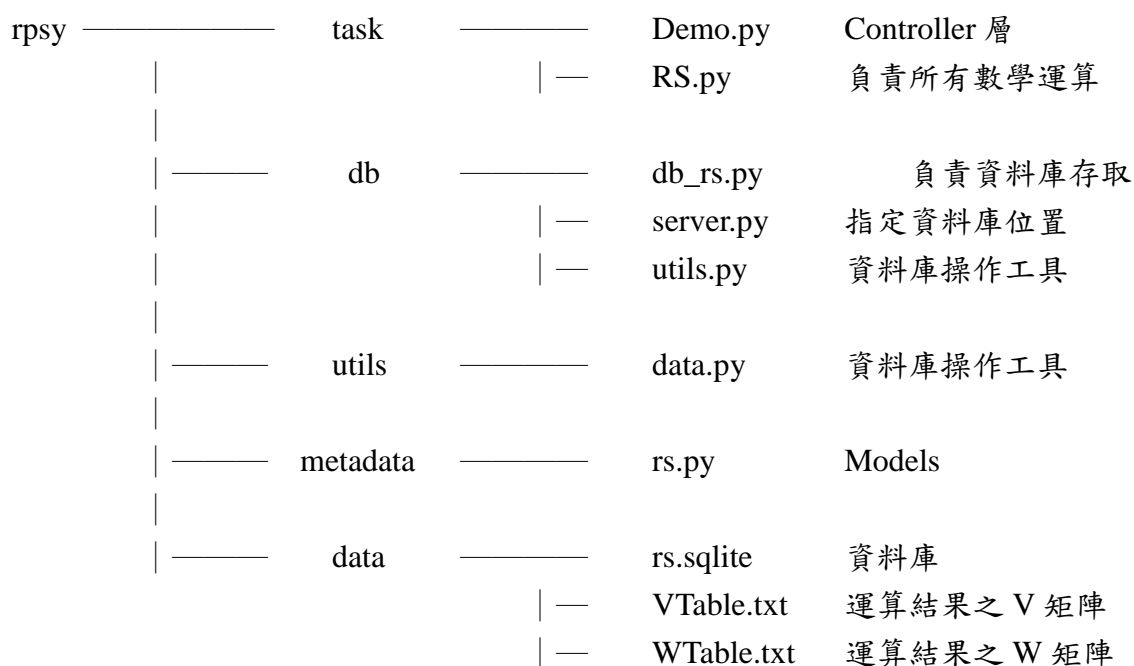
## 待改善處：

目前的輸入參數之格式完全是便於現場示範，且以 Command-Line Interface 做為展示平台，但未來只須依照使用者的需求做調整即可。此外，推薦準度以及運算速度有待最佳化，並且還有許多可以擴充之功能，如數據過大時使用 Hadoop 等 Map reduce 技巧降低計算之需求或是 denoising 提升準確性。

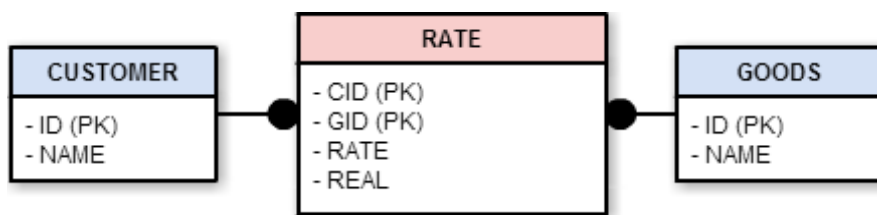
### 近期預計之變更：

1. 與 Twisted 套件結合，或使用相同的架構不同的語言實做，並做成 Web-API 供前台網站做使用。
2. 若本身資料夠多，即可省略步驟一. Save raw data 與步驟二. Train function，不必一次分析大量資料。一開始可以直接逐筆加入商品與顧客之評價，因為當加入新顧客時系統便會再次局部性計算並分析資料。
3. 增加 Cross Validation 功能以便使用者能選取適當的神經元數量與迴圈數。
4. 目前並沒有建立 delete 功能的原因是就算是過時的顧客或是商品，也對分析有一定的貢獻度，因此除非資料實在過於龐大，不然沒有必要進行刪除作業，不過之後也會依照使用需求加上該功能。

