Báo cáo thu hoạch Quý 2

1. Các công việc thực hiện

Trong quý 2 vừa rồi thì em được tham gia vào 2 dự án là MyViettel và dự án recommend của TV360. Sau đây là một số công việc mà em đã thực hiện trong quý 2 như sau:

1.1 Một số công việc của dự án MyViettel mà em thực hiện như sau:

a) Xây dựng model mới về dự đoán nhu cầu Bất động sản

Thực hiện xây dựng thử nghiệm model dự đoán tập người dùng có nhu cầu quan tâm đến bất động sản để quảng cáo đến tập khách hành này quan ứng dụng MyViettel

Đánh giá tập base người dùng:
Đánh giá tập user từ tháng N – k đến tháng N – 1 có hành vi truy cập URL với condition cover được bao nhiêu % tập user tháng N có truy cập bất động sản.

K	**2**	**3**	**4**	**5**	**6**	**7**	**Condition**
Ratio	0.527	0.588	0.63	0.656	0.674	0.669	Topic = 'BAT DONG SAN'
Ratio	**0.899**	**0.913**	**0.9205**	**0.923**	-	-	Topic = 'BAT DONG SAN' or topi = 'TIN DUNG' or topic = 'NGAN HANG'
Ratio	0.57	0.632	0.673	0.70	-	-	Topic = 'BAT DONG SAN' or topi = 'TIN DUNG' or (topic = 'NGAN HANG' and service in ('GOI VAY'

							‘VAY THE CHAP’, ‘VAY DA NGAY’))

Tỷ lệ nhãn:

	Count 0	Count 1	Ratio
Train	48210664	3777501	12.8 : 1
Test	14652081	1065425	13.75 : 1

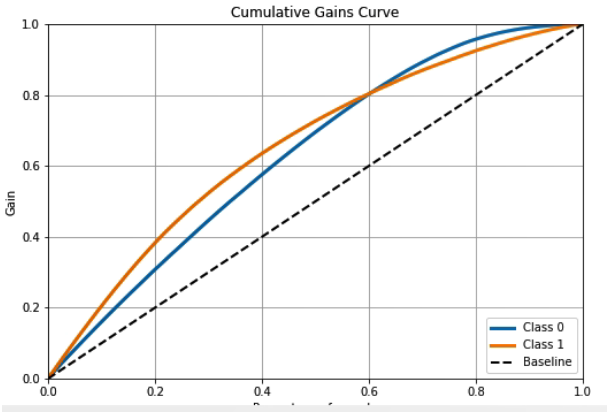
Kết quả thu được như sau:

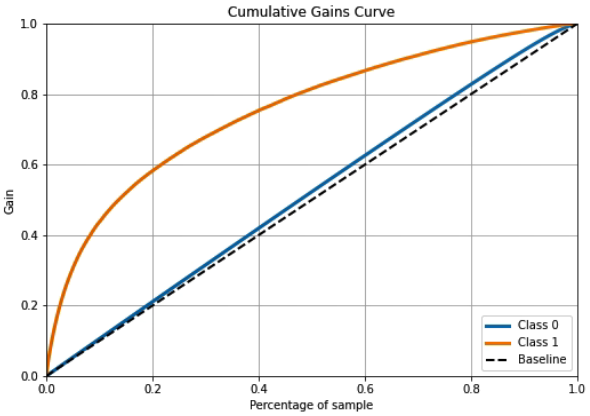
P: Precision Score
R: Recall Score

		Train					Test				
Deepth	Scale	P0	R0	P1	R1	AUC	P0	R0	P1	R1	AUC
10	8	0.748	0.894	0.755	0.52	0.789	0.958	0.915	0.282	0.450	0.773
10	11	0.705	0.842	0.764	0.590	0.789	0.961	0.879	0.237	0.515	0.773
15	11	0.715	0.846	0.773	0.610	0.803	0.962	0.872	0.23	0.526	0.773

