Ứng dụng GNN cho bài toán gợi ý game trên nền tảng Steam

Học phần: Project II - **Mã học phần:** IT3930 - **Mã lớp:** 750645

Sinh viên thực hiện: Đặng Tiến Cường - **MSSV:** 20220020

Giảng viên hướng dẫn: PGS.TS Lê Đức Hậu

Tổng quan về bài toán gợi ý

Mục tiêu bài toán là học:

- Một hàm đánh giá mức độ phù hợp giữa người dùng (user) và sản phẩm (item)
- Một quan hệ thứ tự toàn phần trên tập sản phẩm được cá nhân hóa cho mỗi người dùng.

$$f : \mathcal{U} \times \mathcal{I} \to \mathbb{R} \qquad >_u \subseteq \mathcal{I} \times \mathcal{I}$$

Sao cho:

$$i >_u j \iff f(u,i) > f(u,j)$$

Hai hướng tiếp cận:

- **Learn to predict:** xem bài toán là một bài toán dự đoán độ ưa thích của người dùng đến sản phẩm
- **Learn to rank:** không quan tâm đến giá trị độ ưa thích tuyệt đối mà chỉ học thứ tự

Mô hình sẽ dự đoán k item mà user có khả năng thích nhất.

Loại dữ liệu trong hệ gợi ý

Dữ liệu đầu vào của dataset bao gồm 3 loại chính:

- **Thông tin của người dùng (user):** độ tuổi, giới tính, khu vực sinh sống, công việc,...
- **Thông tin của sản phẩm (item): tiêu đề, thể loại, mô tả nội dung, hình ảnh minh họa,...**
- **Lịch sử tương tác giữa user và item**, thường có dạng danh sách các bộ ba (user, item, interaction). Lịch sử này sẽ được chuyển về dạng ma trận tương tác $\mathbf{R} \in \mathbb{R}^{M \times N}$, trong đó mỗi cột ứng với lịch sử của mỗi user còn mỗi hàng ứng với các item.

Feedback thường có 2 loại dữ liệu:

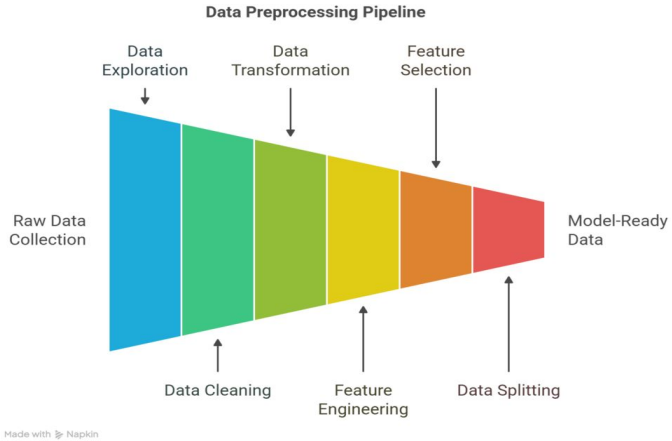
- **Explicit feedback** (phản hồi tường minh): đây là dữ liệu hỏi trực tiếp user để đánh giá về item, thường dưới dạng rating (0-5 sao) hoặc binary (like/dislike).
  - Là loại dữ liệu chất lượng cao, ít nhiễu nhưng khó kiếm do user thường không quan tâm
- **Implicit feedback** (phản hồi ẩn): đây là dữ liệu được thu thập từ hành vi của người dùng, ví dụ như số lần xem sản phẩm, thời lượng sử dụng,...
  - Sở thích của người dùng khó có thể trực tiếp suy luận ra từ implicit feedback.
  - Không có feedback tiêu cực.
  - Độ nhiễu rất cao.

