

TOPIC

Hệ thống gợi ý sản phẩm đơn giản (Recommendation System mini)

Bài toán

Doanh nghiệp muốn:

- Gợi ý sản phẩm cho khách hàng
- Tăng tỉ lệ mua hàng

Ý tưởng

Dựa trên **lịch sử mua hàng / click / rating** → gợi ý sản phẩm tương tự

Thuật toán

- Collaborative Filtering (User-based / Item-based)
- Cosine Similarity
- KNN
- (Nâng cao) Matrix Factorization

Stack

- Python
- Pandas, NumPy
- Scikit-learn
- Flask / FastAPI
- SQLite / CSV

Input – Output

- Input: user_id
- Output: Top 5 sản phẩm nên gợi ý

Mở rộng

- Dùng dữ liệu real (Shopee/Amazon sample)
- Thêm thời gian (recent > old)
- Dashboard cho doanh nghiệp

 **Rất hợp CV & phỏng vấn Data / Backend**

TASKS_NOTE

Khởi tạo folder dự án: RS-mini

Tạo file [README.md](#)

Tạo data

Tối thiểu cần 3 cột:

text

Sao chép mã

```
user_id | product_id | rating (hoặc interaction)
```

Ví dụ:

user_id	product_id	rating
U1	P1	5
U1	P2	3
U2	P1	4

- rating** có thể là:
 - số sao
 - 1 = mua
 - 0.5 = click
 - thời gian xem

↓

Build USER-ITEM MATRIX

```
self.user_item_matrix = None    # Bước 1: Khai báo biến rỗng
self.item_similarity = None     # Bước 2: Khai báo biến rỗng
self._build_matrix()           # Bước 3: Xây dựng ma trận user-item
self._build_similarity()       # Bước 4: Tính độ tương đồng giữa items
...
```

1. **user_item_matrix** - Ma trận User-Item

Là gì? Bảng thể hiện mối quan hệ giữa người dùng và sản phẩm

Ví dụ:

```
...
        Sản phẩm A  Sản phẩm B  Sản phẩm C
User 1      5      0      3
User 2      0      4      5
User 3      2      3      0
...
```

- Số = đánh giá/tương tác (0 = chưa xem)

2. **item_similarity** - Ma trận độ tương đồng

Là gì? Bảng cho biết sản phẩm nào giống sản phẩm nào

Ví dụ:

...

	Sản phẩm A	Sản phẩm B	Sản phẩm C
Sản phẩm A	1.0	0.3	0.8
Sản phẩm B	0.3	1.0	0.5
Sản phẩm C	0.8	0.5	1.0

...

- Giá trị gần 1 = rất giống nhau

- Tính bằng cosine_similarity

Luồng hoạt động:

...

Dữ liệu giao dịch

↓

_build_matrix() → Tạo ma trận User-Item

↓

_build_similarity() → Tính độ giống nhau giữa các sản phẩm

↓

Sẵn sàng gợi ý!

Ma trận chuyển vị T

def _build_similarity(self):

self.item_similarity = consine_sim(self.user_item_matrix.T)

~~~~~

Hàm tính Ma trận chuyển vị (T)

...

.T (Transpose) - Chuyển vị ma trận:

...

TRƯỚC (.T):

User-Item Matrix

SAU (.T):

Item-User Matrix

	A	B	C		User1	User2	User3
User1	5	0	3	A	5	0	2
User2	0	4	5	→ B	0	4	3
User3	2	3	0	C	3	5	0

Tại sao phải chuyển vị?

- Để so sánh **sản phẩm với sản phẩm** (theo cột)
- Mỗi hàng sau khi $\cdot \mathbf{T}$ = vector đánh giá của 1 sản phẩm từ tất cả users

OUTPUT

	user_id	product_id	rating
0	U1	P1	2
1	U2	P2	2
2	U3	P3	3
3	U3	P1	5
4	U2	P1	4

=== USER-ITEM MATRIX ===

product_id	P1	P2	P3
user_id			
U1	2.0	0.0	0.0
U2	4.0	2.0	0.0
U3	5.0	0.0	3.0

=== ITEM SIMILARITY MATRIX ===

product_id	P1	P2	P3
product_id			
P1	1.000000	0.596285	0.745356
P2	0.596285	1.000000	0.000000
P3	0.745356	0.000000	1.000000

=== RECOMMEND FOR U1 ===

product_id	
P3	1.490712
P2	1.192570

dtype: float64

1) Giả sử: **rating = số giờ xem**

use r	product	giờ xem
U1	P1	2
U2	P1	4
U2	P2	2
U3	P1	5
U3	P3	3

→ Đây là **implicit feedback**

(không phải thích/không thích, mà là *hành vi*)

2 USER-ITEM MATRIX nói gì?

	P1	P2	P3
U1	2	0	0
U2	4	2	0
U3	5	0	3

👉 User 1:

- Chỉ xem **P1 (2 giờ)**
- Chưa xem **P2, P3**

🎯 Mục tiêu hệ thống:

“Dựa vào P1 mà U1 đã xem, nên gợi ý P2 hay P3?”

3 ITEM SIMILARITY đang thể hiện điều gì?

Similarity

P1 – P2 = 0.596

P1 – P3 = 0.745

👉 Ý nghĩa cực kỳ quan trọng:

- Người xem **P1**
↳ thường cũng xem **P3** nhiều hơn **P2**

📌 Vì sao?

- U3 xem P1 (5h) và P3 (3h)
 - U2 xem P1 (4h) và P2 (2h)
 - Nhưng mức độ đồng thời P1–P3 mạnh hơn
-

4 Vì sao U1 được gợi ý P3 trước P2?

Công thức hệ thống đang làm (trực giác):

$\text{score}(\text{item}) = \text{similarity}(\text{item}, P1) \times \text{giờ xem } P1$

Tính thật luôn:

U1 xem P1 = 2 giờ

♦ P3

$$\begin{aligned}\text{score}(P3) &= \text{sim}(P3, P1) \times 2 \\ &= 0.745 \times 2 \\ &= 1.49\end{aligned}$$

♦ P2

$$\begin{aligned}\text{score}(P2) &= \text{sim}(P2, P1) \times 2 \\ &= 0.596 \times 2 \\ &= 1.19\end{aligned}$$

 Kết quả:

P3 > P2

5 Diễn giải bằng ngôn ngữ sản phẩm (rất quan trọng)

 Hệ thống đang nói với U1:

“Những người **giống bạn**,
sau khi xem **P1**,
thường **xem P3 nhiều hơn P2**
→ bạn nên thử **P3 trước**”

6 Nếu đây là doanh nghiệp thật thì sao?

 Nếu là video / phim

“Bạn xem phim này 2 giờ
→ người khác xem tiếp phim kia rất nhiều”

 Nếu là e-commerce

“Bạn dành nhiều thời gian xem sản phẩm A
→ khách tương tự thường quan tâm sản phẩm C”

🎵 Nếu là music / podcast

“Bạn nghe track này lâu
→ fan giống bạn hay nghe track kia”

7 Một điểm CỰC KỲ QUAN TRỌNG bạn vừa học được

- 👉 Recommender không gợi ý “phổ biến”
- 👉 Nó gợi ý theo hành vi giống nhau

U1 chưa xem P3
nhưng P3 liên quan mạnh đến thứ U1 đã xem

→ đây chính là **personalization**