Proposals

Nhóm: ...

Đề tài (Topic)

Competition: Otto - Multi-Objective Recommender System (Kaggle)

Đề xuất Dự án (Project Proposal)

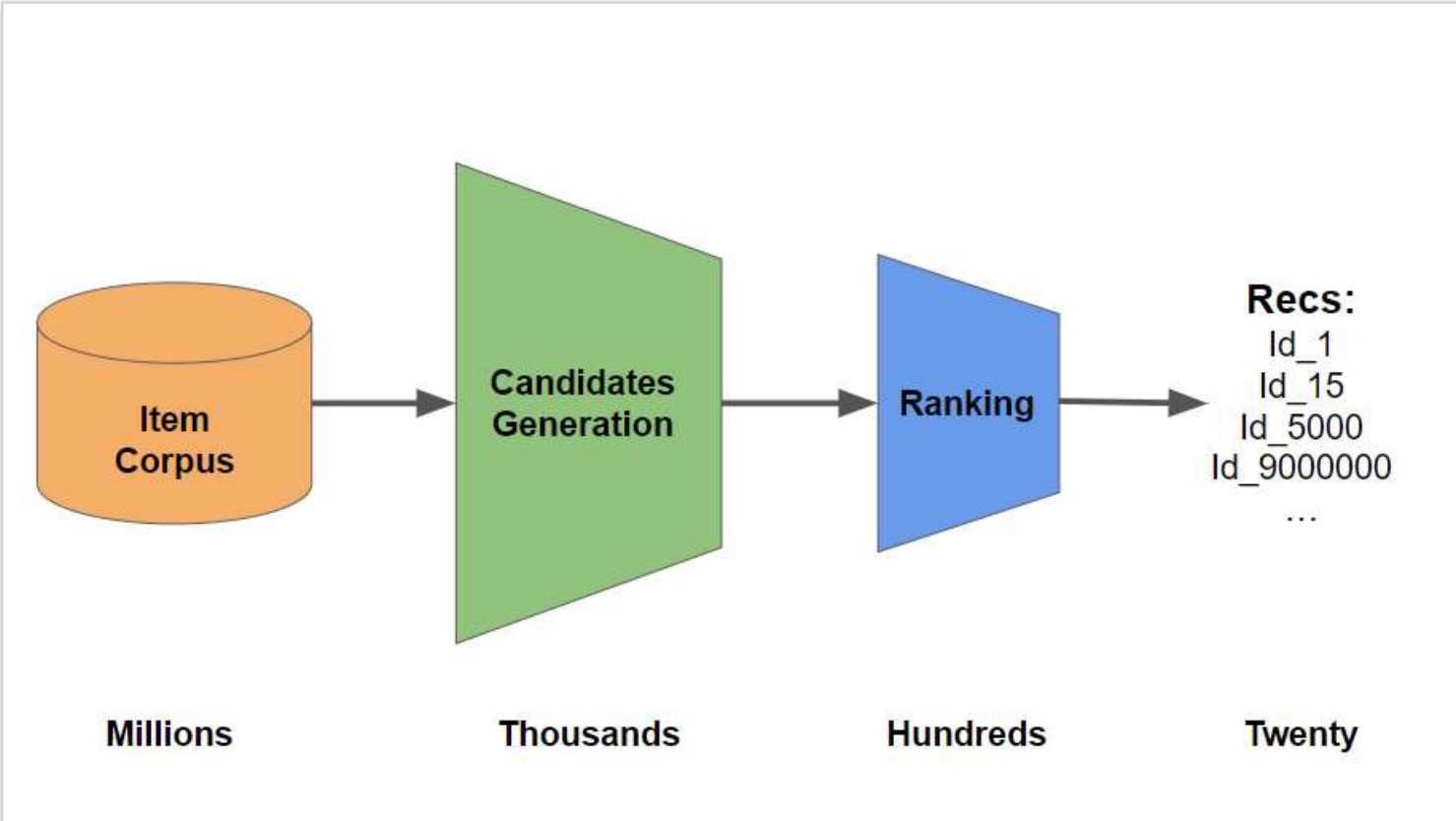
1. Bối cảnh và Giới thiệu (Introduction)

Cuộc thi "OTTO - Multi-Objective Recommender System" trên Kaggle đặt ra một bài toán thực tế và thách thức: xây dựng một hệ thống có khả năng dự đoán đồng thời ba hành động của người dùng (`clicks` , `carts` , `orders`). Project này sẽ thực hiện một nghiên cứu có hệ thống, bắt đầu từ việc tái tạo các giải pháp hàng đầu, sau đó đề xuất và thử nghiệm các tinh chỉnh tiềm năng.