Thu thập và tiền xử lý dữ liệu

Bộ dữ liệu: **Game Recommendation on Steam**

Tổng kích cỡ: **2.24** GB

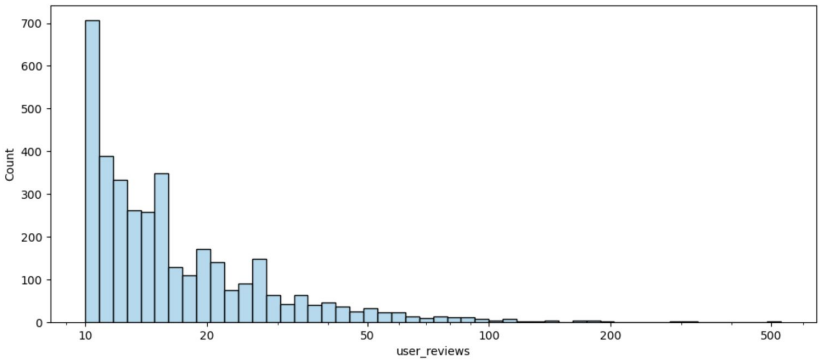
Tổng số bản ghi: **55.511.730** bản

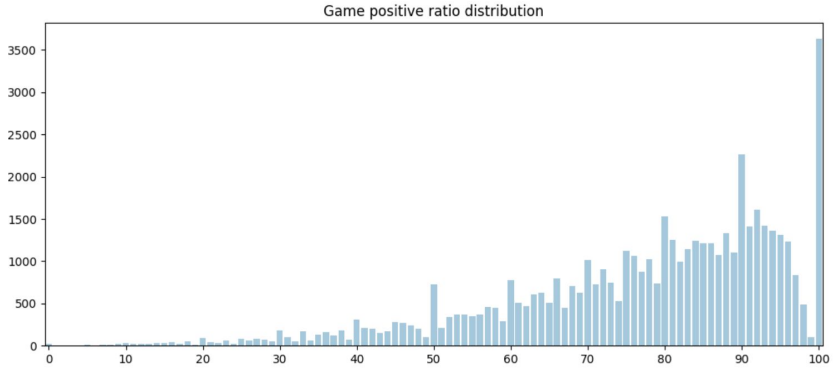


Đường ống (pipeline) tiền xử lý dữ liệu:

Exploration - **A**nalysis - **C**leaning - **E**ngineering - **P**reparation - **S**erving

Hiện tượng Long-tailed

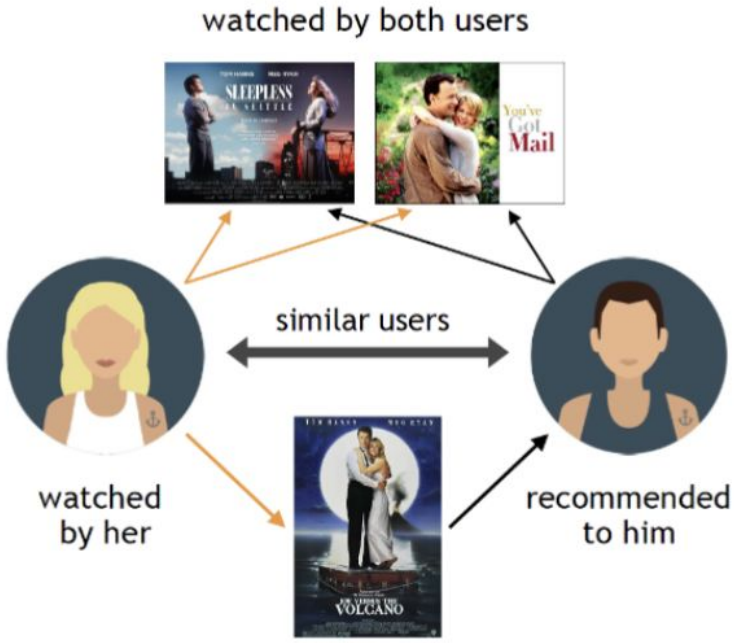




Hiện tượng thưa

0.0	2.56	0.0	-1.194	0.0	0.0	-3.087	-3.748
0.0	3.964	0.0	-3.984	3.771	0.0	0.0	-3.22
0.0	0.0	-2.856	-3.018	0.0	0.0	3.531	-0.726
2.754	0.0	2.062	0.0	2.803	0.0	2.893	0.0
3.763	0.0	3.539	0.0	0.0	-2.179	0.0	2.458
1.374	0.0	0.0	3.148	2.742	-1.552	0.0	0.0
-2.908	0.0	-1.436	0.0	0.0	3.201	0.0	-2.022
3.68	0.0	3.999	0.0	0.0	1.919	3.972	0.0