### 程式架構圖：



### 資料庫關聯圖：



✧ CUSTOMER	— ID	(int)	顧客 ID
	— NAME	(int)	顧客姓名
✧ GOODS	— ID	(int)	商品 ID
	— NAME	(int)	商品名稱
✧ RATE	— CID	(int)	顧客 ID
	— GID	(int)	商品 ID
	— RATE	(float)	評分
	— REAL	(bit)	顧客評分(1)或預測評分(0)

## 推薦系統 API 雛形 — 使用說明 (Ver. 1.2)

功能描述	初次存取評價資料	業務別	資料存取
函式名稱	<a href="#">/Demo/save_raw_data</a>	傳輸介面	
使用情境	初次使用 API 時，僅需執行一次。		
備註	<p>第一次使用時，必須將全體用戶之評分紀錄文字檔(.txt)匯入資料庫，以便存取與分析，因此使用者須將評分文字檔之「相對路徑」為輸入，呼叫 Demo.py 中的 save_raw_data 方程式，文字檔格式參考 Google 問卷調查之輸出格式，如 data/raw_data.txt。</p>  <pre> parallels@ubuntu: /media/psf/Home/Documents/RecommendationAPI/rspy/task parallels@ubuntu:~\$ cd '/media/psf/Home/Documents/RecommendationAPI/rspy/task' parallels@ubuntu:/media/psf/Home/Documents/RecommendationAPI/rspy/task\$ python D emo.py save_raw_data ../data/raw.txt  Save raw data successfully parallels@ubuntu:/media/psf/Home/Documents/RecommendationAPI/rspy/task\$ </pre>		

### 輸入資料說明

層級	屬性	欄位名稱	說 明	欄位型態	備註																
0	Attribute	data	評價資料	char																	
範例	<ul style="list-style-type: none"><li>● Command: cd “套件內 task 資料夾位置” python Demo.py save_raw_data ../data/raw.txt</li><li>● Content-Type: "application/text"</li><li>● Data:<table><tr><td>"Timestamp</td><td>Name</td><td>復仇者聯盟</td><td>鳥人</td></tr><tr><td>2015/05/05</td><td>Frank</td><td>5</td><td></td></tr><tr><td>2015/05/05</td><td>Andy</td><td>5</td><td>5</td></tr><tr><td>2015/05/07</td><td>Allen</td><td>5</td><td>5"</td></tr></table></li><li>● 上述 Data 每行中的每筆資料相隔一個 Tab，若某顧客對該筆商品沒有評價，則不用輸入其數字，如同上例 Frank 只評價電影”鳥人”。</li></ul>					"Timestamp	Name	復仇者聯盟	鳥人	2015/05/05	Frank	5		2015/05/05	Andy	5	5	2015/05/07	Allen	5	5"
"Timestamp	Name	復仇者聯盟	鳥人																		
2015/05/05	Frank	5																			
2015/05/05	Andy	5	5																		
2015/05/07	Allen	5	5"																		