- **Cumulative Gains Curve**

Train **Test**



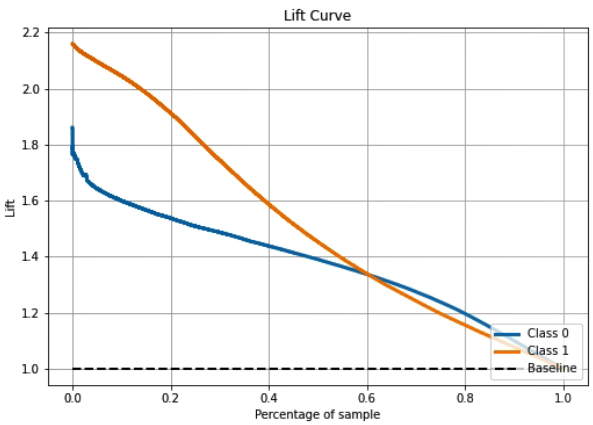


Nhận xét:

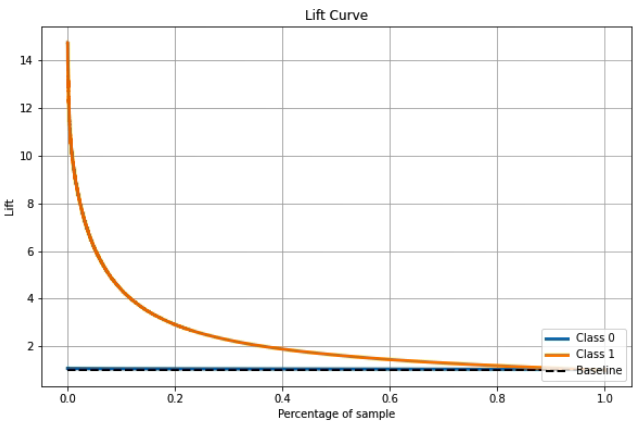
Từ biểu đồ Gain Curve của tập test trên, nhận thấy rằng top 10% isdn (tương đường khoảng 1M user) có score cao nhất cover được ~40% tập user(tương đương 760k user) truy cập BDS vào tháng tiếp theo