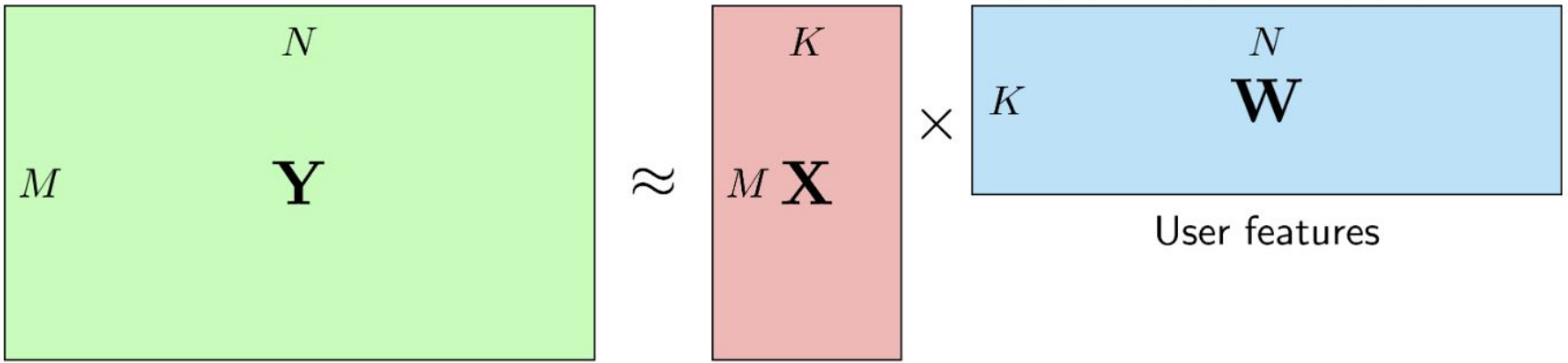
Hình 1.3: Content-based Filtering



Hình 1.4: Collaborative Filtering

Matrix Factorization (MF)

$$\mathbf{Y} \approx \hat{\mathbf{Y}} = \mathbf{X}\mathbf{W}$$



General Matrix Factorization (GMF)

1. **Đầu vào**

$$\mathbf{p}_u = \mathbf{P}^\top \mathbf{e}_u, \quad \mathbf{q}_i = \mathbf{Q}^\top \mathbf{e}_i$$

2. **Kết hợp**

$$\mathbf{z} = \phi(\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u \odot \mathbf{q}_i$$

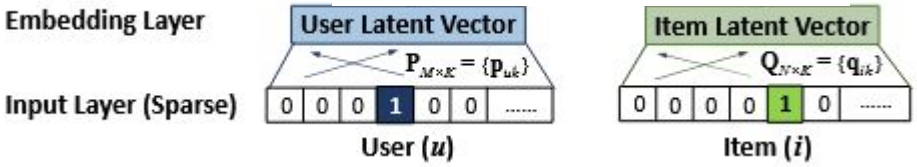
3. **Tổng hợp**

$$\hat{y}_{ui} = \mathbf{h}^T \mathbf{z} = \mathbf{h}^\top (\mathbf{p}_u \odot \mathbf{q}_i)$$

