

BÁO CÁO KHOÁ LUẬN TỐT NGHIỆP

Hệ thống đề xuất lai kết hợp Neural Network và tích hợp Metadata (LightFM + CNNs)

Em đã xây dựng và hoàn thiện mô hình gợi ý sản phẩm cho người dùng dựa trên interaction (view, addtocart, transaction) và sử dụng thêm các thuộc tính bổ sung (metadata) để học và cải thiện gợi ý cho người dùng theo tính cá nhân hoá.

1. Đầu tiên em sửa dụng mô hình LightFM – chuyên cho dữ liệu implicit feedback, để trích xuất **user_embeddings** và **item_embeddings** từ tập dữ liệu. Sau khi train xong mô hình LightFM đó em thu được kết quả

Precision@5: 0.0703

Recall@5: 0.2889

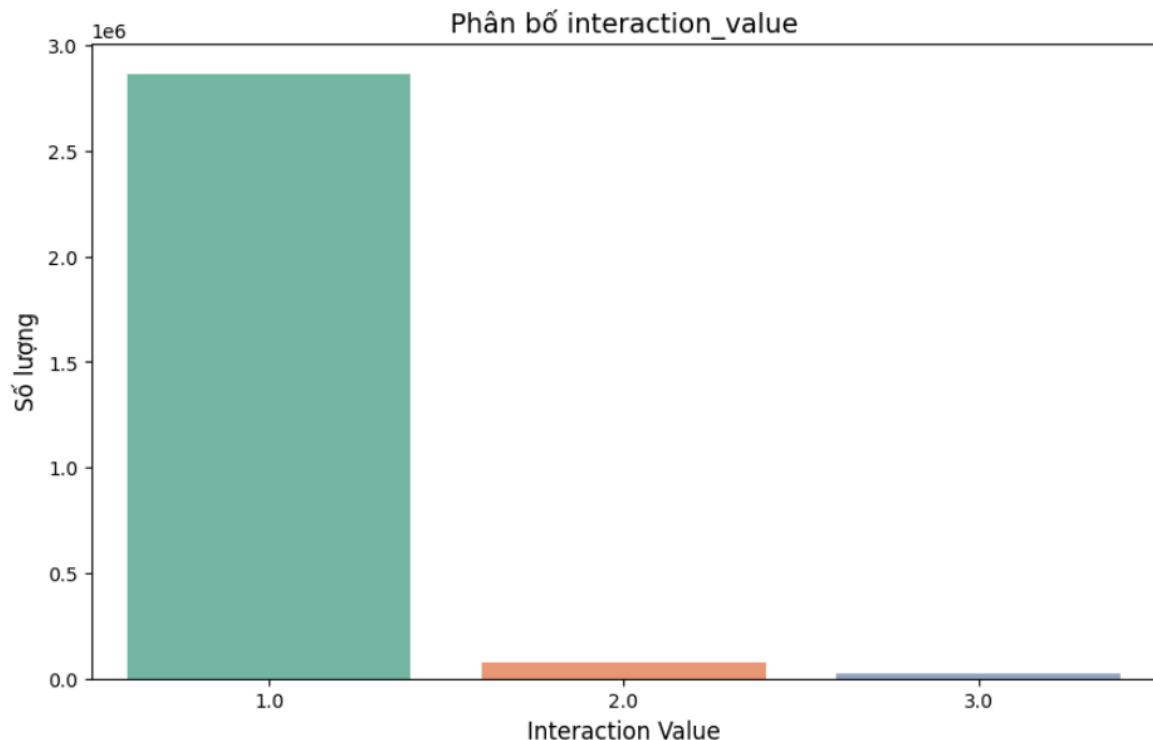
AUC: 0.8575

Và

Shape of user embedding: (1407580, 128)

Shape of item embedding: (235061, 128)

2. Do dataset từ thực tế nên việc mất cân bằng và dữ liệu thừa là bình thường. Nên em đã dùng kỹ thuật **focal loss with class weights** để xử lý vấn đề mất cân bằng giữa các hành vi của khách hàng:



Do dữ liệu thực tế nên view chiếm số lượng áp đảo so với addtocart và transaction.