### 輸出資料說明

層級	屬性	欄位名稱	說明	欄位型態	備註
範例					

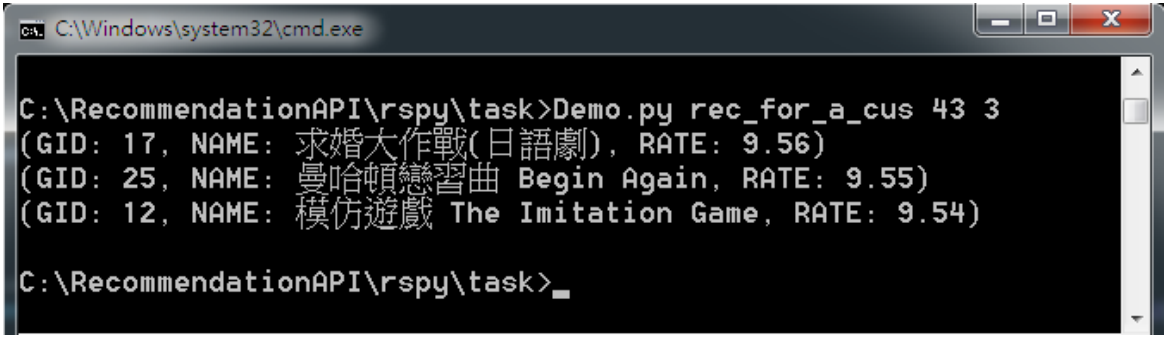
功能描述	初次計算測試資料	業務別	資料分析
函式名稱	<a href="#">/Demo/train</a>	傳輸介面	
使用情境	儲存完資料後，初次計算測試資料，僅需執行一次。		
備 註	<p>第一次使用時，必須計算所有用戶之評分資料，以產生矩陣 V 和矩陣 W 以便預測全體用戶之行為，因此使用者必須呼叫 Demo.py 中的 train 方程式，並輸入參數「神經元數量」以及「計算迴圈數」，若無輸入參數，則預設神經元數量為 10，計算迴圈數為 10000000。神經元數量與計算迴圈數之物理意義請詳閱「必要之參數物理意義簡介」。</p>  <pre> parallels@ubuntu: /media/psf/Home/Documents/RecommendationAPI/rspy/task parallels@ubuntu:~\$ cd '/media/psf/Home/Documents/RecommendationAPI/rspy/task' parallels@ubuntu:/media/psf/Home/Documents/RecommendationAPI/rspy/task\$ python Demo.py train 8 50000 Step: 0 Step: 10000 Step: 20000 Step: 30000 Step: 40000 Time taken: 3.05 seconds. Ein: 0.00863  Train data successfully </pre>		

#### 輸 入 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
範 例		<ul style="list-style-type: none"> <li>Command: cd “套件內 task 資料夾位置” python Demo.py train 8 50000</li> <li>上述命令中每組字串以空格分隔，第一個數字代表「神經元數量」、第二個數字代表「計算迴圈數」</li> </ul>			

#### 輸 出 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
0	Attribute	Ein	測資誤差	char	
String 範 例		<ul style="list-style-type: none"> <li>Data: “0.02”</li> </ul>			

功能描述	商品推薦	業務別	商品推薦
函式名稱	<a href="#">/Demo/rec_for_a_cus</a>	傳輸介面	
使用情境	推薦顧客未知商品。		
備 註	<p>推薦顧客未評價之商品時，使用者須要呼叫 Demo.py 中的 rec_for_a_cus 方程式，並輸入參數「顧客 ID」以及「欲推薦數量」，若無輸入「欲推薦數量」，則欲設為 3，返回值為該數量之推薦商品資料。</p> 		

#### 輸 入 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
範 例					<ul style="list-style-type: none"> <li>Command: cd “套件內 task 資料夾位置” python Demo.py rec_for_a_cus 43 3</li> <li>上述命令中每組字串以空格分隔，第一個數字代表「顧客 ID」、第二個數字代表「預推薦數量」</li> </ul>

#### 輸 出 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
1	Attribute	GID	商品 ID	char	
1	Attribute	NAME	商品名稱	int	
1	Attribute	RATE	預測評分	int	
String 範 例					<ul style="list-style-type: none"> <li>Content-Type: "application/json"</li> <li>由於是在 Terminal 上顯示，在不同系統設定下會有編碼不同之問題。</li> <li>Data: <ul style="list-style-type: none"> <li>[{"ID": 17, "NAME": "求婚大作戰(日語劇)", "RATE": 9.56},</li> <li>{"ID": 25, "NAME": "曼哈頓戀習曲 Began Again", "RATE": 9.55},</li> <li>{"ID": 12, "NAME": "模仿遊戲", "RATE": 9.54}]</li> </ul> </li> </ul>

功能描述	新增商品			業務別	資料存取
函式名稱	/Demo/add_goods			傳輸介面	
使用情境	新商品上架時，將此新商品加入預測系統。				
備註	<div>新增商品時，須呼叫 Demo.py 中的 add_goods 方程式，並將商品名稱之文字檔(.txt)作為輸入參數即可。</div> <div></div>				
輸 入 資 料 說 明					
層 級	屬性	欄位名稱	說 明	欄位型態	備註
範 例	<div>● Command: cd “套件內 task 資料夾位置” python Demo.py add_goods ../data/new_goods.txt</div> <div>● Content-Type: "application/json"</div> <div>● Data: [“鴻揚超人首部曲”, “鴻揚超人二部曲”]</div>				
輸 出 資 料 說 明					
層 級	屬性	欄位名稱	說 明	欄位型態	備註
1	Attribute	ID_List	新商品之商品 ID	Char	
String 範 例	<div>● Data: [“29”, “30”]</div>				