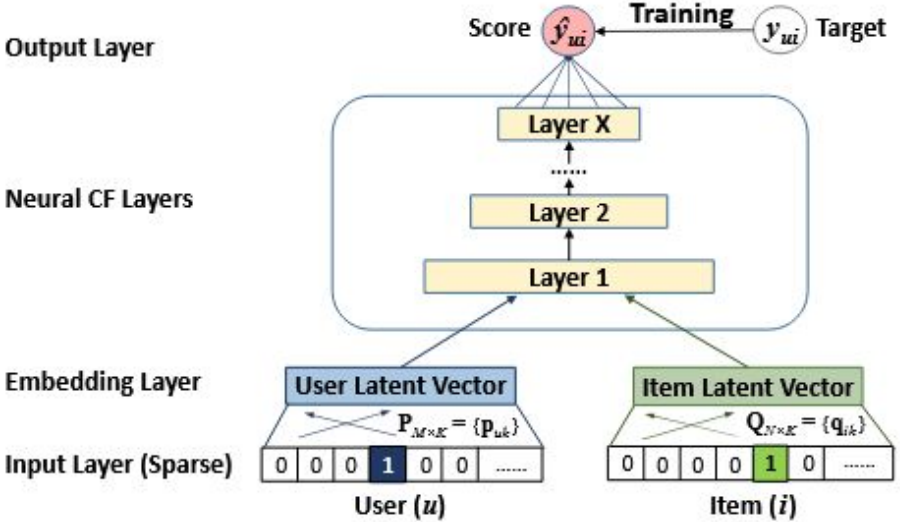
$\mathbf{h} = [1, 1, 1, ..., 1]$ ta có MF truyền thống

4. **Kích hoạt (Optional)**

$$\hat{y}_{ui} = a_{out}(\mathbf{h}^T (\mathbf{p}_u \odot \mathbf{q}_i))$$



Multi-layer Perceptron (MLP)



1. **Tầng đầu vào và tầng nhúng**

$$\mathbf{p}_u = \mathbf{P}^\top \mathbf{e}_u, \quad \mathbf{q}_i = \mathbf{Q}^\top \mathbf{e}_i$$

2. **Các tầng neuron**

$$\mathbf{z}_1 = \phi(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}$$

$$\phi_2(\mathbf{z}_1) = a_2(\mathbf{W}_2^T \mathbf{z}_1 + b_2)$$

.....

$$\phi_L(\mathbf{z}_{L-1}) = a_L(\mathbf{W}_L^T \mathbf{z}_{L-1} + b_L)$$

3. **Tầng đầu ra**

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \phi_L(\mathbf{z}_{L-1}))$$

Neural Matrix Factorization (NeuMF)

1. **GMF**

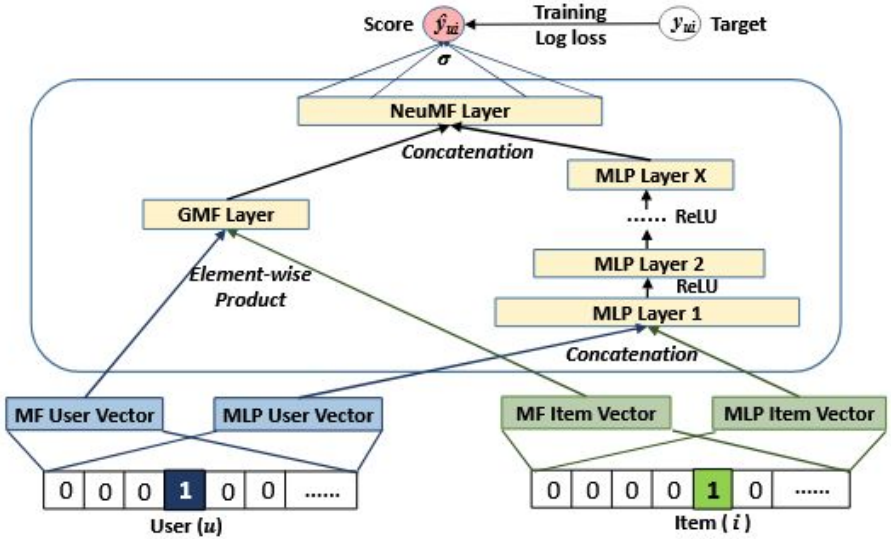
$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G$$