3. Tiếp đến em tạo config metadata (thông tin giữa user và item) để dễ maintain sau này. Hiện tại đang có 17 thuộc tính bao gồm timestamp, price, hour, dayofweek, age, gender, session, ... đã được xử lý và chuẩn hoá cho mô hình học sâu.
4. Xây dựng mô hình học sâu:
 - Embedding: dùng embedding cho đặc trưng rời rạc, tải trọng số embedding từ trước cho user và product
 - Tầng Conv1D + GlobalMaxPooling1D áp dụng cho embedding của user và product → trích xuất đặc trưng cục bộ từ vector nhúng
 - Thực hiện phép nhân phân tử giữa user và product embedding để mô hình hóa tương tác.
 - Áp dụng Dense + BatchNormalization + Dropout cho các đặc trưng liên tục.
 - Kết hợp tất cả embedding, tương tác và đặc trưng liên tục thành một vector đặc trưng tổng hợp.
 - Chuỗi tầng Dense (512 → 256 → 128 → 64) với regularization (L2), BatchNorm và Dropout.
 - Dùng softmax cho phân loại 3 lớp. (view, addtocart, transaction)

Total params: 211,054,563 (805.11 MB)
Trainable params: 610,115 (2.33 MB)
Non-trainable params: 210,444,448 (802.78 MB)

5. Hiện tại do hết thời gian train nên em chưa fine-tuning nên kết quả chưa được cải thiện.

Kết quả huấn luyện:

Test Loss: 0.4054, Test Accuracy: 0.6484, Test AUC: 0.8027, Test Precision : 0.9711, Test Recall: 0.3869

Với AUC → Mô hình đạt hiệu suất tốt, khả năng phân biệt cao

Mặc dù Precision rất cao (0.9711), nhưng Recall thấp (0.3869) → mô hình dự đoán đúng rất chính xác, nhưng bỏ sót nhiều trường hợp dương tính

Accuracy đạt 64.84% và loss là 0.4054, phản ánh hiệu quả tổng thể vừa phải.

```

classification_report:
              precision    recall  f1-score   support

     0       0.9881      0.6561      0.7886     573623
     1       0.0373      0.3699      0.0678      14577
     2       0.0405      0.5706      0.0757       4807

 accuracy          0.6484     593007
 macro avg       0.3553      0.5322      0.3107     593007
 weighted avg    0.9571      0.6484      0.7651     593007

```

Validation Loss: 0.4057, Validation Accuracy: 0.6472, Validation AUC: 0.8016,
Validation Precision: 0.9708, Validation Recall: 0.3859

Test Loss: 0.4054, Test Accuracy: 0.6484, Test AUC: 0.8027, Test Precision: 0.9711, Test
Recall: 0.3869

Train Loss: 0.3930, Train Accuracy: 0.6560, Train AUC: 0.8121, Train Precision: 0.9760,
Train Recall: 0.4053

6. Ngoài dữ liệu Implicit feedback, còn huấn luyện thêm mô hình gợi ý cho Explicit Feedback

Tương tự như mô hình trên, em dùng mô hình Surprise chuyên cho dữ liệu Explicit (Rating từ 1 đến 5) để trích xuất embedding và dùng nó cho mô hình học sâu:

Với mô hình SVD để trích xuất embedding sau khi train thu được kết quả:

Test RMSE: 1.0288059450509446

Test MAE : 0.7111821952053239

User embeddings shape: (199888, 128)

Item embeddings shape: (63800, 128)

Đối với mô hình học sâu thì cấu trúc tương tự nhưng metadata sẽ có chút thay đổi cho phù hợp với dữ liệu

Total params: 34,358,785 (131.07 MB)
Trainable params: 601,601 (2.29 MB)
Non-trainable params: 33,757,184 (128.77 MB)