- **Lift curve**

Train

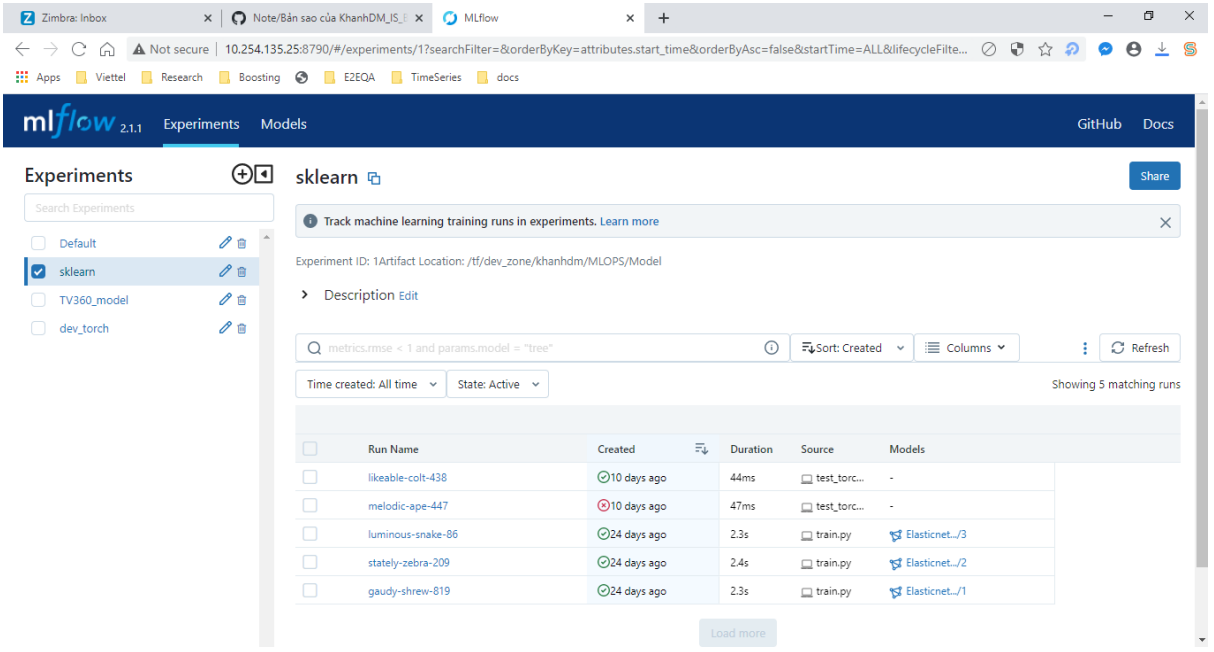


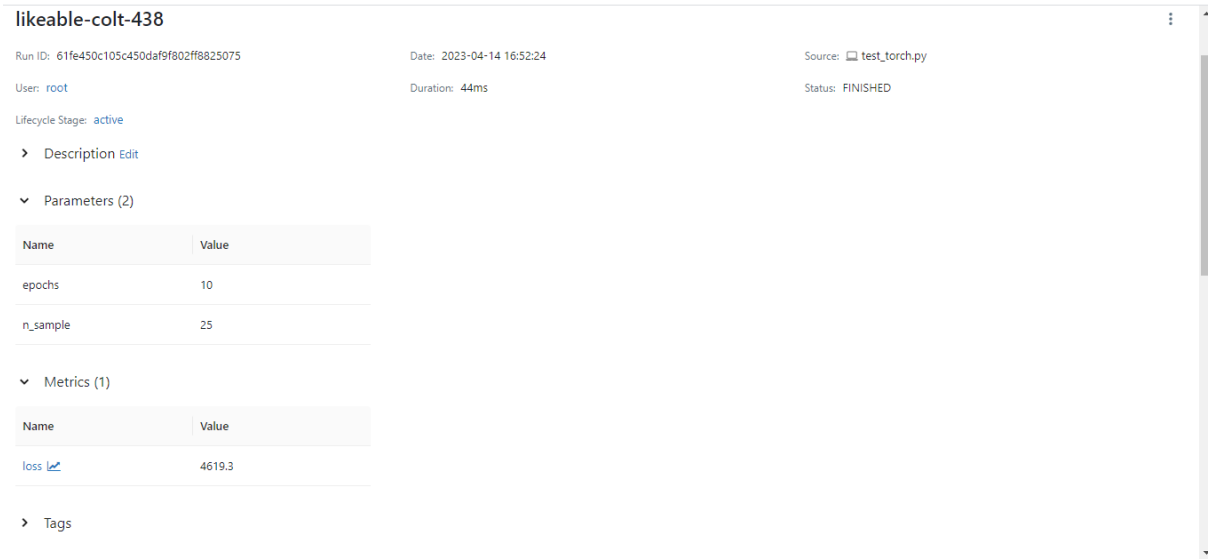
Test



b) Setup thử nghiệm MLFlow cho chức năng tracking model experiment, model registry

