

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
BỘ MÔN ĐỒ ÁN CÔNG NGHỆ THÔNG TIN



Cao Xuân Thắng – **19119222**
Trần Minh Sang – **19110278**

Đề tài:

TÌM HIỂU VỀ DỊCH VỤ RECOMMENDATION
CỦA AWS

BÁO CÁO ĐỒ ÁN

Giảng viên hướng dẫn:
TS. Huỳnh Xuân Phụng

KHOÁ 2019-2023

Tp. Hồ Chí Minh, tháng 12, năm 2021

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1 :.....MSSV 1:

Họ và tên Sinh viên 2 :.....MSSV 2:

Ngành: Công nghệ Thông tin

Tên đề tài:

Họ và tên Giáo viên hướng dẫn:

NHẬN XÉT

1. Về nội dung đề tài khối lượng thực hiện:

.....
.....
.....

2. Ưu điểm:

.....
.....
.....

3. Khuyết điểm

.....
.....
.....

4. Đánh giá loại :

5. Điểm :

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên hướng dẫn

(ký & ghi rõ họ tên)

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH ẢNH	2
PHẦN1: MỞ ĐẦU	4
PHẦN 2: NỘI DUNG.....	5
1. CHƯƠNG 1: Recommendation System	5
1.1. Khái quát về Recommendation System	5
1.2. Mục tiêu của Recommendation System.....	5
1.3. Một số ví dụ về hệ thống gợi ý (Recommendation System)	6
1.4. Phương pháp tiếp cận.....	7
2. CHƯƠNG 2: Dịch vụ Recommendation của AWS.....	11
2.1. Giới thiệu Recommendation của Amazon Personalize.....	11
2.2. Thuật toán triển khai	11
2.3. Đặc điểm nổi bật	13
2.4. Cách cài đặt và sử dụng Recommendation trên AWS	14
3. CHƯƠNG 3: CHƯƠNG TRÌNH MOVIE-RECOMMENDATION-SYSTEM.....	26
PHẦN 3: KẾT LUẬN.....	33
PHẦN 4: Danh mục tài liệu tham khảo	34

DANH MỤC HÌNH ẢNH

Hình 1 Create bucket	14
Hình 2 Bỏ chọn Block all public access.....	15
Hình 3 Sau khi tạo Bucket	16
Hình 4 Tải dữ liệu từ máy lên.....	17
Hình 5 Đã tạo xong dữ liệu Bucket.....	18
Hình 6 Dataset group	18
Hình 7 Sau khi tạo Dataset group	19
Hình 8 Import User-item	19
Hình 9 Tạo new schema	20
Hình 10 Nhập tên Dataset inport job name.....	21
Hình 11 Copy URL S3.....	21
Hình 12 Paste URL vào Data location.....	22
Hình 13 Kết quả sau khi tạo Schema.....	22
Hình 14 Vào Solution and recipes, chọn Create solution	23
Hình 15 Nhập đầy đủ thông tin và nhấn Next	23
Hình 16 Kết quả sau khi tạo solution	24
Hình 17 tại Campains chọn Create campaign	24
Hình 18 Nhập thông tin và chọn Create campaign	25
Hình 19: Cấu trúc chương trình	26
Hình 20: Dữ liệu danh sách movie từ đầu đến 36	27
Hình 21: Dữ liệu danh sách movie từ 17736 đến 17770	27
Hình 22: Dữ liệu danh sách data-full.....	28
Hình 23: MovieDescription.cs	28
Hình 24: Movie.cs.....	29
Hình 25: CsvMovieDescriptionMapping.cs	29
Hình 26: CsvMovieMapping.cs.....	29
Hình 27: LoadModel trong RecommendationSystem.cs	30

Hình 28: Tìm đánh giá của Id người dùng	30
Hình 29: Program.cs	31
Hình 30: Danh sách các bộ phim và điểm trung bình của bộ phim đó	31
Hình 31: Xem UserId 30878 đánh giá bộ phim với MovieId 28	32
Hình 32: Muốn xem tiếp đánh giá thì chọn Y và nếu không thì chọn n để kết thúc chương trình.....	32

PHẦN1: MỞ ĐẦU

Tính cấp thiết của đề tài

Với công nghệ 4.0 như hiện nay đã xuất hiện rất nhiều các nền tảng web được sử dụng. Chính với sự bùng nổ về công nghệ đó ta không thể không nhắc đến Amazon Web Services (AWS) là một trang web là một nền tảng xây dựng các trang web, phần mềm, ... mà chúng ta hay sử dụng. Trên thế giới, các công ty lớn hầu như đều sử dụng AWS để phát triển trang web hay phần mềm để giúp cho công ty họ phát triển. Bên trong AWS ta không thể bỏ qua dịch vụ Recommendation đã giúp cho nhiều doanh nghiệp trở nên phát triển hơn. Nhờ vào dịch vụ này mà các doanh nghiệp có thể quảng bá khắp thế giới các mặt hàng, sản phẩm mà doanh nghiệp đó đang triển khai hay sản xuất giúp người dùng quan tâm, tin tưởng đến nó.

Mục tiêu của đề tài

Đề tài ứng dụng dịch vụ Recommendation của AWS đề ra các mục tiêu là:

- + Nghiên cứu và tìm hiểu lý thuyết chung về AWS
- + Ứng dụng thuật toán Recommendation để cài đặt lại

Đối tượng phạm vi nghiên cứu

Nhóm chúng em nghiên cứu về thuật toán Recommendation System gợi ý ra danh sách các bộ phim đã xem qua và được đánh giá cao để gợi ý cho người dùng có thể quan tâm đến bộ phim đó.

Phương pháp nghiên cứu

Về mặt lý thuyết: chúng em nghiên cứu các tài liệu liên quan đến Recommendation trên AWS, hiểu được cách thức hoạt động của nó.

Về mặt thực nghiệm: tìm hiểu cách thức hoạt động của Recommendation mà từ đó viết ra một phần mềm hoạt động tương tự với thuật toán Recommendation. Và hiểu cách hoạt động của nó trên AWS để từ đó ta có thể xây dựng trang web hay phần mềm.

PHẦN 2: NỘI DUNG

1. CHƯƠNG 1: Recommendation System

1.1. Khái quát về Recommendation System

Với sự phát triển của công nghệ 4.0 và sự phát triển nhanh chóng của thương mại điện tử cùng với sự tương tác trực tiếp online của mọi người hiện nay và trong tương lai gần, hệ thống gợi ý (Recommendation systems) đã được phát triển và ứng dụng vào nhiều trang web hay dịch vụ mà chúng ta tương tác hằng ngày. Hệ thống gợi ý là một lớp con của hệ thống lọc thông tin mà dự đoán trước cho người sử dụng cần tìm dữ liệu gì. Được sử dụng rộng rãi trong nhiều lĩnh vực trong cuộc sống và được sử dụng phổ biến trong nhiều công ty thương mại điện tử hiện nay. Ý tưởng cho hệ thống gợi ý này là ta sử dụng các dữ liệu thu nhập dữ liệu từ người dùng hay lịch sử người dùng đã tìm kiếm trước đó và dự đoán dữ liệu mà người dùng sẽ tìm kiếm, gợi ý cho người dùng những sản phẩm, tính năng hay dịch vụ mà người dùng cần đến. Đối tượng mà hệ thống gợi ý hướng đến là user, còn sản phẩm hay dữ liệu hệ thống đưa ra gợi ý là item. Qua đó, chúng ta thấy được hệ thống gợi ý dựa trên tương tác trong quá khứ giữa user và item để dự đoán cho người dùng trong tương lai.

1.2. Mục tiêu của Recommendation System

Để giúp cho người dùng có thể tìm kiếm nhanh chóng, phù hợp với người dùng cũng như trong kinh doanh giúp cho tăng doanh thu và đem lại lợi nhuận cho mặt hàng đó, đem hướng chú ý của user về những item đó mà hệ thống gợi ý đã đưa ra một số mục tiêu và tiêu chí kỹ thuật:

- + Relevance (Tính phù hợp): Gợi ý phải phù hợp, thích hợp với người dùng, do người dùng thường sử dụng những gợi ý cảm thấy thú vị và phù hợp.
- + Novelty (Tính mới lạ): Hệ thống gợi ý hiệu quả nhất khi nó gợi ý những dữ liệu mà người dùng chưa từng thấy trong lịch sử tìm kiếm. Ví dụ, một sản phẩm trên Shopee khi mà ta tìm kiếm thường xuyên nó vẫn đưa ra những gợi ý đó sẽ làm cho người dùng không cảm thấy mới mẻ hay thích thú cho việc tìm kiếm và mua sử dụng, thay vào đó gợi ý thêm những sản phẩm liên quan thì sẽ giúp cho người dùng cảm thấy ấn tượng, có thể người dùng sẽ mua nó và điều đó giúp

tăng doanh thu sản phẩm đó cho shop bán hàng. Việc lặp đi lặp lại gợi ý làm giảm sự đa dạng trong việc tìm kiếm.