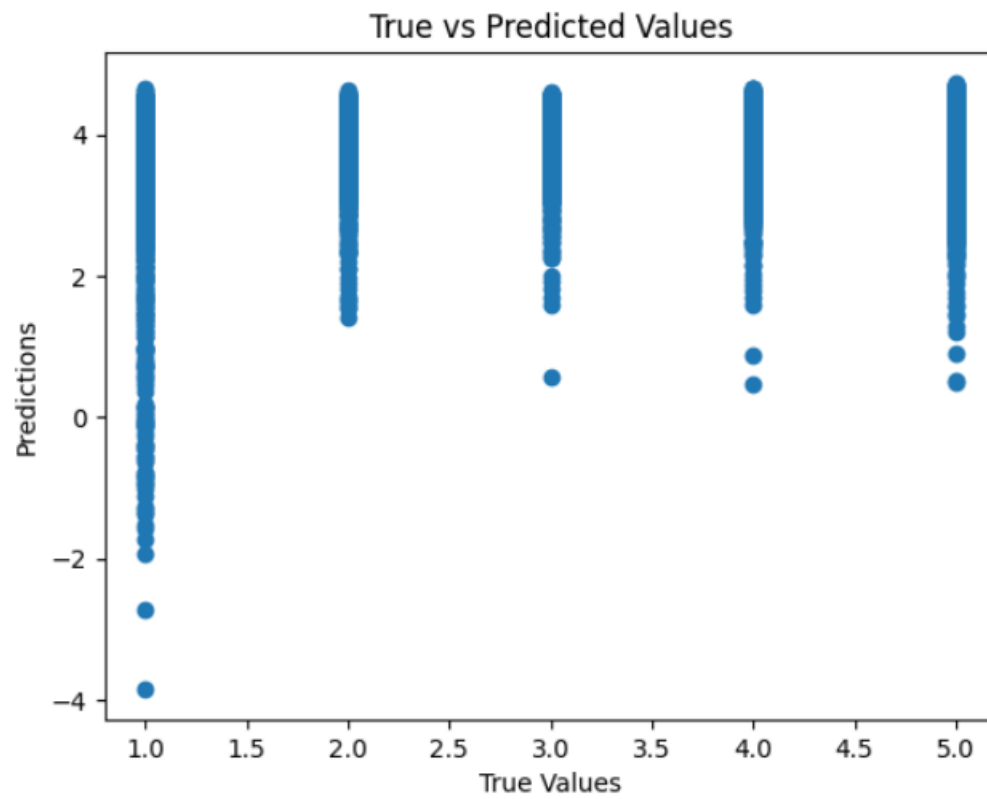
Kết quả:

Test Loss: 1.6894495487213135

Test MAE: 1.0197868347167969

Test MSE: 1.6628917651739987

Test R²: 0.02341184400144314



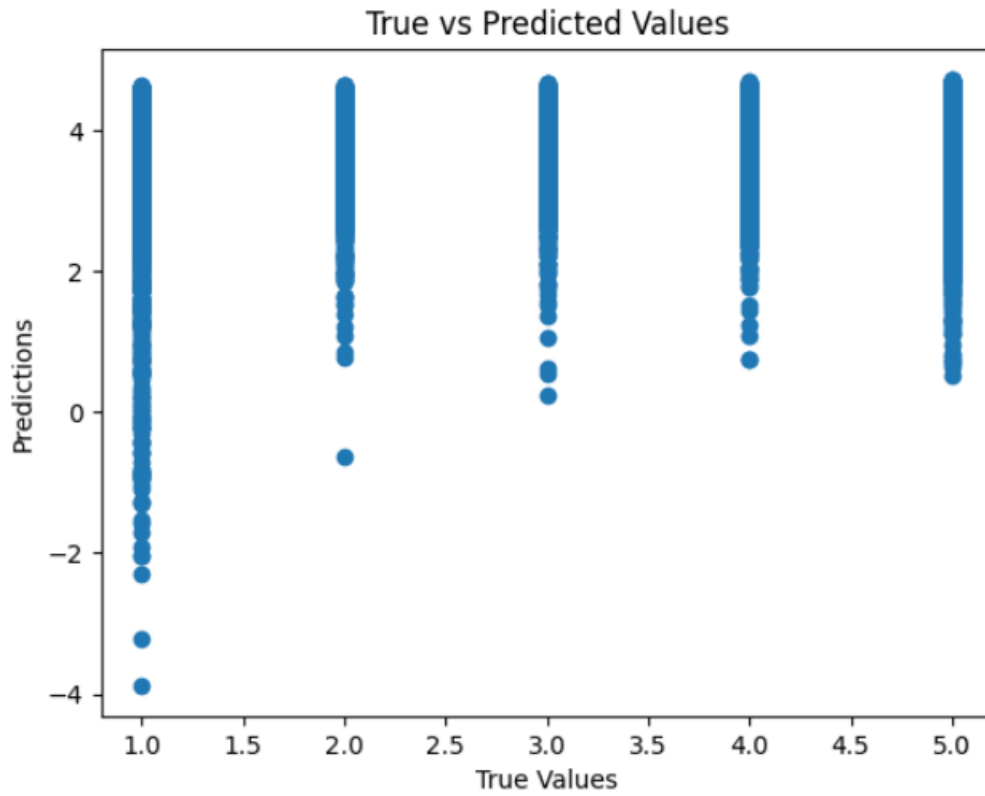
Sau khi fine-tuning mô hình:

Test Loss: 1.6850906610488892

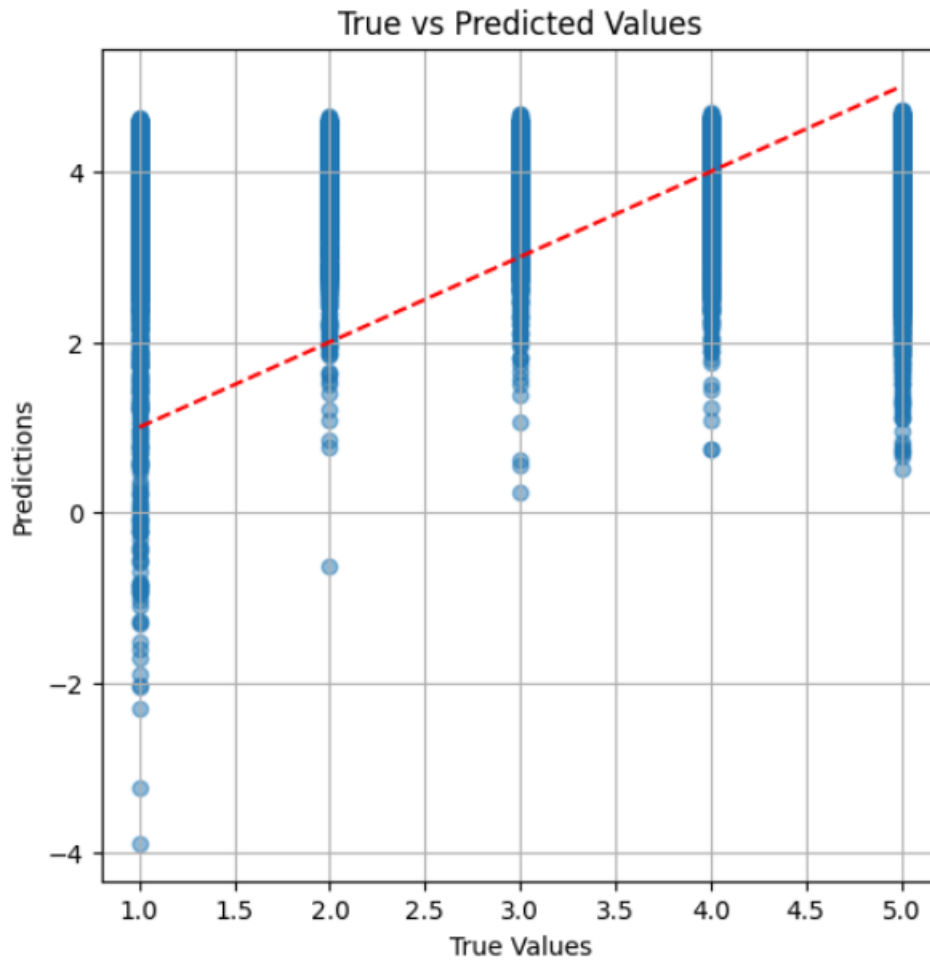
Test MAE: 1.006415843963623

Test MSE: 1.667323549975893

Test R^2 : 0.02080913188384992



Train Loss: 1.5786, MAE: 0.9760
Validation Loss: 1.5786, MAE: 0.9760
Test Loss: 1.6851, MAE: 1.0064
Test MAE (sklearn): 1.0064
Test RMSE: 1.2912
R² Score: 0.0208



So sánh giá trị thực tế và dự đoán:

	Actual	Predicted
0	5.0	4.424239
1	5.0	4.182939
2	3.0	4.423306
3	2.0	4.043856
4	5.0	4.002339
5	4.0	4.361325
6	5.0	4.527347
7	5.0	4.383998
8	4.0	4.340656
9	5.0	4.500286

Top 5 sản phẩm được gợi ý cho user 2134:

1. Item ID: 29832 | Predicted Rating: 4.6341
2. Item ID: 15233 | Predicted Rating: 4.6170
3. Item ID: 28193 | Predicted Rating: 4.6138
4. Item ID: 6187 | Predicted Rating: 4.6134