2. Kiến trúc Nền tảng: Mô hình Candidate-ReRank

Với một danh mục sản phẩm khổng lồ (hơn 1.8 triệu `aid`), việc sử dụng một mô hình phức tạp để chấm điểm và xếp hạng tất cả các sản phẩm cho mỗi người dùng là bất khả thi về mặt tính toán. Để giải quyết thách thức về quy mô này, các hệ thống gợi ý hiện đại, bao gồm các giải pháp hàng đầu trong cuộc thi OTTO, đều dựa trên một kiến trúc hai giai đoạn được gọi là **Candidate-ReRank**.

- **Giai đoạn 1: Candidate Generation (Tạo Ứng viên):** Đây là giai đoạn tập trung vào **Recall**. Mục tiêu là nhanh chóng sàng lọc từ 1.8 triệu sản phẩm xuống một danh sách ngắn (vài trăm) các ứng viên có khả năng liên quan cao nhất. Giai đoạn này ưu tiên tốc độ và đảm bảo không bỏ sót các gợi ý tiềm năng. Các kỹ thuật phổ biến ở giai đoạn này bao gồm Ma trận Co-visitation và các mô hình học biểu diễn (Embedding-based models) như MLP, BERT4Rec, GNN.
- **Giai đoạn 2: Re-ranking (Tái xếp hạng):** Đây là giai đoạn tập trung vào **Precision**. Với một danh sách ứng viên đã được thu hẹp, giai đoạn này có thể sử dụng một mô hình phức tạp hơn (ví dụ: LGBMRanker) với hàng trăm đặc trưng chi tiết để chấm điểm và sắp xếp lại các ứng viên một cách cực kỳ chính xác. Mục tiêu là chọn ra 20 gợi ý tốt nhất từ danh sách ứng viên để đưa ra dự đoán cuối cùng.



Kiến trúc này tạo ra một hệ thống vừa hiệu quả về mặt tính toán, vừa có độ chính xác cao, và sẽ là kiến trúc nền tảng cho tất cả các thử nghiệm trong dự án này.

3. Phân tích các Giải pháp Hiện có

3.1. Giải pháp Gốc (Baseline - Notebook "Candidate ReRank model")

- **Kiến trúc:** Candidate-ReRank 2 giai đoạn.
- **Candidate Generation:** Sử dụng 5 nguồn ứng viên bao gồm lịch sử người dùng, top phổ biến, và 3 loại ma trận Co-visitation được tối ưu hóa.
- **Re-ranking:** Điểm yếu lớn nhất. Sử dụng một bộ các luật thủ công (`if-else`) để sắp xếp và lựa chọn.
- **Kết luận:** Là một điểm khởi đầu quan trọng với phần tạo ứng viên chất lượng, nhưng cần một bộ não Re-ranker thông minh hơn.

3.2. Giải pháp Vô địch (1st Place Solution)

- **Kiến trúc:** Candidate-ReRank 2 giai đoạn.
- **Candidate Generation:** Sử dụng kiến trúc **"Hybrid"**, kết hợp Co-visitation và một mô hình **Neural Network (MLP/Transformer)** để học embeddings. Điều này giúp nắm bắt cả các mối quan hệ trực tiếp và các mối quan hệ ẩn.
- **Re-ranking:** Sử dụng một **ensemble LGBMRanker** với một bộ ~200 features được thiết kế chi tiết.

- **Kết luận:** Sức mạnh đến từ sự đa dạng của nguồn ứng viên và một mô hình Re-ranker cực kỳ mạnh mẽ, nhất là Feature Engineering. Project sẽ cố gắng tái tạo một phần các feature quan trọng theo ý tưởng của 1st Place Solution.

3. Đề xuất Cải tiến và Lộ trình Thực nghiệm

Dự án sẽ được triển khai theo các cột mốc, trong đó mỗi cột mốc là một thử nghiệm được xây dựng dựa trên thành công của cột mốc trước đó.