- + Serendipity (tính tình cờ): Dữ liệu được gợi ý là dữ liệu không ngờ tới đối với người dùng, có thể được coi là cách gợi ý có yếu tố may mắn. Serendipity khác với Novelty ở chỗ những gợi ý này thực sự gây ngạc nhiên với người dùng. Mục tiêu này được sử dụng trong trường hợp mà người dùng chỉ quen thuộc với một loại sản phẩm cụ thể, mà không biết đến những loại sản phẩm khác mà có thể gây ngạc nhiên và thích thú đối với những loại sản phẩm mới này. Serendipity tìm cách để mở rộng sự gợi ý đối với người dùng. Có thể coi đây là một kế hoạch gợi ý lâu dài, nhằm mang đến những sở thích mới, những mối quan tâm mới cho người dùng mà có thể giúp nâng cao được doanh số và lợi nhuận.
- + Mở rộng sự đa dạng trong việc gợi ý: Hệ thống gợi ý thường gợi ý một danh sách. Nếu chúng giống nhau, xảy ra trường hợp người dùng có thể không thích gợi ý nào cả. Bên cạnh đó, khi hệ thống gợi ý cố gắng đưa ra gợi ý nhiều kiểu hay loại khác nhau, khả năng mà người dùng thích ít nhất một gợi ý sản phẩm sẽ tăng lên một cách đáng kể, giảm được sự nhàm chán, không hài lòng cho người dùng cho việc gợi ý lặp đi lặp lại quá giống nhau.

1.3. Một số ví dụ về hệ thống gợi ý (Recommendation System)

GroupLens Recommender System

GroupLens được coi là hệ thống gợi ý tiên phong, nó được xây dựng để gợi ý cho Usenet news. Hệ thống sẽ thu thập những ratings của người đọc Usenet và sử dụng chúng để dự đoán liệu một người dùng có thích bài báo hay không trước khi họ đọc chúng. Trong hệ thống này, một số thuật toán "collaborative filtering" đã được sử dụng. Một số ý tưởng sau đó cũng được phát triển dựa theo hệ thống này, những ý tưởng đó áp dụng cho sách và phim, với bộ dữ liệu khá quen thuộc cho những người mới bắt đầu là BookLens và MovieLens.

Amazon Recommender System

Amazon cũng được coi là một trong những hệ thống gợi ý tiên phong, đặc biệt là trong thương mại và hệ thống gợi ý đã được sử dụng rất thành công ở Amazon. Hệ thống này sử dụng đánh giá của người dùng, hành vi của người dùng và lịch sử hoạt động của người dùng để làm dữ liệu dự đoán. Ratings ở Amazon có thang điểm từ 1 đến 5, còn lịch sử mua hàng và lịch sử hoạt động có thể dễ dàng được thu thập bằng khi mà user thực hiện đăng nhập trên trang web của Amazon. Những gợi ý cho người dùng được hiển thị trên trang chủ của Amazon, ngay khi người dùng truy cập vào trang web. Trong nhiều trường hợp, những items được gợi ý cũng được giải thích, ví dụ như trong mối quan hệ của sản phẩm được gợi ý với những sản phẩm mà user đã mua trước đó.

Facebook Friend Recommendations

Facebook thường gợi ý cho chúng ta kết bạn với những người mà chúng ta có thể quen, nhằm mục đích tăng số lượng kết nối trên mạng xã hội. Kiểu gợi ý này với mục tiêu có chút khác biệt so với hệ thống gợi ý sản phẩm. Trong khi hệ thống gợi ý sản phẩm trực tiếp tạo ra lợi nhuận cho người cung cấp dịch vụ bằng cách tăng doanh số sản phẩm bán được, thì trong mạng xã hội, việc gợi ý kết bạn nhằm tăng số lượng kết nối trên mạng xã hội và từ đó nâng cao trải nghiệm của người dùng. Cùng với việc tăng được số lượng kết nối, lợi nhuận thu được từ việc quảng cáo cũng sẽ tăng lên.

1.4. Phương pháp tiếp cận

Các hệ thống khuyến nghị/ gợi ý (Recommendation System/ Recommender) được xây dựng dựa trên 3 hướng tiếp cận đó là Collaborative Filtering (CF), Content-Based Filtering và Hybrid Recommendation Systems.

Collaborative Filtering

Collaborative Filtering (Lọc cộng tác) dựa trên item đã tìm kiếm trong quá khứ sẽ tìm kiếm trong tương lai và họ sẽ thích những item tương tự. Hệ thống chỉ tạo các đề xuất bằng cách sử dụng thông tin về cấu hình xếp hạng cho những người dùng hoặc mặt hàng khác nhau. Bằng cách định vị người dùng/ item ngang hàng có lịch sử xếp hạng tương tự với người dùng hoặc mặt hàng hiện

tại, họ tạo các đề xuất bằng cách sử dụng vùng lân cận này. Các phương pháp lọc cộng tác được phân loại là dựa trên bộ nhớ và dựa trên mô hình.

Khi xây dựng một mô hình từ hành vi của người dùng, người ta thường phân biệt giữa các hình thức thu thập dữ liệu rõ ràng và ngầm định.

Phương pháp lọc cộng tác này gặp ba vấn đề khó giải quyết: cold start, scalability và sparsity.

Bắt đầu lạnh (Cold start): Đối với người dùng hoặc mặt hàng mới, không có đủ dữ liệu để đưa ra các đề xuất chính xác.

Khả năng mở rộng (Scalability): Trong nhiều môi trường mà các hệ thống này đưa ra đề xuất, có hàng triệu người dùng và sản phẩm. Do đó, một lượng lớn sức mạnh tính toán thường là cần thiết để tính toán các khuyến nghị.

Độ thưa thớt (Sparsity): Số lượng item được tìm kiếm trên các trang thương mại điện tử lớn là vô cùng lớn. Những người dùng tích cực nhất sẽ chỉ đánh giá một tập hợp con nhỏ của cơ sở dữ liệu tổng thể. Vì vậy, ngay cả những mặt hàng phổ biến nhất cũng có rất ít xếp hạng.

Content-Based Filtering

Content-Based Filtering (Phương pháp lọc dựa trên nội dung) dựa trên mô tả của mặt hàng và hồ sơ về sở thích của người dùng. Phương pháp này phù hợp nhất với các trường hợp có dữ liệu đã biết về một mục (tên, vị trí, mô tả, v.v.), nhưng trong user không có dữ liệu.

Các từ khóa được sử dụng mô tả các mặt hàng và hồ sơ người dùng xây dựng chỉ ra loại mặt hàng mà người dùng thích. Có thể nói là đề xuất các mục tương tự như những mục mà người dùng đã thích trong quá khứ hoặc đang kiểm tra ở hiện tại.

Tạo hồ sơ người dùng, hệ thống tập trung nhất vào hai loại thông tin, đó là: Một mô hình sở thích của người dùng và lịch sử tương tác của người dùng với hệ thống giới thiệu.

Cơ bản, phương pháp này sử dụng một hồ sơ item (một tập hợp các thuộc tính và tính năng rời rạc) đặc trưng cho item trong hệ thống. Hệ thống tạo hồ sơ

dựa trên nội dung của người dùng dựa trên vectơ trọng số của các tính năng của mặt hàng.

Hệ thống gợi ý dựa trên nội dung cũng có thể bao gồm hệ thống tư vấn dựa trên ý kiến. Trong một số trường hợp, người dùng được phép đánh giá hay phản hồi về item. Các văn bản đánh giá do người dùng tạo là dữ liệu ngầm cho hệ thống giới thiệu. Các phương pháp tiếp cận phổ biến của hệ thống tư vấn dựa trên ý kiến sử dụng các kỹ thuật khác nhau bao gồm khai thác văn bản, truy xuất thông tin, phân tích tình cảm và học sâu.

