

免費玩家禮包推薦



目的

- 提供免費玩家高折扣禮包，提升付費率
- 進而延續其留存及付費可能性

優點

- 免費玩家本就不付費，高折扣禮包不會造成收益下降
- 易驗證，只要成功破冰都算是成效

缺點

- 僅針對破冰目的，無法提升付費玩家收益

| 大綱

1

推薦系統概述

2

本專案採用模式

3

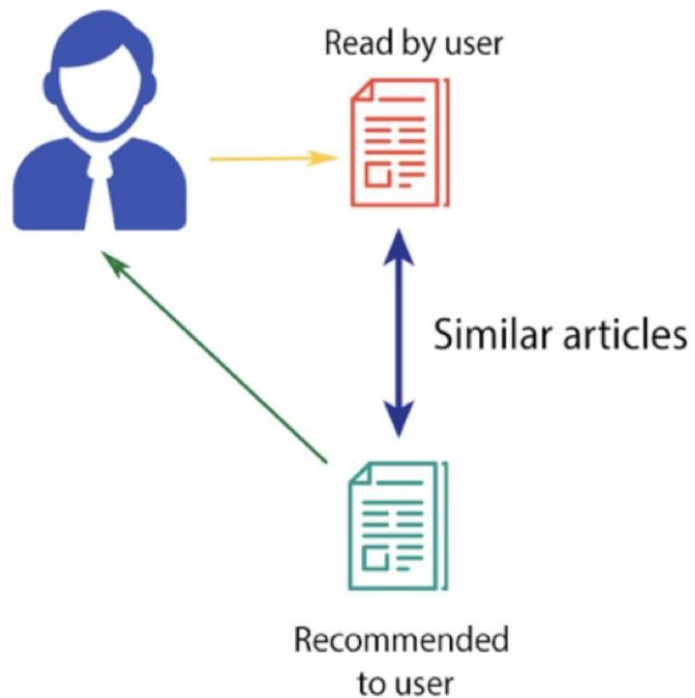
執行進度及實踐程序

1. 推薦系統概述

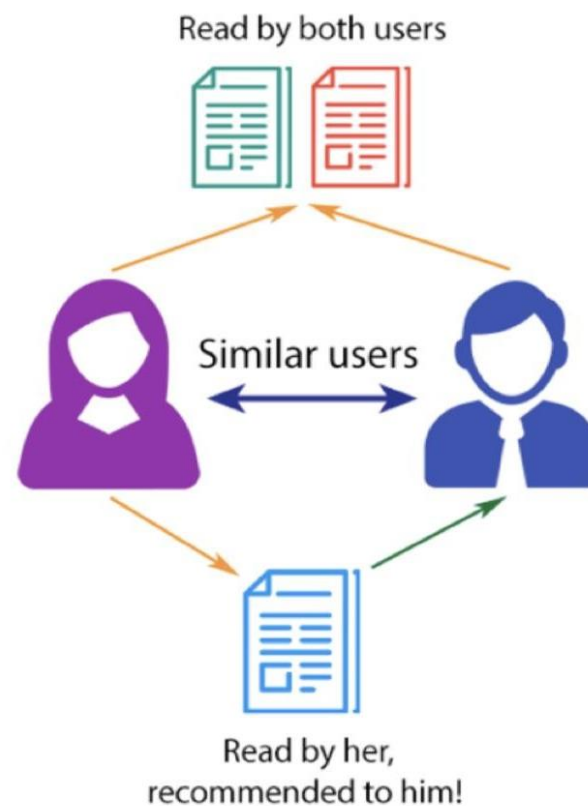


基本概念

CONTENT-BASED FILTERING



COLLABORATIVE FILTERING



- 假設用戶對於某些項目的興趣是由其特徵決定的，如電影的類型、演員、導演、年份等

- 假設用戶的興趣和其他用戶相似的項目也可能感興趣，需要用戶的行為數據，如評分、購買行為

收集評分

用戶

商品

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

顯式回饋 (explicit feedback)

- 實質矩陣：量化容易，如打分、評價
- 二元矩陣：如喜歡、不喜歡

隱式回饋 (implicit feedback)