2. **MLP**

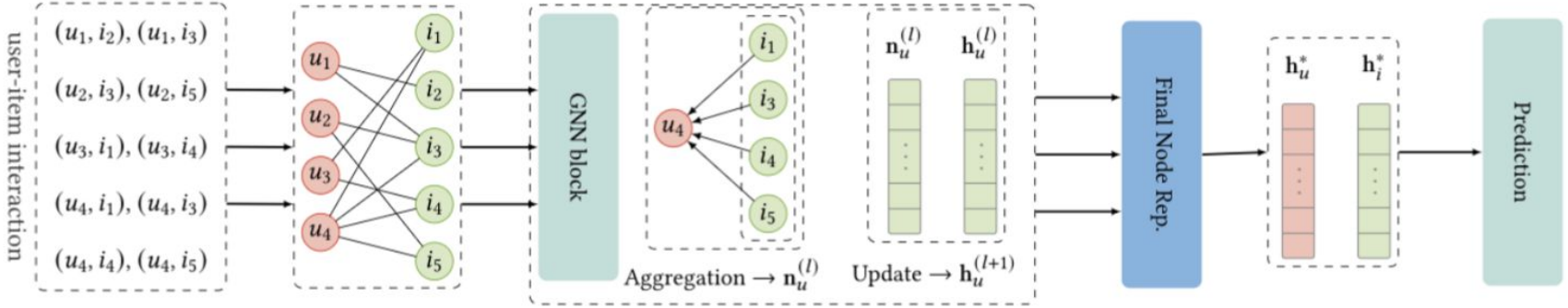
$$\phi^{MLP} = a_L(\mathbf{W}_L^T(a_{L-1}(...a_2(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + b_2)...)) + b_L)$$

3. **Tầng đầu ra**

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix})$$



GNN-based Models



Hình 3.8: Framework chung cho các mô hình GNN áp dụng cho bài toán hệ gợi ý dựa trên **lọc cộng tác**

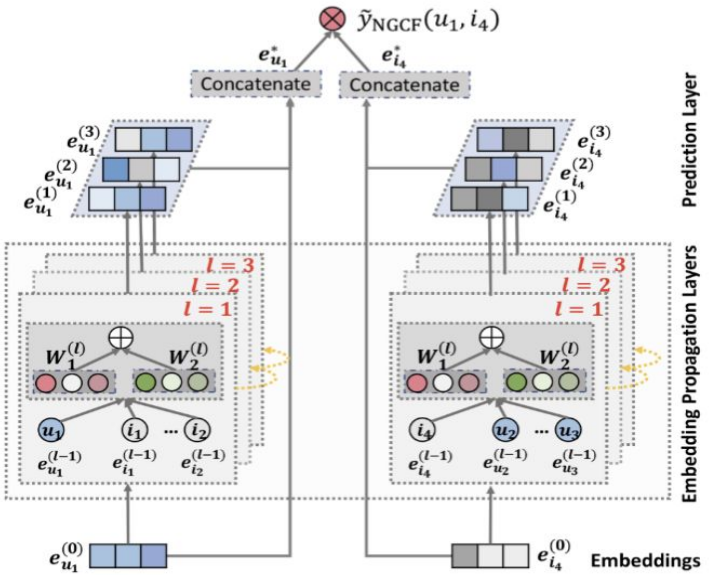
Ba thao tác chính trong GNN Block: Message, Aggregate, Update

$$\textbf{MESSAGE:} \quad \mathbf{m}_{u \leftarrow i}^{(l)} = \mathrm{MSG}^{(l)}\left(\mathbf{h}_u^{(l)}, \mathbf{h}_i^{(l)}, e_{ui}\right)$$

$$\textbf{AGGREGATE:} \quad \mathbf{n}_u^{(l)} = \mathrm{AGG}^{(l)}\left(\left\{\mathbf{m}_{u \leftarrow i}^{(l)} \mid i \in \mathcal{N}_u\right\}\right)$$

$$\textbf{UPDATE:} \quad \mathbf{h}_u^{(l+1)} = \mathrm{UPD}^{(l)}\left(\mathbf{h}_u^{(l)}, \mathbf{n}_u^{(l)}\right)$$