Hybrid Recommendation Systems

Hầu hết các hệ thống gợi ý hiện nay sử dụng phương pháp kết hợp, kết hợp lọc cộng tác, lọc dựa trên nội dung và các phương pháp tiếp cận khác. Phương pháp tiếp cận kết hợp thực hiện bằng cách đưa ra các dự đoán dựa trên nội dung và dựa trên cộng tác một cách riêng biệt và sau đó kết hợp chúng; hay có thể thực hiện bằng cách thêm các khả năng dựa trên nội dung vào cách tiếp cận dựa trên cộng tác (và ngược lại); hoặc bằng cách thống nhất các phương pháp tiếp cận thành một mô hình. Các phương pháp này có thể được sử dụng khắc phục một số vấn đề phổ biến trong hệ thống khuyến nghị như khởi động nguội, thưa thớt.

Một số kỹ thuật lai bao gồm:

Weighted (Có trọng số): Kết hợp số điểm của các thành phần đề xuất khác nhau.

Switching (Chuyển đổi): Lựa chọn các thành phần đề xuất và áp dụng thành phần đã chọn.

Mixed (Hỗn hợp): Các đề xuất từ những người giới thiệu khác nhau được trình bày cùng nhau để đưa ra đề xuất.

Feature Combination (Kết hợp tính năng): Các tính năng bắt nguồn từ các nguồn kiến thức khác nhau được kết hợp với nhau và đưa ra cho một thuật toán đề xuất duy nhất.

Feature Augmentation (Tăng cường tính năng): Tính toán một tính năng hoặc tập hợp các tính năng, sau đó là một phần của đầu vào cho kỹ thuật tiếp theo.

Cascade (Xếp tầng): Các đề xuất được ưu tiên nghiêm ngặt, với những đề xuất có mức độ ưu tiên thấp hơn sẽ phá vỡ mối quan hệ trong việc chấm điểm những đề xuất cao hơn.

Meta-level (Siêu cấp): Một kỹ thuật đề xuất được áp dụng và tạo ra một số loại mô hình, sau đó là đầu vào được sử dụng bởi kỹ thuật tiếp theo.

Tóm lại Collaborative Filtering là hướng tiếp cận dựa vào dữ liệu trong quá khứ người sử dụng trong khi Content-Based Filtering lại chỉ dựa vào các đặc điểm / đặc trưng của sản phẩm. Trong khi Hybrid Recommendation Systems lại dựa vào đồng thời cả dữ liệu trong quá khứ người dùng và cả các đặc điểm của sản phẩm. Ví dụ hệ thống gợi ý của Techurls được sử dụng để gợi ý các bài báo về công nghệ liên quan mà người dùng nên xem qua.

2. CHƯƠNG 2: Dịch vụ Recommendation của AWS

2.1. Giới thiệu Recommendation của Amazon Personalize

Amazon Personalize cho phép các nhà phát triển xây dựng ứng dụng bằng chính công nghệ máy học (Machine Learning) mà Amazon.com dùng để đề xuất các nội dung cá nhân hóa theo thời gian thực, kể cả khi họ chưa có chuyên môn về Machine Learning.

Amazon Personalize giúp dễ dàng xây dựng ứng dụng mang lại trải nghiệm cá nhân hóa, bao gồm cả nội dung đề xuất cho sản phẩm cụ thể, xếp hạng lại sản phẩm cá nhân hóa và tiếp thị trực tiếp tùy chỉnh. Amazon Personalize là một dịch vụ máy học được quản lý toàn phần. Dịch vụ này không sử dụng các hệ thống đề xuất dựa trên quy tắc cố định cứng nhắc mà sẽ đào tạo, tinh chỉnh và triển khai các mô hình Machine Learning tùy chỉnh để cung cấp các đề xuất được tùy chỉnh chi tiết cho khách hàng trong các ngành như bán lẻ, truyền thông và giải trí.

Amazon Personalize cung cấp cơ sở hạ tầng cần thiết và quản lý toàn bộ quy trình Machine Learning, bao gồm xử lý dữ liệu, xác định tính năng, sử dụng các thuật toán tốt nhất, đồng thời đào tạo, tối ưu hóa và lưu trữ các mô hình. Bạn sẽ nhận được kết quả thông qua Giao diện lập trình ứng dụng (API) và chỉ phải trả tiền cho những gì mình sử dụng, không cần nộp phí tối thiểu hoặc cam kết trả trước. Toàn bộ dữ liệu được mã hóa để đảm bảo tính bảo mật và riêng tư. Dữ liệu chỉ được dùng để tạo các đề xuất cho người dùng của bạn.

Nhờ có Amazon Personalize, có thể thiết kế và khởi chạy một công cụ đề xuất cá nhân. Công cụ đề xuất cá nhân dựa vào các dữ liệu đã cung cấp và hoạt động mà người dùng thực hiện thường xuyên trong quá khứ mà tạo thành một danh sách gợi ý (đề xuất) giúp cho người dùng thuận tiện hơn khi sử dụng và giúp hoàn thành công việc một cách nhanh chóng.

2.2. Thuật toán triển khai

Để triển khai dịch vụ Recommendation của AWS sau khi chuẩn bị dữ liệu đề xuất, ta nên chú ý đến một số thuật toán triển khai sau:

User-Personalization recipe

Công thức User-Personalization (aws-user-Personalization) được tối ưu hóa cho tất cả các trường hợp đề xuất được cá nhân hóa. Nó dự đoán các mục mà người dùng sẽ tương tác dựa trên các tập dữ liệu Interactions, Items, and User datasets. Khi đề xuất các mặt hàng, nó sử dụng tính năng tự động thăm dò Items.

Với tính năng khám phá tự động, Amazon Personalize tự động kiểm tra các đề xuất Items khác nhau, học hỏi từ cách người dùng tương tác với Items được đề xuất này và tăng cường đề xuất cho Items thúc đẩy mức độ tương tác và chuyển đổi tốt hơn. Điều này cải thiện khả năng khám phá mặt hàng và mức độ tương tác khi bạn có một danh mục thay đổi nhanh hoặc khi Items mới, chẳng hạn như các bài báo hoặc chương trình khuyến mãi, phù hợp hơn với người dùng khi mới.

Bạn có thể cân bằng số lượng để khám phá (nơi các mục có ít dữ liệu tương tác hơn hoặc mức độ liên quan được đề xuất thường xuyên hơn) với số lượng cần khai thác (nơi các đề xuất dựa trên những gì chúng tôi biết hoặc mức độ liên quan). Amazon Personalize tự động điều chỉnh các đề xuất trong tương lai dựa trên phản hồi ngầm của người dùng.

Công thức được xác định trước này có các thuộc tính sau:

Name – aws-user-personalization

Recipe Amazon Resource Name (ARN) – arn:aws:personalize:::recipe/aws-user-personalization

Algorithm ARN – arn:aws:personalize:::algorithm/aws-user-personalization

Popularity-Count recipe

Mức độ phổ biến-Số lượng đề xuất Items phổ biến nhất dựa trên tất cả dữ liệu hành vi người dùng của bạn. Các Items phổ biến nhất có nhiều tương tác nhất với những người dùng duy nhất. Công thức trả về các mặt hàng phổ biến giống nhau cho tất cả người dùng. Popularity-Count là một cơ sở tốt để so sánh với các công thức khác bằng cách sử dụng các chỉ số đánh giá mà Amazon Personalize tạo ra khi bạn tạo một phiên bản giải pháp.

Công thức được xác định trước này có các thuộc tính sau:

Name – aws-popularity-count

Recipe ARN – arn:aws:personalize:::recipe/aws-popularity-count

Algorithm ARN – arn:aws:personalize:::algorithm/aws-popularity-count

Feature transformation ARN – arn:aws:personalize:::feature-transformation/sims

Recipe type – USER_PERSONALIZATION

Popularity-Count không có siêu tham số được hiển thị.

Personalized-Ranking recipe

Personalized-Ranking recipe tạo ra thứ hạng được cá nhân hóa của các mặt hàng. Một Personalized-Ranking là một danh sách các mục khuyến cáo rằng đang tái xếp cho một người dùng cụ thể. Điều này rất hữu ích nếu bạn có một bộ sưu tập các mục được sắp xếp, chẳng hạn như kết quả tìm kiếm, khuyến mại hoặc danh sách được sắp xếp và bạn muốn cung cấp một personalized re-ranking cho từng người dùng của mình.