- 一類矩陣：只記錄積極的反應，如購買次數、觀看時長。空值可能是用戶不喜歡或不瞭解該項目，比較貼近現實世界

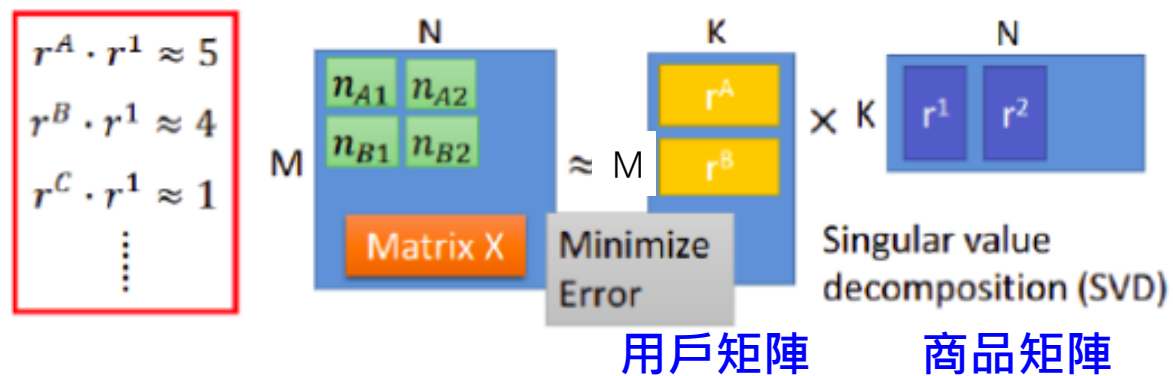
矩陣分解法

- ALS Matrix Factorization：計算快、泛化能力弱、不能處理非線性問題
- Logistic Matrix Factorization：能處理非線性、計算成本高
- Bayesian Personalized Ranking：對稀疏數據適應能力差

矩陣分解



No. of Otaku = M No. of characters = N No. of latent factor = K



$$L = \sum_{(i,j)} (r^i \cdot r^j - n_{ij})^2$$

- 什麼是 latent factor K ?

舉例

Assume the dimensions of r are all 2 (there are two factors)

	A	0.2	2.1	1 (金星)	2 (銀星)	3 (通用技能書)	4 (高級技能書)
B	B	0.2	1.8	0.0	0.1	1.9	2.2
C	C	1.3	0.7	2.2	1.5	-0.3	0.5
D	D	1.9	0.2				
E	E	2.2	0.0				