Kiến trúc mô hình NGCF

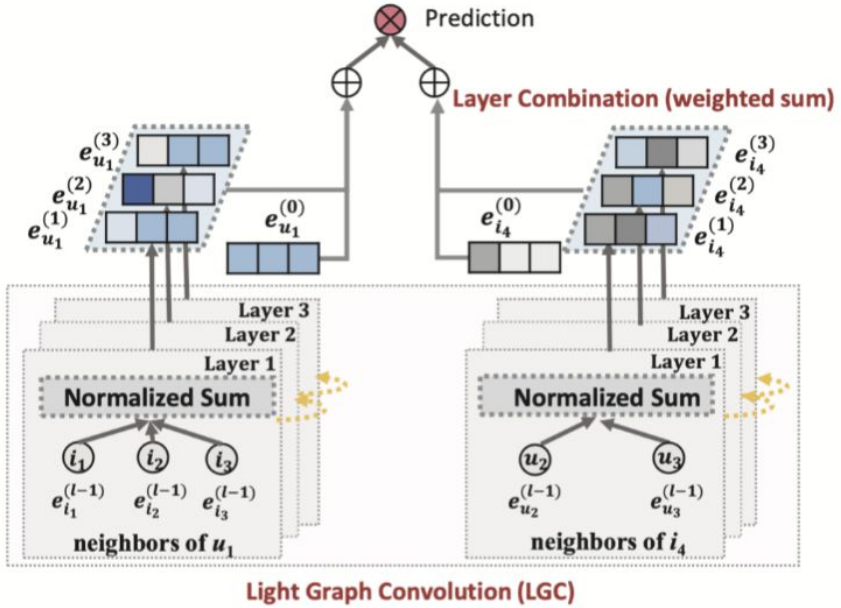


$$\textbf{MESSAGE}: \quad \mathbf{m}_{u\leftarrow i}^{(l)} = \mathbf{W}_1^{(l)}\,\mathbf{h}_u^{(l)} + \mathbf{W}_2^{(l)}\left(\mathbf{h}_u^{(l)} \odot \mathbf{h}_i^{(l)}\right)$$

$$\textbf{AGGREGATE}: \quad \mathbf{n}_u^{(l)} = \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\,\mathbf{m}_{u\leftarrow i}^{(l)}$$

$$\textbf{UPDATE}: \quad \mathbf{h}_u^{(l+1)} = \sigma\left(\mathbf{n}_u^{(l)}\right)$$

Kiến trúc mô hình LightGCN



$$\mathbf{h}_u^{(l+1)} = \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\,\mathbf{h}_i^{(l)}$$

$$\mathbf{h}_u^{*} = \sum_{l=0}^{L} \alpha^{(l)}\,\mathbf{h}_u^{(l)}, \quad \alpha_l = \frac{1}{L+1},$$

Hàm mất mát

Binary Cross Entropy Loss

$$\mathcal{L}_{\text{BCE}} = -\sum_{i=1}^{n} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

Bayesian Personalized Ranking Loss

$$\begin{aligned}
\mathcal{L}_{\text{BPR}} &= -\ln p(\Theta \mid >_u) \\
&= -\sum_{(u,i,j) \in \mathcal{D}_S} \ln \sigma(\hat{y}_{u,i} - \hat{y}_{u,j}) - \lambda_\Theta \|\Theta\|^2 \\
&= -\sum_{(u,i,j) \in \mathcal{D}_S} \ln \sigma(\hat{y}_{u,i,j}) - \lambda_\Theta \|\Theta\|^2
\end{aligned}$$

Tất cả mô hình được huấn luyện bằng hàm mất mất BPR để tối ưu cho mục tiêu xếp hạng.

Metric đánh giá

Precision@k:

$$\text{Precision@k} = \frac{TP}{TP + FP} = \frac{|\text{relevant items recommended}|}{|k|}$$

Recall@k:

$$\text{Recall@k} = \frac{TP}{TP + FN} = \frac{|\text{relevant items recommended}|}{|\text{all relevant items}|}$$

NDCG@k:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2(i+1)}, \quad \text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}},$$

Hit Rate @k:

$$\text{HitRate@k} = \begin{cases} 1, & |\text{relevant items recommended}| \geq 1, \\ 0, & \text{ngược lại.} \end{cases}$$