Để đào tạo một mô hình, Personalized-Ranking recipe sử dụng dữ liệu trong tập dữ liệu Tương tác của bạn và nếu bạn đã tạo chúng, tập dữ liệu Mục và tập dữ liệu Người dùng trong nhóm tập dữ liệu của bạn. Với Personalized-Ranking, tập dữ liệu Items của bạn có thể bao gồm Unstructured text metadata và tập dữ liệu Interactions của bạn có thể bao gồm Contextual metadata.

Công thức được xác định trước này có các thuộc tính sau:

Name – aws-personalized-ranking

Recipe Amazon Resource Name (ARN) – arn:aws:personalize:::recipe/aws-personalized-ranking

Algorithm ARN – arn:aws:personalize:::algorithm/aws-personalized-ranking

Feature transformation ARN – arn:aws:personalize:::feature-transformation/JSON-percentile-filtering

Recipe type – PERSONALIZED_RANKING

2.3. Đặc điểm nổi bật

Cung cấp đề xuất chất lượng cao, theo thời gian thực: dùng thật toán Machine Learning để tạo ra các đề xuất chất lượng cao hơn đáp ứng được yêu cầu của người dùng.

Dễ dàng triển khai các đề xuất cá nhân hoá trong vài ngày: triển khai đề xuất cá nhân hoá tùy chỉnh được Machine Learning hỗ trợ giúp ta nhanh chóng tạo một cách dễ dàng với vài cú nhấp chuột

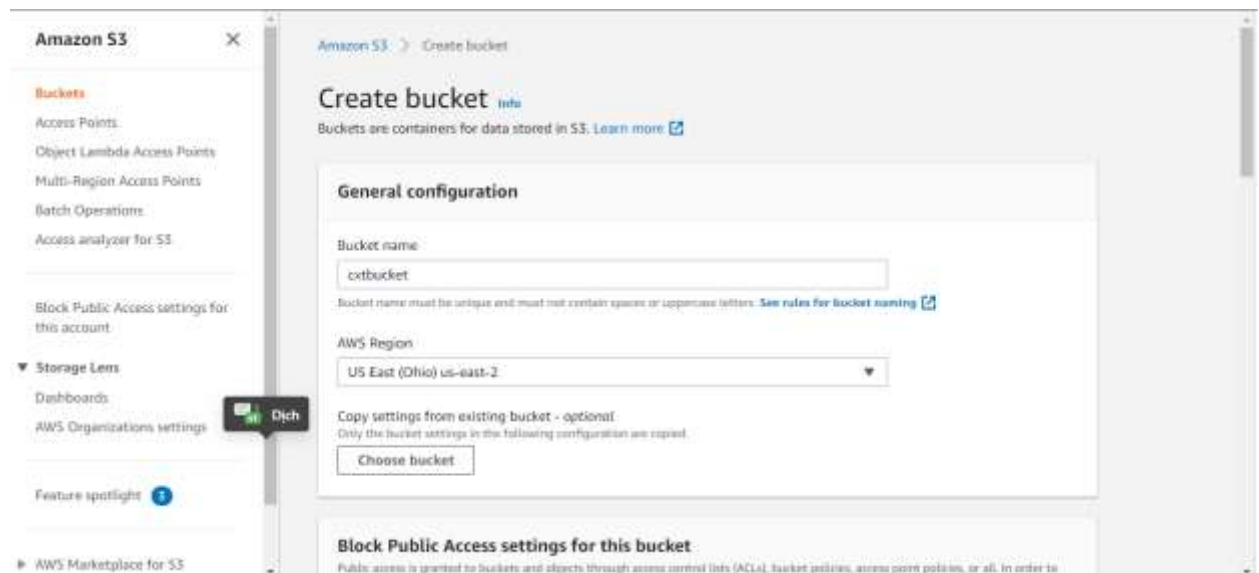
Bảo mật và quyền riêng tư về dữ liệu: toàn bộ dữ liệu được mã hoá, bảo mật, chỉ tạo đề xuất cho khách hàng của bạn, dữ liệu này không chia sẻ giữa khách hàng hay với Amazon.com.

Cá nhân hoá mọi điểm tiếp xúc trong quá trình của khách hàng: có thể thực hiện trên các kênh và thiết bị khác nhau mà không chi trả các chi phí cơ sở hạ tầng hoặc tài nguyên nào, cho phép sử dụng các đề xuất một cách linh hoạt trên thời gian thực dựa trên những yêu cầu của người dùng.

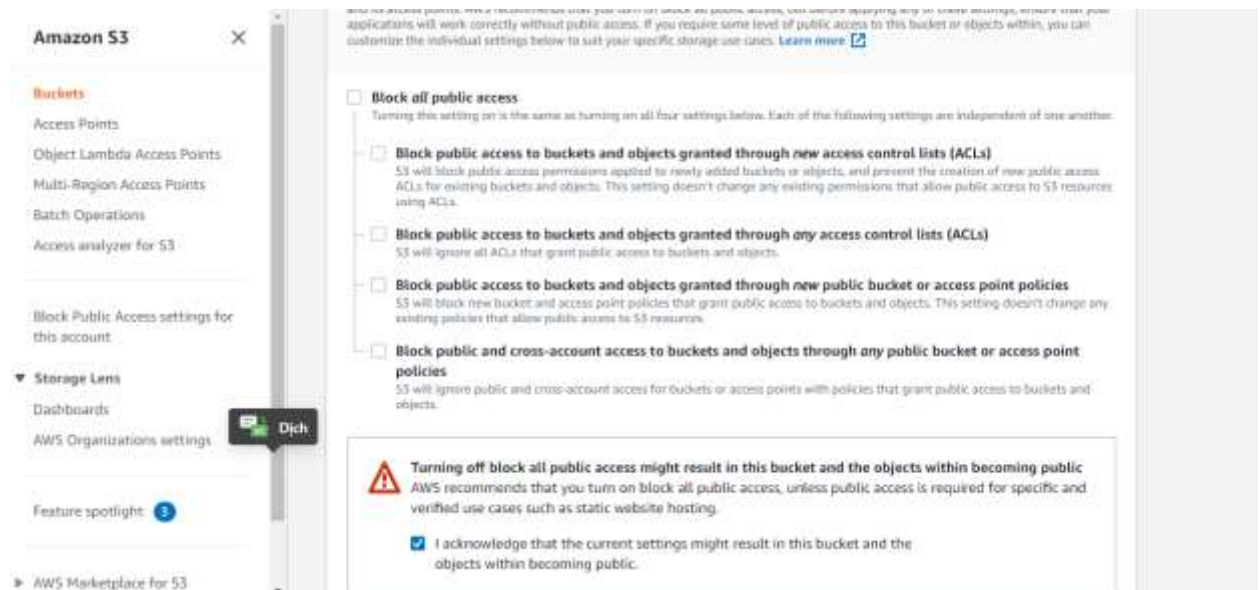
2.4. Cách cài đặt và sử dụng Recommendation trên AWS

Đầu tiên chúng ta phải tạo Buckets tại trang Amazon S3 của mình.

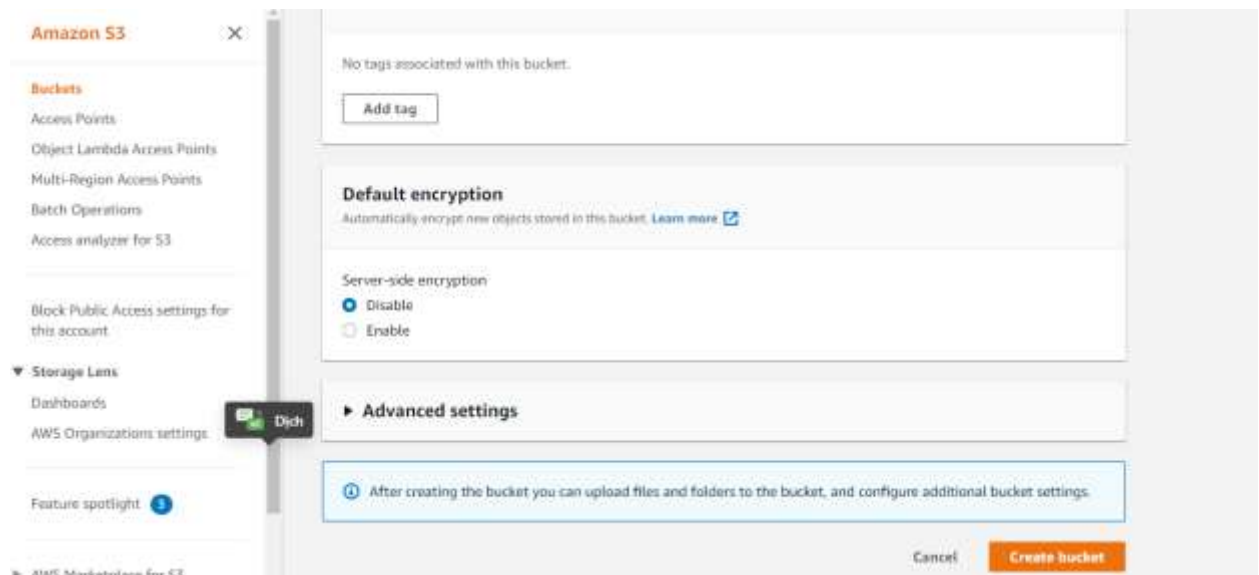
Ta bấm Create bucket -> Nhập Bucket name, Region -> Bỏ chọn Block all public access -> Phần Manage system permissions chọn Grant Amazon S3 Log Delivery group with access to this bucket -> Bấm Next -> Bấm Create bucket.