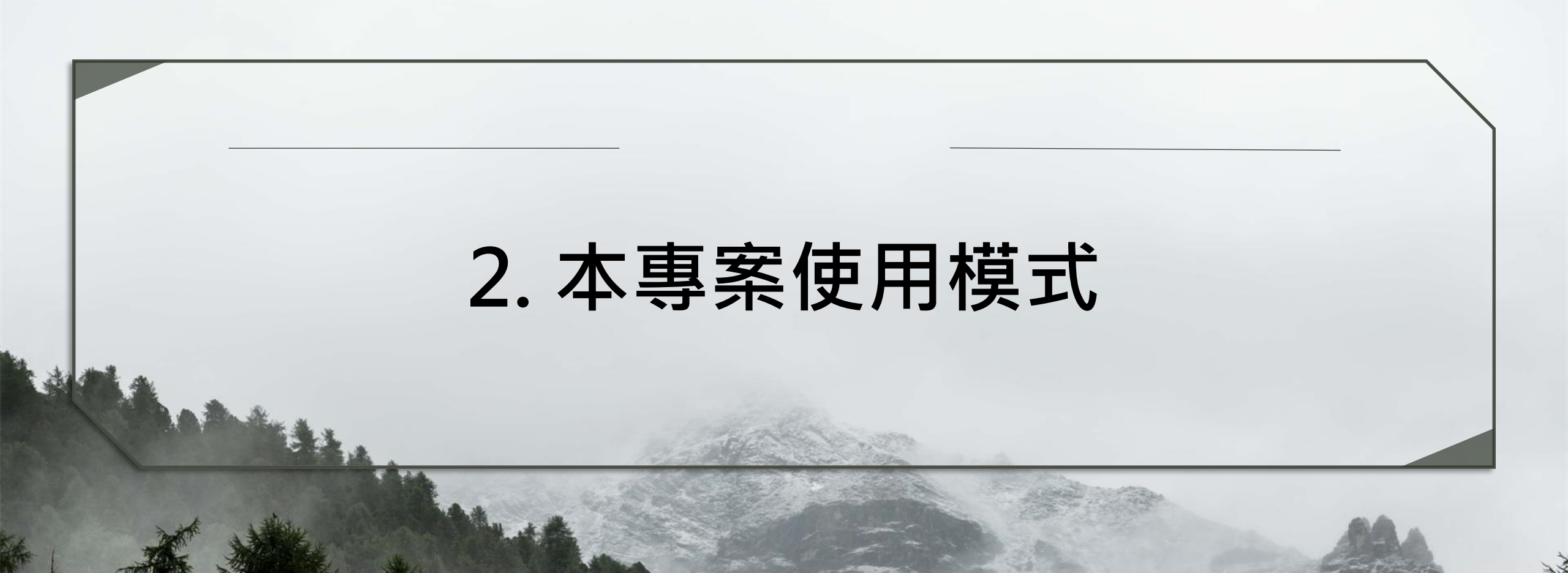
喜歡培養 喜歡球員

玩家A多喜歡金星

$$C_{(A,1)} = 0.2 \times 0.0 + 2.1 \times 2.2 = 4.6$$

玩家A多喜歡通用技能書

$$C_{(A,3)} = 0.2 \times 1.9 + 2.1 \times (-0.3) = -0.25$$



2. 本專案使用模式



Logistic Matrix Factorization

- 使用sigmoid函數，將矩陣U和V的內積轉化為[0,1]範圍內的概率值
- 正則化：LMF中使用L2正則化項對矩陣U和V的大小進行控制，以避免過擬合問題

$$L(\mathbf{P}, \mathbf{Q}) = \sum_{(u,i) \in D} (r_{u,i} - \sigma(\mathbf{p}_u \cdot \mathbf{q}_i))^2 + \lambda(\|\mathbf{P}\|^2 + \|\mathbf{Q}\|^2)$$

L2

- 採用梯度下降法求解，沿目標函數下降最快方向進行迭代計算 (偏微分)

```
from implicit.cpu.lmf import LogisticMatrixFactorization
...
# f 代表latent factors，考量多少個隱含因素，即k值，定義分解矩陣的特徵數量。
# 通常越大的特徵數量會有較好的預測能力，但也可能導致過擬合。默認值為 64
# learning rate：梯度下降法求解時的步伐，訓練時的學習率，控制參數更新的速度。默認值為 0.01
# reg：正則化參數，用於控制模型的複雜度。較大的值會使得模型較為簡單，較小的值會使得模型較為複雜。默認值為 0.01
# iterations，迭代次數，通常需要多次迭代才能收斂到最優解。默認值為 15
...
```

1000000

- 綁鑽商品被購買次數至少超過100次
- 時裝品項過多，故將類似時裝歸為同一類。如：時裝－真實、時裝－可愛
- 從開服至今所有購買綁鑽紀錄

[illegible]

訓練及測試集

- 訓練：測試 = 8：2
- 以未遮擋的8成資料建立訓練集，遮擋的2成資料作為測試

ID	商品A	B	C	D	E	F	G	H	I	J
1	5									
2	3					8		1		1
3		5								
4									6	
5			6					99		12
6	10					4			5	
7		85		1		20				50

預測結果

- 抓取近 1 天有上線，且有綁鑽消費紀錄的用戶資料
- 模型將上述【用戶矩陣】dot【商品矩陣】，推前3項
如 喜好排序 [0.9, 0.8, 0.5, 0.3, 0.2, 0.2]，但紅色已買過，故推 1, 3, 4項

							Metrics
↓ Start Time	Duration	Run Name	User	Source	Version	Models	precision
✓ 12 hours ago	4.6s	-	root	train.py	-	jjna_free_.../12	0.378
✓ 1 day ago	3.3s	-	root	train.py	-	jjna_free_.../11	0.372
✓ 1 day ago	3.1s	-	root	train.py	-	jjna_free_.../10	0.379
✓ 2 days ago	4.6s	-	root	train.py	-	jjna_free_.../9	0.377
✓ 3 days ago	5.2s	-	root	train.py	-	jjna_free_.../8	0.361