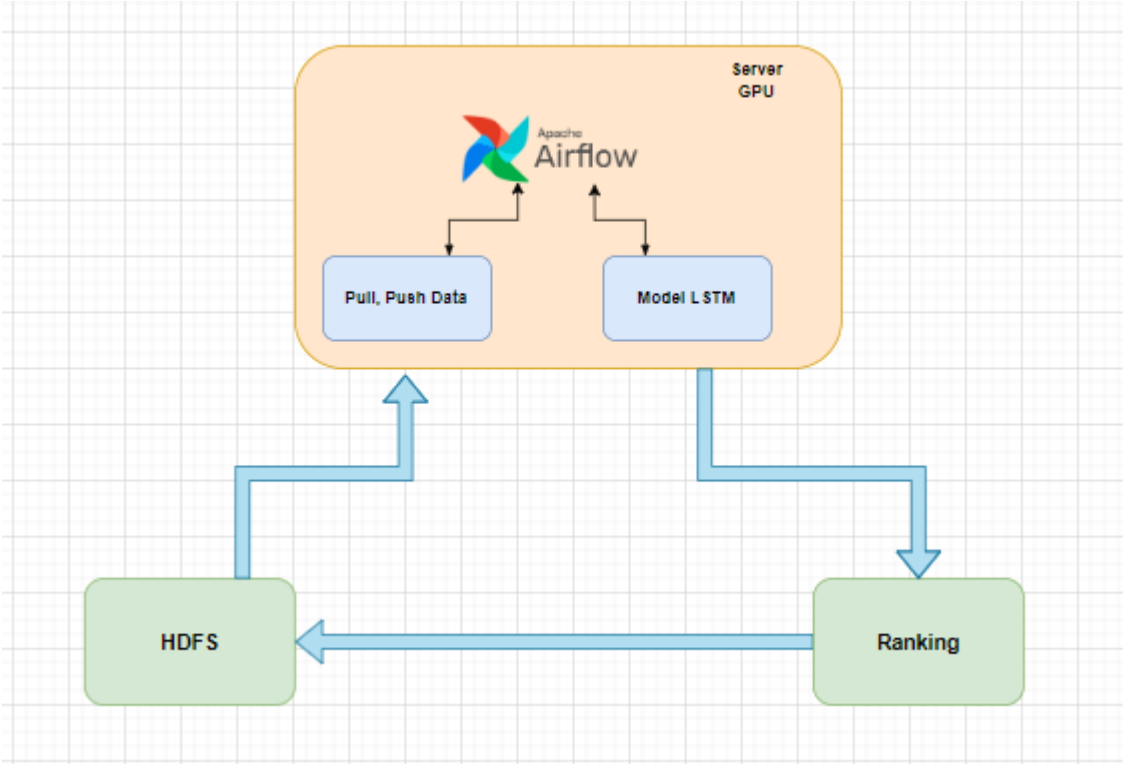
Dựng webserver cho thư viện MLFLow và testing trên một số thư viện mà team đang dùng như sklearn và pytorch. Hiện tại phân model tracking experiment đang setup xong và thu được một số kết quả như hình dưới đây:





Về phân model registry thì đang gặp một số lỗi về loading model bằng ID và state của model.

c) Xây dựng luồng tích hợp model Time Series cho các dịch vụ merchant



Pipeline bao gồm 3 thành phần chính:

- **HDFS**: Lưu trữ các bảng dữ liệu

- **Server GPU**:
  - **Pull, push data**: Module kéo dữ liệu từ HDFS về DGX và push dữ liệu từ DGX lên HDFS. 7 ngày sẽ pull dữ liệu về 1 lần để infer cho tuần sau
  - **Model LSTM**: Model core dự đoán nhu cầu người dùng có hay không trong 7 ngày cho các merchant:
    - Game
    - Vietlott
    - Education
    - Ecommerce
    - Travel
    - BaoHiem
    - Credit
    - Invest
- **Ranking**
  Model ranking các merchant recommend cho người dùng

Về tiến độ công việc hiện tại thì về cơ bản pipeline đã xây dựng xong các bước như:

- Sử dụng Airflow để lập lịch pull data từ HDFS về DGX để infer kết quả dự đoán và push data từ DGX lên HDFS
- Setting scheduler để ranking kết quả

Kết quả của model pipeline hiện tại:

- Prediction

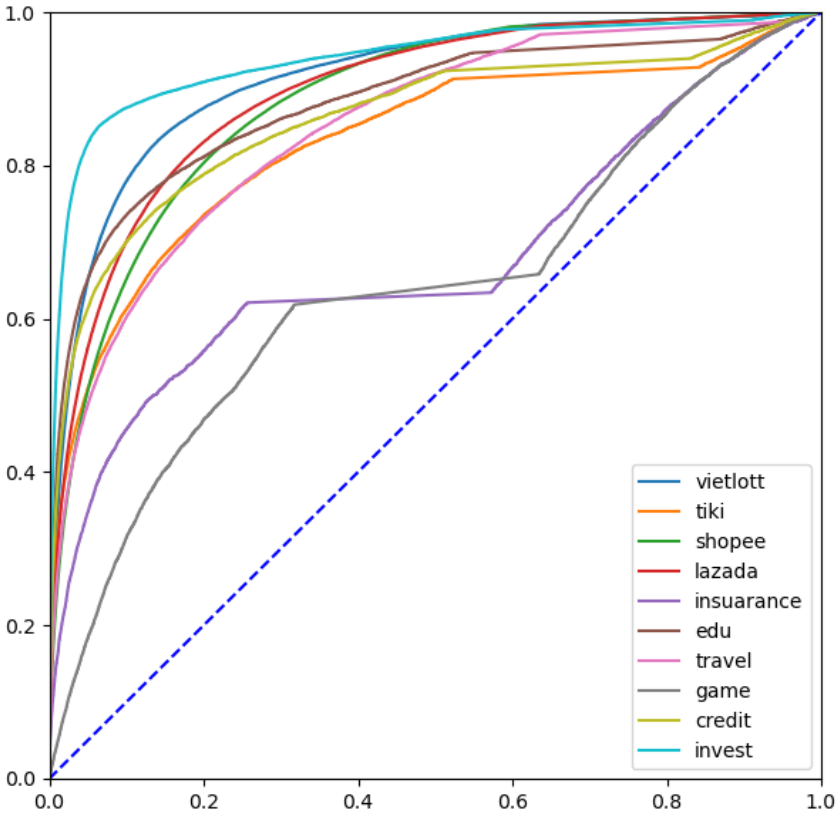
	Precision 0	Precision 1	Recall 0	Recall 1	AUC
Vietlott	0.991	0.211	0.900	0.776	0.903
Ecom	0.927	0.788	0.882	0.863	0.939
Edu	0.994	0.094	0.928	0.605	0.809
Travel	0.993	0.057	0.884	0.543	0.756
Game	0.951	0.513	0.811	0.828	0.895
Credit	0.996	0.08	0.948	0.576	0.795
Invest	0.995	0.21	0.944	0.77	0.897

- Ranking
  NDCG cho pipeline hiện tại là **0.73**

d) Đánh giá lại model Time series 3d với các nhãn yêu cầu bên anh Cường bao gồm các nhãn liên quan đến shopee, tiki, laza và sửa lại các label như bảo hiểm, game theo đúng như đang triển khai campain.

	P0	P1	R0	R1	AUC
Vietlott	0.989	0.249	0.861	0.829	0.913
Tiki	0.994	0.062	0.865	0.666	0.825
Shopee	0.937	0.523	0.796	0.808	0.885
Lazada	0.972	0.361	0.819	0.816	0.896
Insuarance	0.996	0.032	0.91	0.44	0.686
Edu	0.994	0.103	0.886	0.737	0.875

Travel	0.993	0.069	0.804	0.724	0.845
Game	0.993	0.024	0.876	0.348	0.635
Credit	0.995	0.083	0.877	0.73	0.866
Invest	0.995	0.271	0.932	0.854	0.937



e) Vận hành luồng triển khai cho production

- Vận hành luồng Ecom chạy hằng ngày cho việc recommend sản phẩm trên sàn thương mại điển tử trên app MyViettel
- Vận hành luồng triển khai dự đoán khách hàng tiềm nằng Vietlott cho bên chuyển đổi số
- Vận hành luồng mời mua thẻ game trên app MyViettel
- Fix luồng chuyển đổi schema từ default sang adp_new_db

1.2 Tham gia vào dự án recommend TV360.

a) Tìm hiểu luồng vận hành của recommend kênh của TV360

Tìm hiểu tài liệu giải pháp, source code cho luồng recommend kênh hiện tại của dự án TV360

b) Lên giải pháp giải quyết vấn đề popularity bias của recommend kênh

B1. Phân nhóm kênh truyền hình

Các kênh truyền hình được phát sóng sẽ mang những tính đặc trừng và nét riêng biệt của chương trình, từ đó góp phần mang lại những tính riêng biệt của channel đó. Tuy nhiên, các kênh lại có những tính chất khá giống nhau như: VTV1, Quộc Hội HD, Kênh Quốc Hội HD, Nhân Dân, HTV9 … là nhưng kênh có nội dùng liên quan đến Đảng nhà nước, thông tin thời sự trong và ngoài nước. Do đó cần một model có thể **tự động phân cụm các kênh** dựa trên nội dung các chương trình truyền hình của kênh đó mà **không phải label bằng tay** như version trước.