5. Item ID: 9206 | Predicted Rating: 4.6130

Ngoài ra em còn đánh giá mô hình với vấn đề cold start:

1. Tạo dữ liệu cold-start:

Cold-start user (test): 40052 samples

Cold-start item (test): 8137 samples

Non Cold-start item (test): 69778 samples

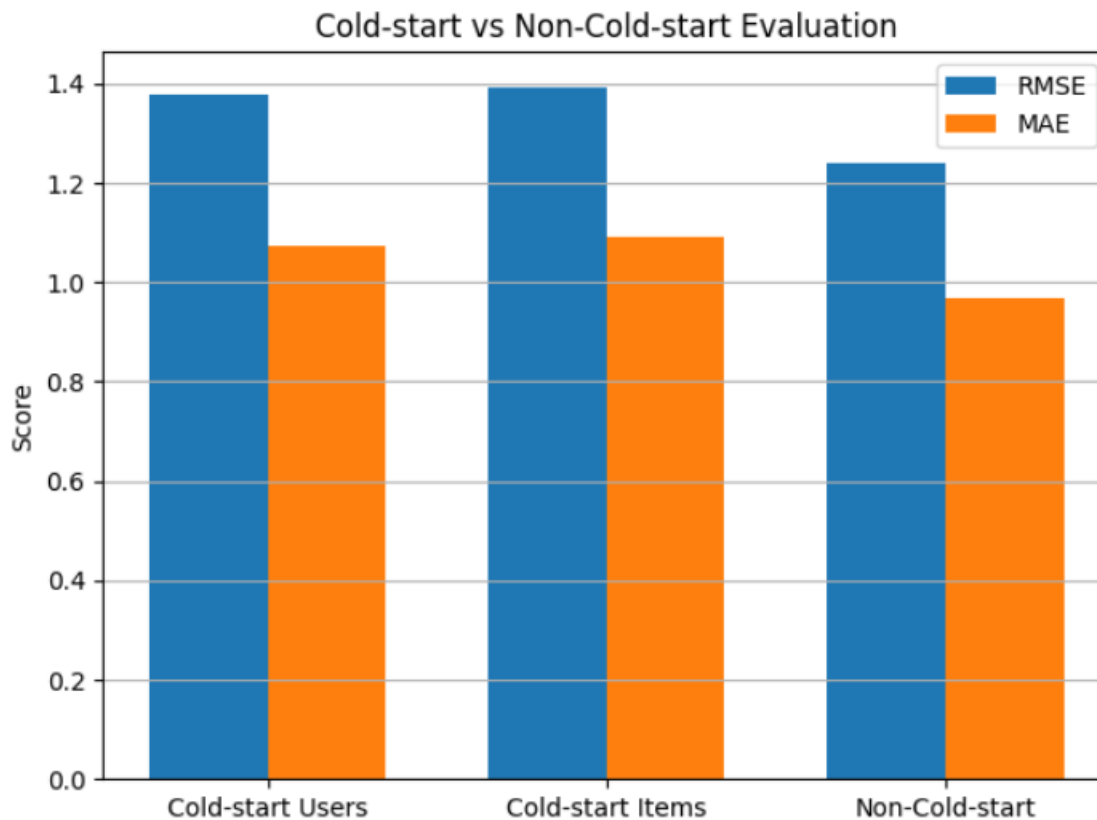
2. Đánh giá:

Cold-start Evaluation Results:

[Cold-start Users] RMSE: 1.3768, MAE: 1.0733

[Cold-start Items] RMSE: 1.3939, MAE: 1.0919

[Non-Cold-start] RMSE: 1.2385, MAE: 0.9669



Kết quả đánh giá cho thấy mô hình đạt hiệu quả dự đoán tốt hơn trong tình huống Non-Cold-start so với các trường hợp Cold-start. Cụ thể, sai số RMSE và MAE đều thấp hơn rõ rệt khi hệ thống đã có dữ liệu về người dùng và sản phẩm. Ngược lại, hiệu suất giảm đáng kể trong các trường hợp Cold-start, đặc biệt với sản phẩm mới, phản ánh thách thức phổ biến trong hệ thống gợi ý khi xử lý dữ liệu mới hoặc chưa đủ thông tin.