目前結果，平均推3項會有1項成功命中

3.執行進度及實踐程序



執行進度

- 模型已於MLflow測試機部署
- 排程控制定期更新訓練集，每日預測1次
- 有控制id清單功能，預測後且上線玩家，7天內不會再推薦
- 清單格式：

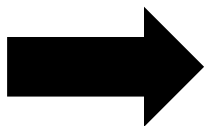
```
-d '{
  "gamecode": "xxx",
  "time": 1640966400,
  "mode": "FreePlayerGiftRecommend",
  "count": 4,
  "hash": "d4d042b79499718112faf5a3c69d33a5",
  "list": [
    {
      "rid": "aaaaa",
      "extList": ["real_gift_id1", "real_gift_id2", "real_gift_id3"]
    },
    {
      "rid": "bbbbb",
      "extList": ["real_gift_id1", "real_gift_id3", "real_gift_id5"]
    }
  ]
}
```

#account_id	領取獎勵日期
0	2023-02-20
1	2023-02-20
30	2023-02-15
34	2023-02-15
43	2023-02-19
...	...
6518	2023-02-20
6519	2023-02-20
6520	2023-02-20
6526	2023-02-20
6527	2023-02-21

實踐程序-1

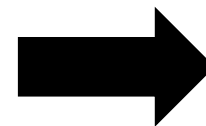
- 預測結果 (綁鑽商品)

- 1.周期限购商店购买_中级突破卡
- 2.购买服装_黑曜-上身
- 3.神秘商店购买_高级突破卡



- 對應類別

- 1.培養-球員
- 2.時裝-Kuso
- 3.培養-球員



- 推薦禮包

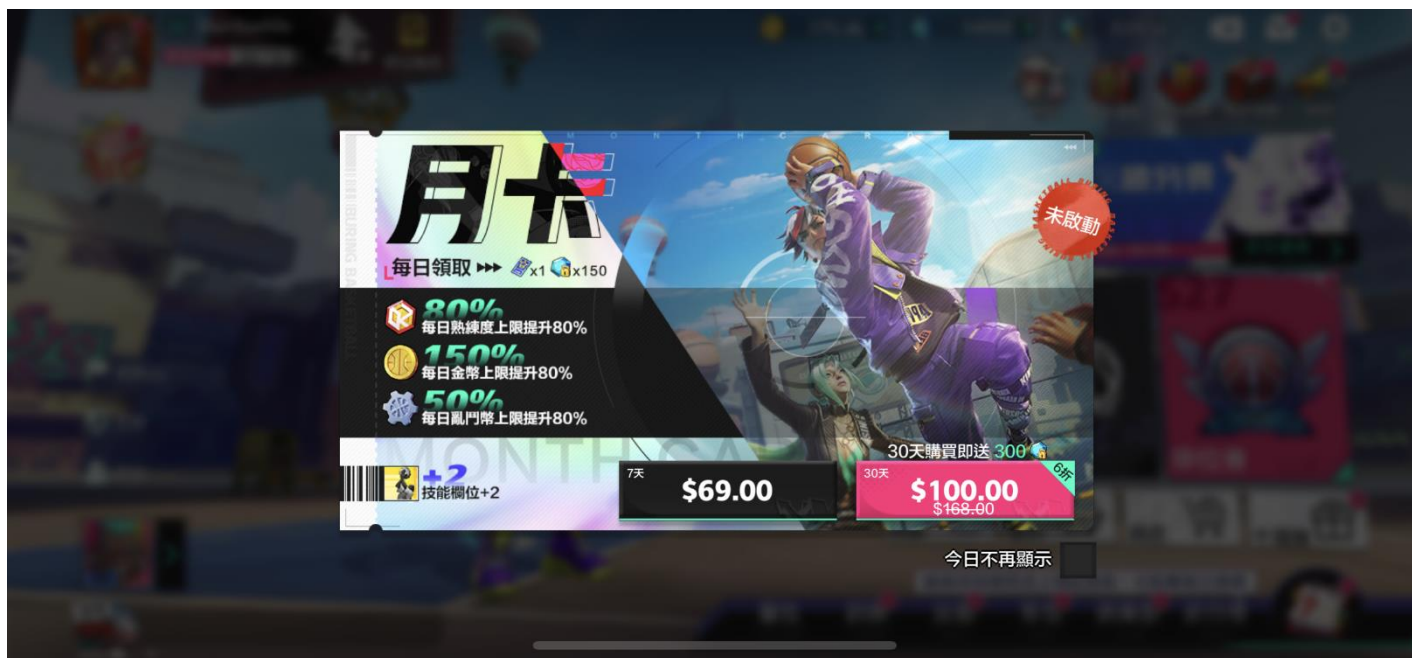
- 1.培養-球員 \$0.99禮包
- 2.培養-球員 \$2.99禮包
- 3.時裝-Kuso \$0.99禮包