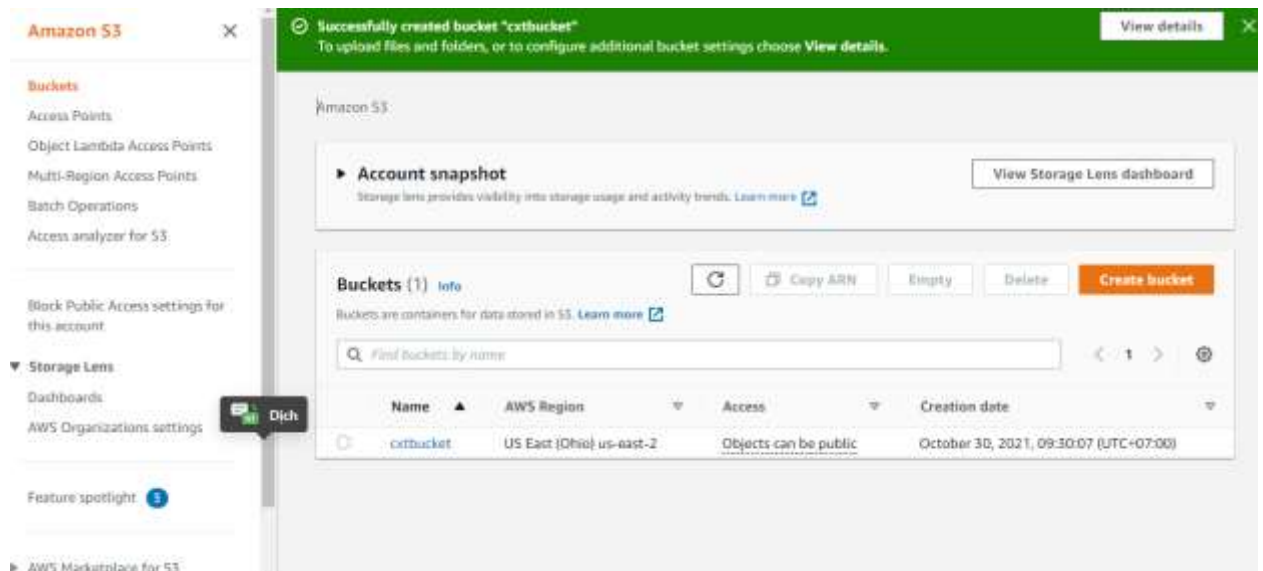


Hình 1 Create bucket



Hình 2 Bỏ chọn Block all public access





Hình 3 Sau khi tạo Bucket

Ta đã có Bucket. Chọn Bucket vừa tạo -> Tải dữ liệu phù hợp lên từ máy (.csv) -> Trong mục Permissions -> Ta vào Bucket Policy nhập vào phần Bucket policy editor đoạn code:

Bucket Policy json:

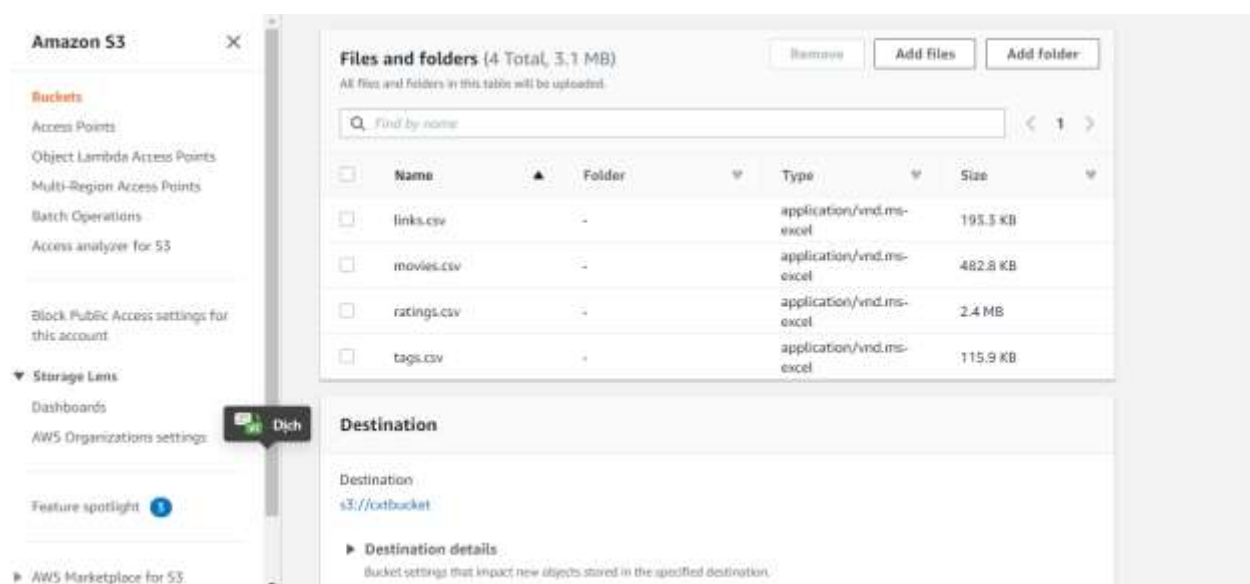
```
{
  "Version": "2012-10-17",
  "Id": "PersonalizeS3BucketAccessPolicy",
  "Statement": [
    {
      "Sid": "PersonalizeS3BucketAccessPolicy",
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],

```

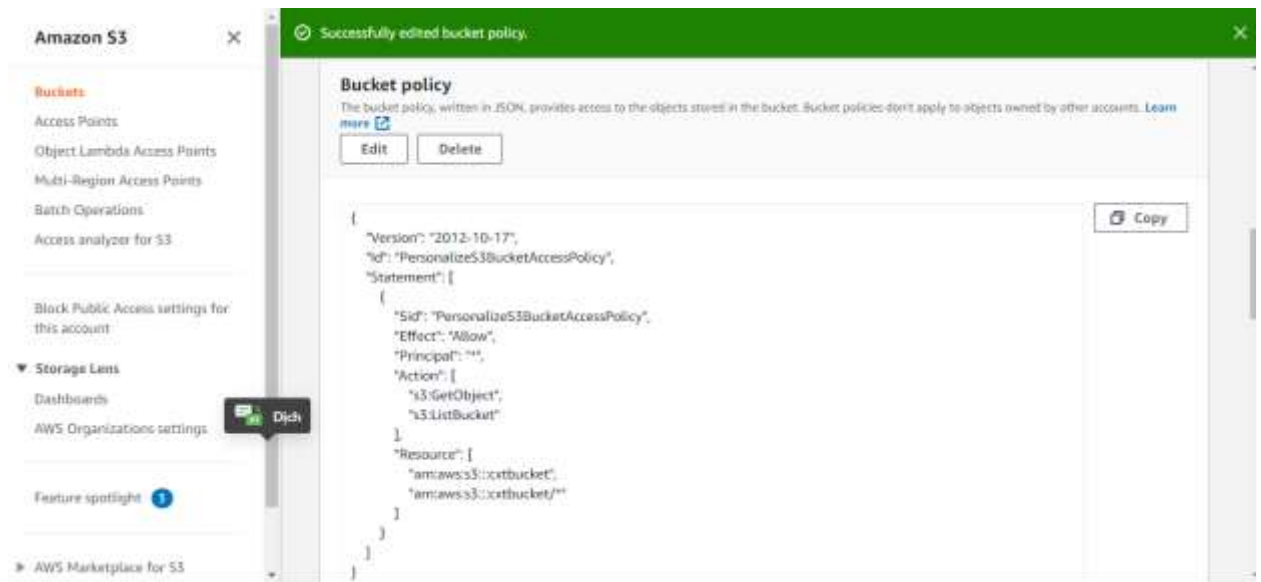
```

"Resource":
[
  "arn:aws:s3:::bucket-name",
  "arn:aws:s3:::bucket-name/*"
]
}
]
}