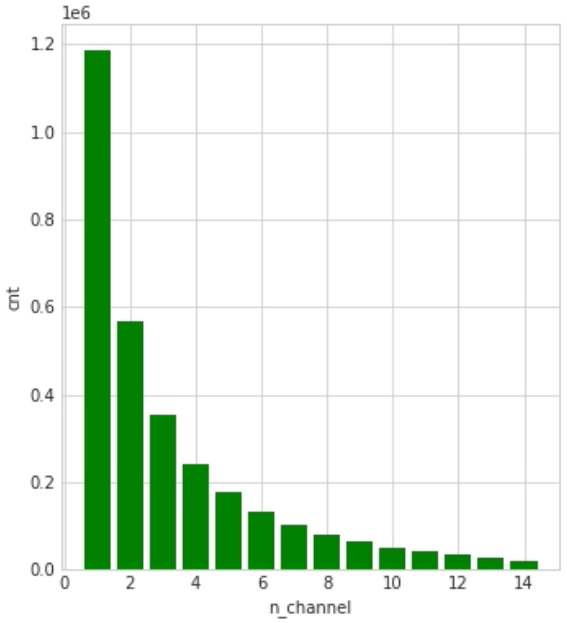
Cách thức xây dựng phân cụm:

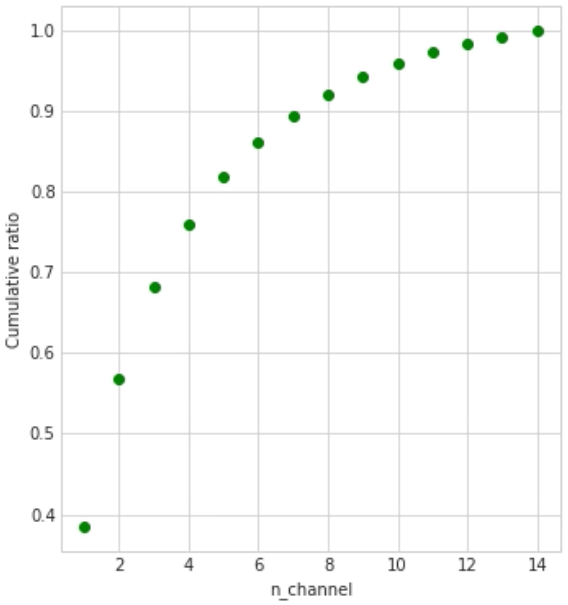
Ý tưởng chính: Từ các tên chương trình truyền hình được phát sóng trên kênh, chúng ta sẽ trích xuất các keyword (là các danh từ) để xây dựng đặc trưng cho kênh.

Sử dụng Model ViTokenizer trong Pyvi để trích xuất các danh từ trong câu.

Ví dụ: Bản tin thời sự quốc phòng ngày 24/5 [bản_tin, thời_sự, quốc_phòng]

Nhóm warm start:





Biểu đồ thống kê số lượng kênh và số lượng user xem trong 30 ngày gần nhất

Từ biểu đồ trên, ta có thể thấy rằng ~60% người dùng active(~3M user) chỉ xem <= 2 kênh, ~80% người dùng active chỉ xem <= 4 kênh trong 1 tháng gần nhất

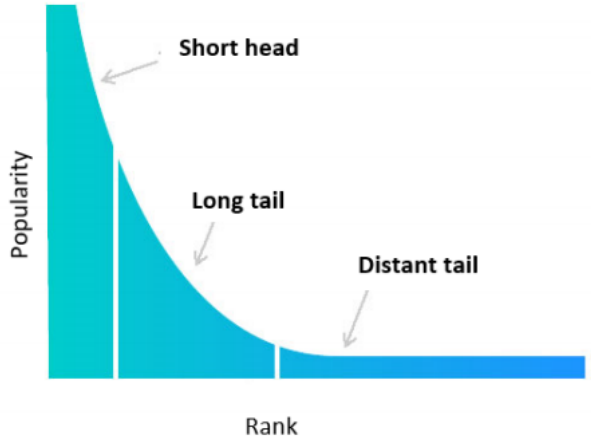
Trong N channel được recommend cho người dùng thì:

- 2 kênh đầu tiên là kênh mà người dùng xem nhiều nhất trong 3 ngày gần nhất Theo đánh giá độ chính xác với time range lần lượt là 3, 7, 10, 15, 20, 30 thì với N =3 thì cho độ chính xác **recall ~38%**
- N -2 kênh còn lại sẽ được filter theo các bước sau

Step 1: Retrieval các sản phẩm mà người dùng có thể quan tâm

Dựa vào log tương tác 30 ngày gần nhất, ta thu được 1 list các kênh mà người dùng quan tâm, kết hợp với kết quả từ model cluster item Ta thu được 1 danh sách các item mà người dùng có thể quan tâm

Step 2: Lọc top k sản phẩm



Sau khi thu được một danh sách các item từ bước 1, để tránh việc recommend quá nhiều vào các kênh nằm trong short head(các kênh được click nhiều, xem nhiều) và tránh vấn đề popular bias trong recommend thì sử dụng thêm phương pháp regularization để tăng khả năng cơ hội các item Long tail được recommend.

Chi tiết giải pháp:

Chúng ta sẽ mô hình hóa công thức đánh core như sau:

$$\min_{P,Q} acc(P,Q) + \lambda reg(P,Q)$$

Trong đó:

- acc(.) là objective function
- reg(.) là regularization term

- γ là hệ số điều chỉnh mức độ ảnh hưởng của regularization term

Mục đích chính của regularization là điều chỉnh mức độ cân bằng giữa các item short head và long tail. Chúng ta đĩnh nghĩa 1 danh sách item cân bằng khi đạt được tỷ lệ cân bằng 50/50 giữa short head và long tail item.
Trong giải pháp này:

- Acc(.)

  $L = \sum_{i \in L_u} score_i + \sum_{j \in L_u} score_j$

  Trong score là là giá trị normalize theo softmax từ giá trị:

$$video_{score} = \frac{watch\_duration}{num\_view} * 0,5^{1/30(d-d_{watch})}$$

- Regularization

$$ILBU(L_u) = \frac{1}{N(N-1)} \sum_{i,j \in L_u} d(i,j)$$

Trong đó: N là số lượng item trong danh sách recommned
d(i, j) =1 nếu i và j trong cùng 1 tập (short head hoặc long tail)

Một số hạn chế của model:

- Định nghĩa thế nào là short head channel, long tail channel. Hiện tại đang định nghĩa short head channel là top 10, 15% kênh được xem nhiều nhất trong tháng. Long tail channel là các kênh còn lại.

Kết quả:

Sau khi tích hợp giải pháp regularization bias trên thì đánh giá trên tập dữ liệu log từ ngày 25/5/ → 30/5 như sau:

- Tỷ lệ popular item trong list recommend giảm từ 32%(version luồng hiện tại) → 21% (sau khi tích hợp giải pháp)

- Recall@8 giảm từ **88% -> 69%.** Tuy nhiên nếu tính thêm trường hợp các sản phẩm cùng 1 cụm thì recall@8 giảm từ **88% → 86%**. Chứng tỏ giải pháp sau khi tích hợp regularization bias thì có sự giảm về Recall@8 tuy nhiên tăng sự đa dạng cho recommend của người dùng.

2. Các kiến thức học được

- EDA và xây dựng model
- Tìm hiểu luồng recommend của dự án Tv360
- Tìm hiệu về nghiệp vụ của dự án
- Tìm hiểu các paper liên quan đến Ranking và bias in recommend

3. Các công việc dự kiến cần làm quý tiếp theo

- Tìm hiểu thêm về luồng recommend của các block còn lại của dự án TV360
- Nghiên cứu thêm các paper SOTA hiện tại của recommend để áp dụng cho dự án TV360
- Tìm hiểu về các kiến thức liên quan đến model serving