商品種類
花式
時裝-Kuso
時裝-可愛
時裝-真實
時裝-現代
時裝-運動
培養-卡牌
培養-技能
培養-球員
培養-貼紙
球員
資源-大亂鬥
資源-金幣
資源-時裝
慶祝

- 需設計左表\$0.99、\$2.99、\$4.99直購禮包
- 15種共45項

實踐程序-2

- 每日API提供預測資料
- 當玩家上線時，以拍臉圖彈出限時禮包
- 關閉後仍需有紅點顯示並倒數計時
- 建議設計無法用綁鑽購買商品



簡報結束 補充資料



Logistic Matrix Factorization

1. 會先計算所謂的confidence level，並且將R(紀錄某用戶對某商品的喜愛程度) 轉成C

$$C_{ui} = \alpha \cdot r_{ui}$$

2. 用戶u 購買商品i的機率 $p(l_{ui} | x_u, y_i, \beta_u, \beta_i) = \frac{\exp(x_u y_i^T + \beta_u + \beta_i)}{1 + \exp(x_u y_i^T + \beta_u + \beta_i)}$

3. 看到矩陣C的機率 $\mathcal{L}(R | X, Y, \beta) = \prod_{u,i} p(l_{ui} | x_u, y_i, \beta_u, \beta_i)^{\alpha r_{ui}} (1 - p(l_{ui} | x_u, y_i, \beta_u, \beta_i))$

4. 找出X,Y,B 使的 $\mathcal{L}(R | X, Y, B)$ 最大

ALS Matrix Factorization

1. 會先計算所謂的confidence level

$$c_{ui} = 1 + \alpha \cdot r_{ui}$$

2. 新變量 p_{ij} : 用戶 u 對商品 v 的偏好

$$p_{ij} = \begin{cases} 1, & r_{ij} > 0 \\ 0, & r_{ij} = 0 \end{cases}$$

3. 目標函式 $\min_{u,v} \sum_{i,j} c_{ij} (p_{ij} - u_i^T v_j)^2 + \lambda (\sum_i |u_i|^2 + \sum_j |v_j|^2)$

KNN - 1

1. 需要先衡量商品相似程度
2. 主要用cosine similarity

	用戶A	用戶B	用戶C	用戶D	用戶E	用戶F
商品A	1	1	1	0	0	0
商品B	0	0	0	1	1	1

$$\begin{aligned} a \cdot b &= \\ 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 1 + 0 \cdot 1 &= 0 \end{aligned}$$



商品A跟商品B的cos sim = 0

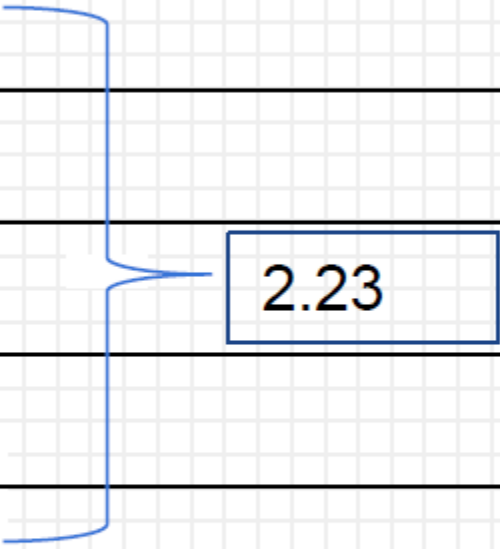
$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

KNN - 2

用戶A 買過的 商品	對商品的評 分	跟某個未買過商品的相似度
a1	5	0.75
a2	8	0.78
a3	3	0.4
a4	2	0.2
a5	4	0.1



對該未曾買
過的商品的
喜歡程度

$$5*0.75/2.23 + 8*0.78/2.23 + 3*0.4/2.23 + \dots$$