Cột mốc 0: Xây dựng Pipeline Nền tảng

- **Hành động:** Tái cấu trúc baseline, thiết lập quy trình Local Validation. Thay thế các luật thủ công bằng một mô hình **LGBMRanker** mạnh mẽ.
- **Mục tiêu:** Tạo ra một baseline vững chắc với "bộ não" Machine Learning, làm cơ sở để so sánh tất cả các cải tiến sau này.

Cột mốc 1: Tái tạo Kiến trúc Hybrid 1st Solution (với MLP)

- **Hành động:**
  i. Huấn luyện một mô hình **MLP** đơn giản để học `item` và `session` embeddings, mô phỏng theo giải pháp vô địch.
  ii. Sử dụng thư viện **FAISS** để xây dựng index và tạo ra nguồn ứng viên mới từ các embedding này một cách hiệu quả.
  iii. **Ensemble** các ứng viên từ Co-visitation và MLP+FAISS.
  iv. Cập nhật Re-ranker với các feature mới (`mlp_rank` , `mlp_similarity_score`).
- **Mục tiêu:** Đạt được hiệu suất gần tương đương với 1st Solution (do kém ở Feature Engineering)

Cột mốc 2: Cải tiến với các Mô hình Deep Learning Nâng cao

- **Giả thuyết:** Các mô hình phức tạp hơn có thể nắm bắt các mẫu hình mà MLP bỏ lỡ.
- **Thử nghiệm 2A (Nâng cấp Tuần tự): BERT4Rec**
  - **Hành động:** Thay thế MLP bằng **BERT4Rec**. BERT4Rec có khả năng hiểu sâu hơn về **ngữ cảnh và thứ tự** trong một session.
  - **So sánh:** Đánh giá xem việc hiểu sâu hơn về trình tự có mang lại cải thiện so với MLP hay không.
- **Thử nghiệm 2B (Nâng cấp Cấu trúc): LightGCN**
  - **Hành động:** Xây dựng đồ thị item-item và huấn luyện **LightGCN**. GNN giải quyết vấn đề **"tầm nhìn đường hầm"** của các mô hình tuần tự bằng cách học từ cấu trúc quan hệ toàn cục.
  - **So sánh:** Đánh giá xem việc bổ sung "tầm nhìn toàn cảnh" từ GNN có tạo ra các ứng viên chất lượng và đa dạng hơn so với các phương pháp chỉ dựa trên session hay không.

Cột mốc 3: Đổi mới về Kiến trúc Hệ thống (Nếu còn thời gian)

- **Hành động:** Hiện thực hóa ý tưởng **"Adaptive Candidate Generation"**. Xây dựng một cơ chế retry, nơi Re-ranker có thể yêu cầu bổ sung ứng viên (ví dụ, từ nguồn GNN) nếu chất lượng của danh sách hiện tại không đạt ngưỡng.
- **Mục tiêu:** Chuyển đổi pipeline từ một luồng "tĩnh" sang một hệ thống "động", có khả năng tự điều chỉnh, một khái niệm rất tiên tiến.

4. Phương pháp Đánh giá

1. **Điểm số Recall@20 (Local CV):** Thước đo chính để so sánh hiệu quả giữa các thử nghiệm. Sử dụng cách chia train/valid đã được kiểm nghiệm từ cộng đồng trong competition.
2. **Điểm LB (Leaderboard):** Thước đo trên bộ test chính thức (leaderboard).
3. **Phân tích Chi tiết:** So sánh recall trên từng loại (`clicks` , `carts` , `orders`) và phân tích các trường hợp lỗi để hiểu sâu hơn về điểm mạnh/yếu của mỗi phương pháp.
4. **Phân tích Trade-off:** Ghi nhận và so sánh thời gian training/inference để đánh giá sự đánh đổi giữa độ chính xác và chi phí tài nguyên cho mỗi kiến trúc được đề xuất.

5. Công cụ & Môi trường

- **Môi trường:** Kaggle Notebooks (Tận dụng GPU và môi trường cài đặt sẵn).
- **Ngôn ngữ:** Python.
- **Thư viện chính:** Polars, RAPIDS cuDF, PyTorch, PyTorch Geometric, FAISS, Scikit-learn, LightGBM.