```

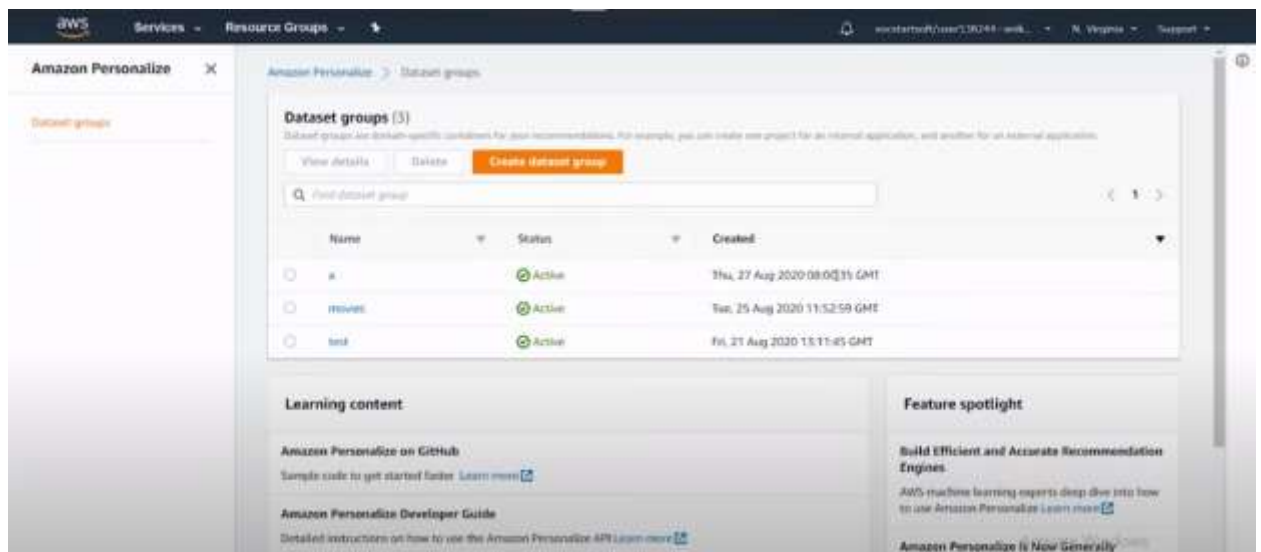


Hình 4 Tải dữ liệu từ máy lên

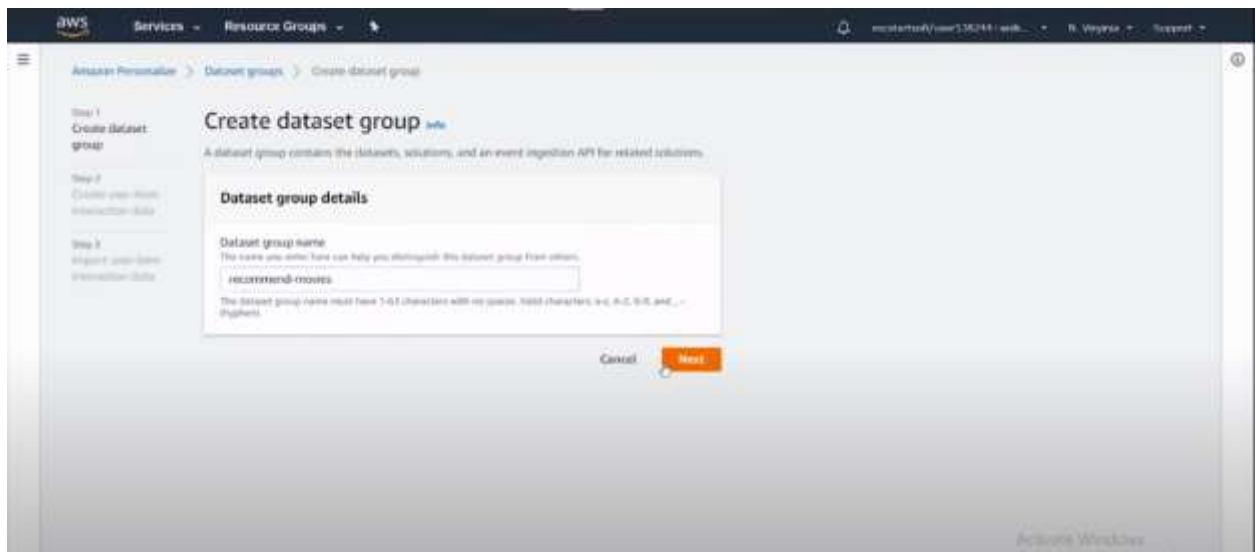


Hình 5 Đã tạo xong dữ liệu Bucket

Truy cập vào Amazon Personalize của mình -> Bấm vào Create dataset group -> Nhập tên cho Dataset group name -> Bấm Next.

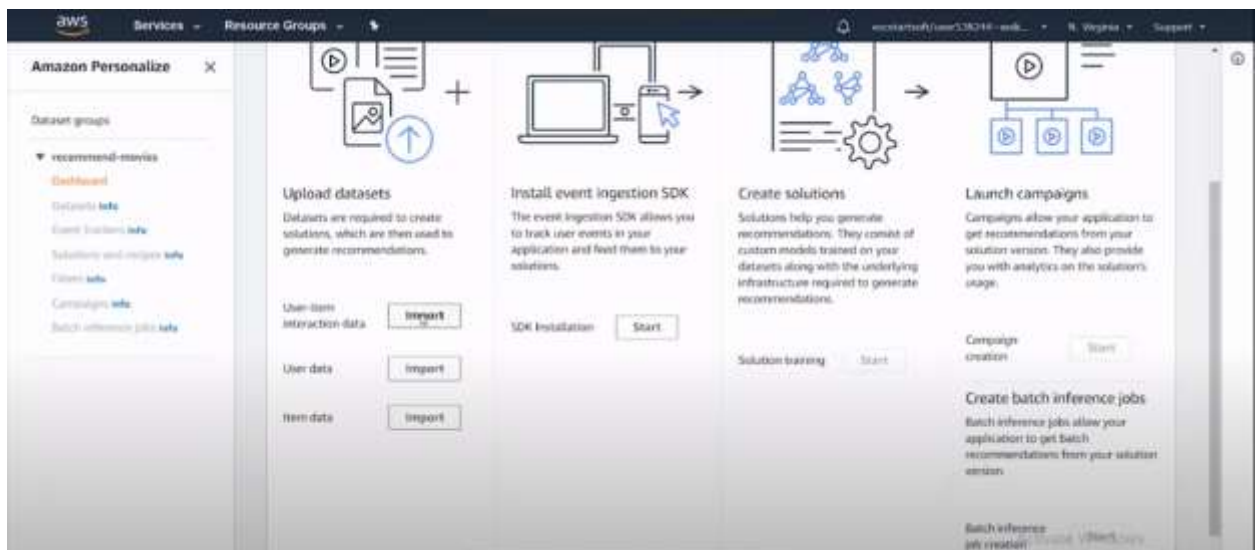


Hình 6 Dataset group



Hình 7 Sau khi tạo Dataset group

Quay lại trang Amazon Personalize của mình, bấm vào Dataset group vừa tạo -
>Trong mục Dashboard -> Tại phần User-item interaction data -> Bấm Import.