功能描述	新增顧客	業務別	資料存取
函式名稱	<a href="#">/Demo/add_a_cus</a>	傳輸介面	
使用情境	新顧客(會員)加入時，將此新顧客加入預測系統。		
備 註	<p>新增顧客時，須呼叫 Demo.py 中的 add_a_cus 方程式，並將該用戶之評分文字檔 (.txt)作為輸入參數即可。</p>  <pre> parallels@ubuntu: /media/psf/Home/Documents/RecommendationAPI/rsfy/task parallels@ubuntu:~\$ cd '/media/psf/Home/Documents/RecommendationAPI/rsfy/task' parallels@ubuntu:/media/psf/Home/Documents/RecommendationAPI/rsfy/task\$ python D emo.py add_a_cus ../data/new_cus.txt  Add a new customer successfully </pre>		

#### 輸 入 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
1	Attribute	Name	顧客姓名	char	
2	Attribute	GID	商品 ID	int	
2	Attribute	RATE	評分	int	

範 例	<ul style="list-style-type: none"> <li>Command: cd “套件內 task 資料夾位置” python Demo.py add_a_cus ../data/new_cus.txt</li> <li>Content-Type: "application/json"</li> <li>Data: <ul style="list-style-type: none"> <li>{"NAME": "新顧客之姓名",</li> <li>"DATA": [{"GID": 1, "RATE": 5},</li> <li>          {"GID": 3, "RATE": 6},</li> <li>          {"GID": 14, "RATE": 1},</li> <li>          {"GID": 16, "RATE": 3},</li> <li>          {"GID": 21, "RATE": 9}]}</li> </ul> </li> </ul>				
-----	---	--	--	--	--

#### 輸 出 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
1	Attribute	ID	新顧客之 ID	Char	
String 範 例	<ul style="list-style-type: none"> <li>Data: “44”</li> </ul>				

功能描述	更新顧客資料	業務別	資料存取
函式名稱	<a href="#">/Demo/update_a_cus</a>	傳輸介面	
使用情境	顧客更改評價或會員名稱時。		
備 註	<p>更新顧客評分時，須呼叫 Demo.py 中的 update_a_cus 方程式，並將該用戶之評分文字檔(.txt)作為輸入參數即可。</p>  <pre> parallels@ubuntu: /media/psf/Home/Documents/RecommendationAPI/rs.py/task parallels@ubuntu:~\$ cd '/media/psf/Home/Documents/RecommendationAPI/rs.py/task' parallels@ubuntu:/media/psf/Home/Documents/RecommendationAPI/rs.py/task\$ python Demo.py update_a_cus ../data/update_cus.txt  Update a new customer successfully </pre>		

#### 輸 入 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
1	Attribute	ID	顧客 ID	int	
1	Attribute	Name	顧客姓名	char	
2	Attribute	GID	商品 ID	int	
2	Attribute	RATE	評分	int	

範 例	<ul style="list-style-type: none"> <li>● Command: <code>cd “套件內 task 資料夾位置”</code> <code>python Demo.py update_a_cus ../data/update_cus.txt</code></li> <li>● Content-Type: "application/json"</li> <li>● Data: <ul style="list-style-type: none"> <li>{“ID”: 43,</li> <li>“NAME”: “更新顧客姓名”,</li> <li>“DATA”: [{"GID": 25, "RATE": 1},</li> <li>          {"GID": 26, "RATE": 1},</li> <li>          {"GID": 27, "RATE": 1}]}</li> </ul> </li> </ul>				
-----	--	--	--	--	--

#### 輸 出 資 料 說 明

層 級	屬性	欄位名稱	說 明	欄位型態	備註
1	Attribute	ID	更新顧客之 ID	Char	
String 範 例	<ul style="list-style-type: none"> <li>● Data: “43”</li> </ul>				