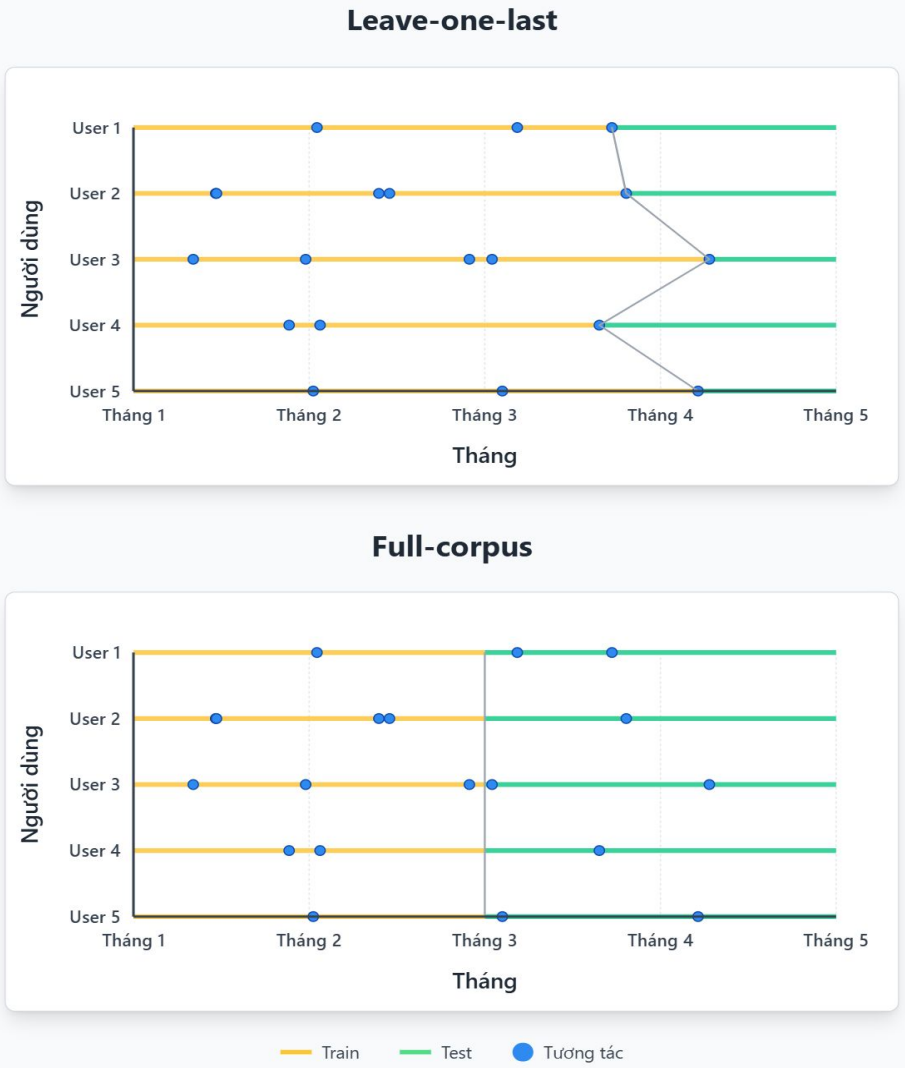
Chiến lược đánh giá

Leave-one-last:

- Phổ biến trong các nghiên cứu
- Khối lượng tính toán ít, tận dụng tối đa dataset cho train
- **Điểm yếu:** tiềm ẩn **time leakage**, **Precision@k** và **Recall@k** mất đi ý nghĩa

Full-corpus:

- Phản ánh chính xác môi trường thực tế
- Điểm yếu: Chi phí tính toán cao, nhiều cold-start, không thích hợp cho dataset có nhiều item



Kết quả đánh giá

Bảng 5.1: Kết quả đánh giá mô hình trên **Full-corpus** và **Leave-one-last**

Mô hình	Full-corpus				Leave-one-last	
	Precision@10	Recall@10	NDCG@10	HitRate@10	NDCG@10	HitRate@10
MF-BPR	0.0111	0.0123	0.0143	0.0927	0.1614	0.3151
NeuMF	0.0127	0.0140	0.0161	0.1029	0.1655	0.3146
NGCF	0.0244	0.0281	0.0315	0.1837	**0.2685**	**0.5165**
LightGCN	**0.0256**	**0.0298**	**0.0331**	**0.1906**	0.2681	0.5069