Hình 8 Import User-item

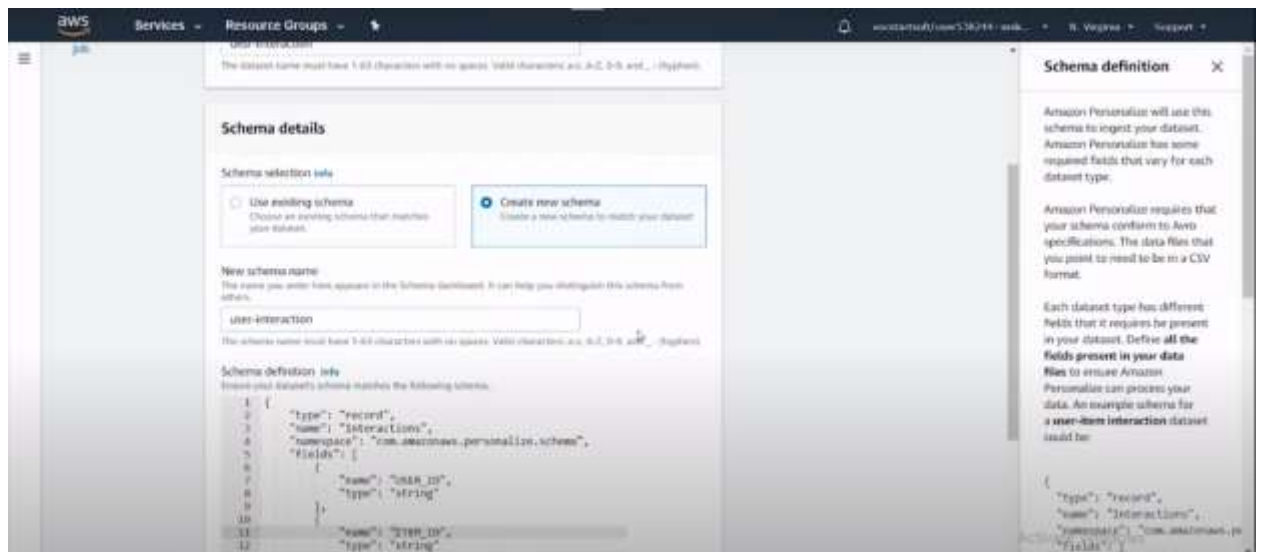
Nhập tên cho Dataset name -> Chọn vào Create new schema -> Nhập tên cho New schema name. Bởi vì data có dạng .csv nên bạn phải thay đổi Schema definition với định dạng:

```
{
  "type": "record",
  "name": "Interaction",
```

```

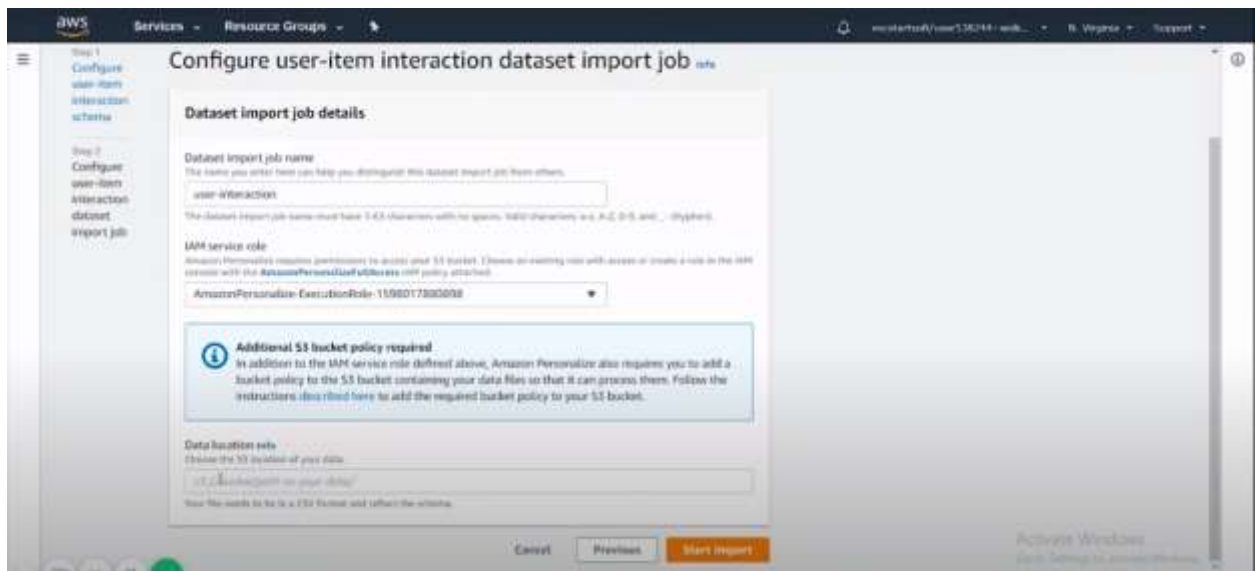
    "namespace": "com.amazonaws.personalize.schema",
    "fields":
    [
      {
        "name": "Ten_cot_1",
        "type": "Kieu_du_lieu_cot_1",
      },
      {
        "name": "Ten_cot_2",
        "type": "Kieu_du_lieu_cot_2",
      }
    ]
    "version": "1.0"
  }

```



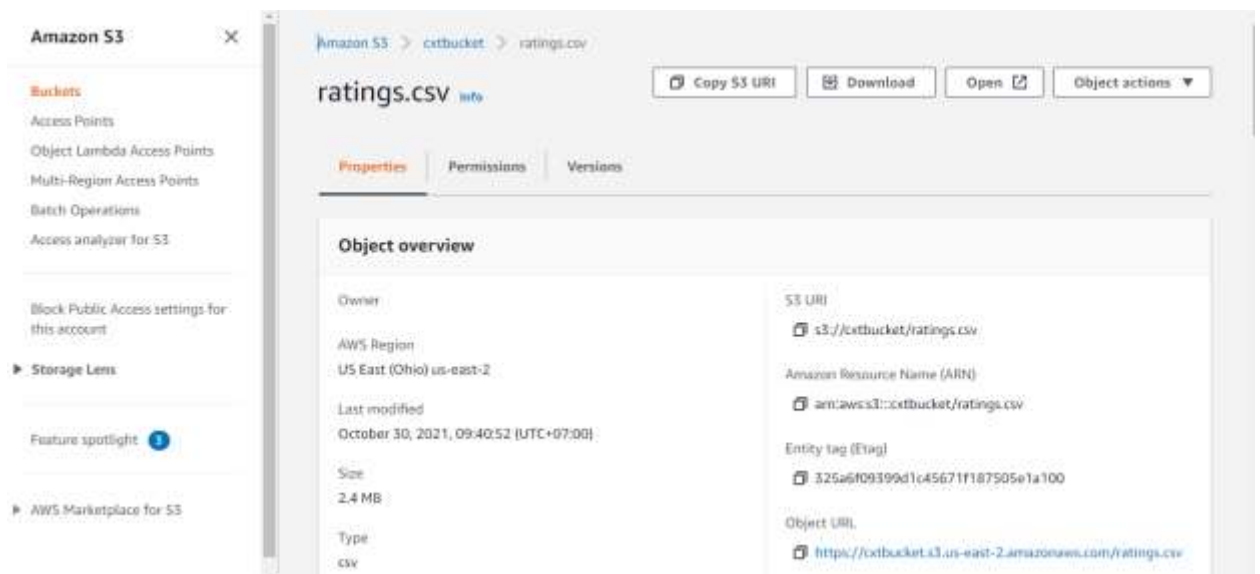
Hình 9 Tạo new schema

Sau khi hoàn thành ta bấm Next -> Nhập tên cho Dataset import job name.

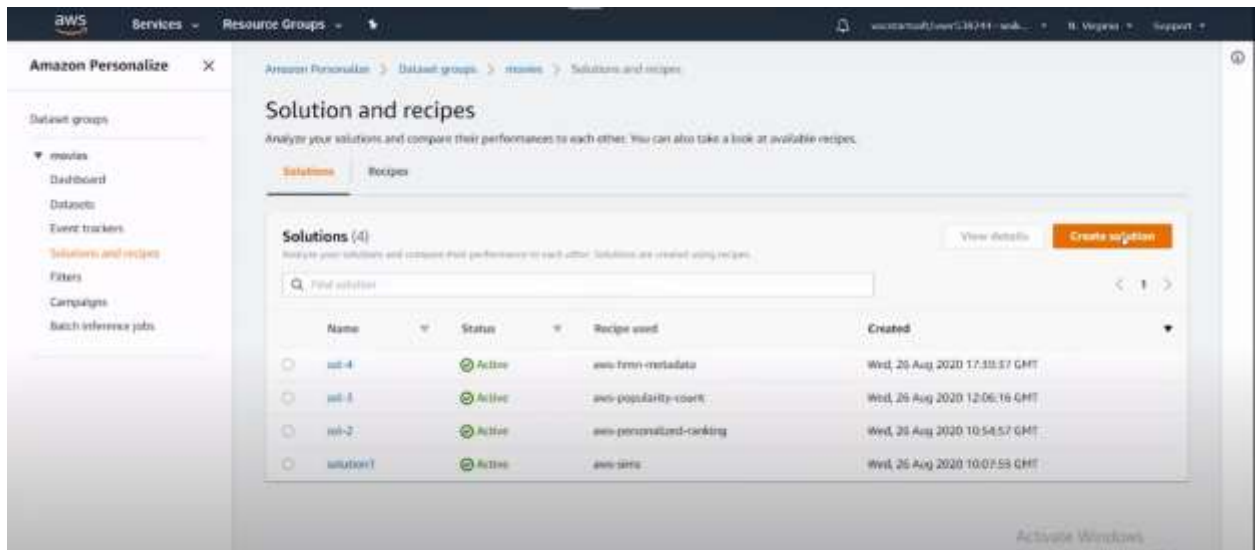


Hình 10 Nhập tên Dataset import job name

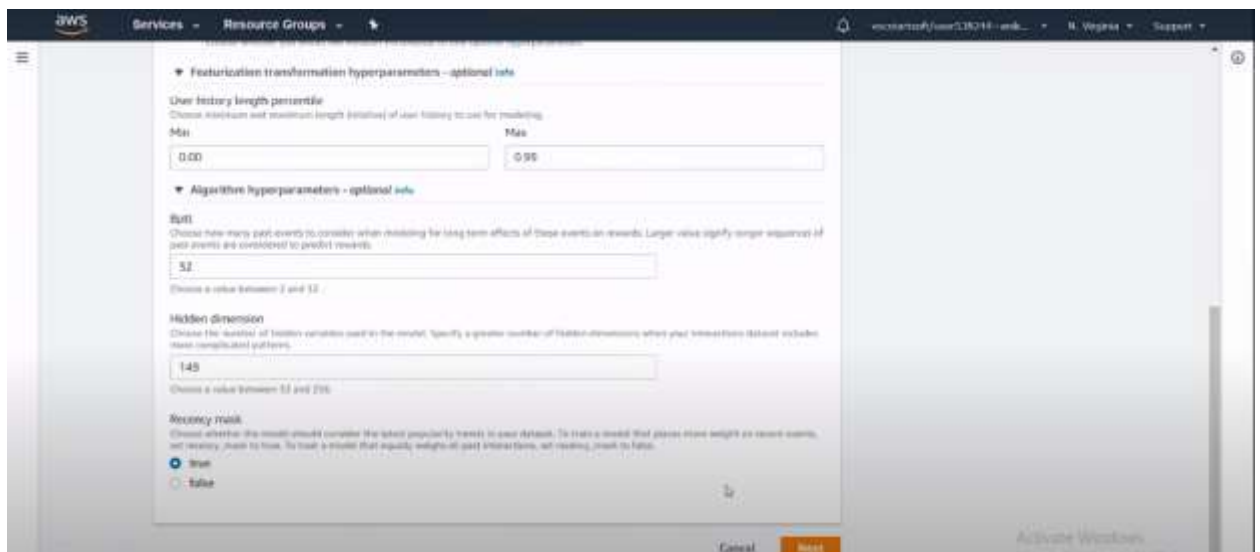
Copy URL S3 -> Paste đường dẫn vào Data location theo thiết kế dataset.



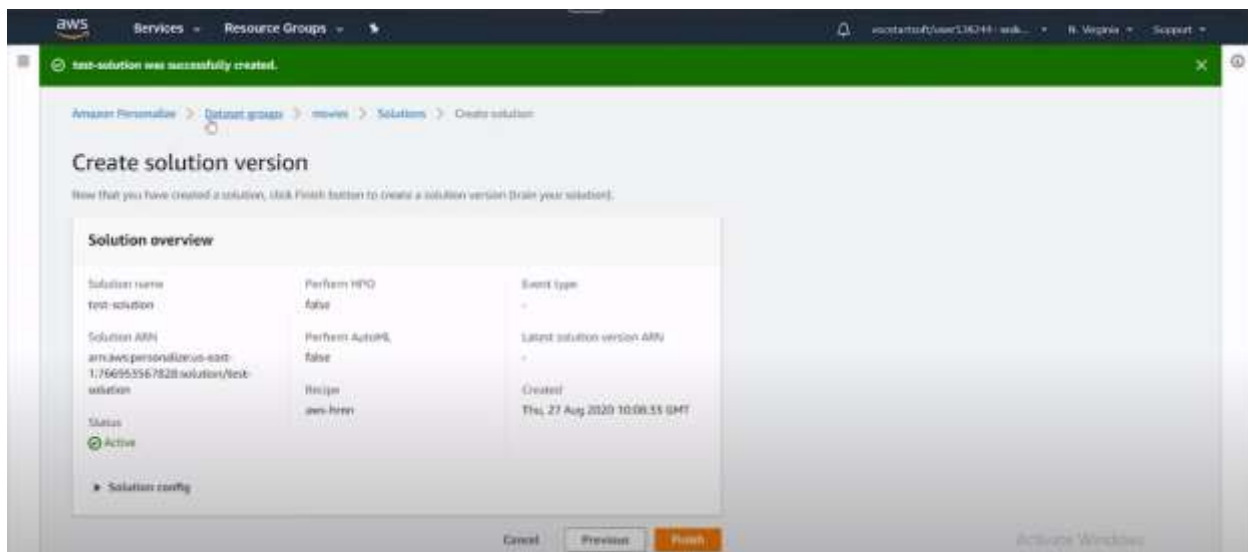
Hình 11 Copy URL S3



Hình 14 Vào Solution and recipes, chọn Create solution

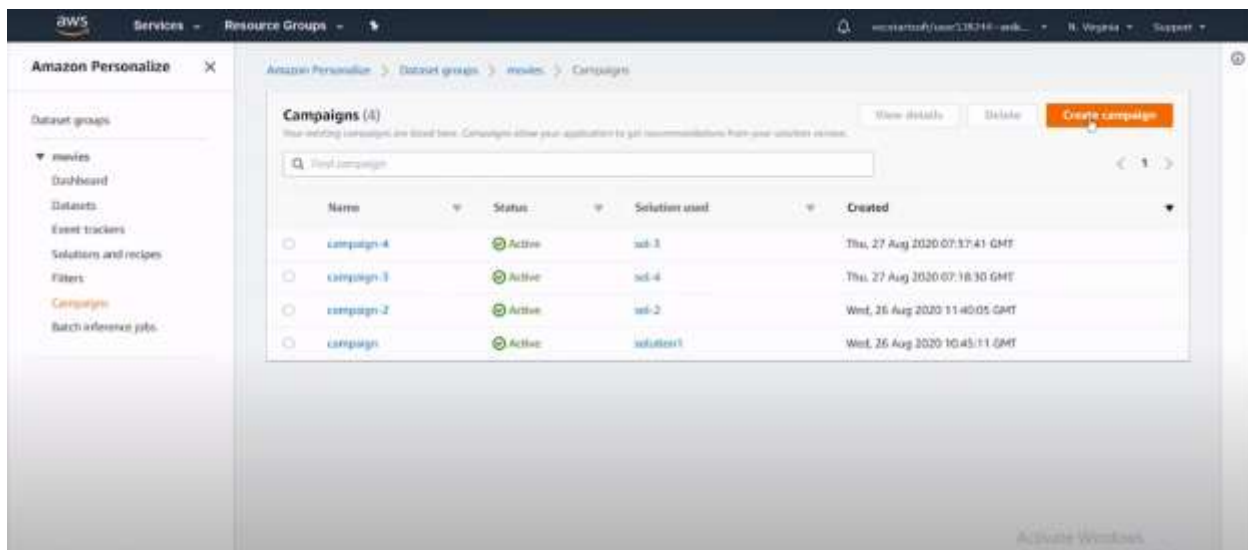


Hình 15 Nhập đầy đủ thông tin và nhấn Next



Hình 16 Kết quả sau khi tạo solution

Tại mục Campaigns -> Bấm vào Create campaign -> Nhập Campaign name, Solution, Solution ID -> Bấm Create campaign.



Hình 17 tại Campaigns chọn Create campaign

Amazon Personalize > Dataset groups > movies > Campaigns > Create new campaign

Create new campaign

Campaign details

Campaign name
The text you enter here appears in the Campaigns dashboard and detail page. It can help you distinguish this campaign from others.

test-campaign

The campaign name must have 1-45 characters with no spaces. Valid characters are: A-Z, 0-9, and _ (underscore).

Solution
The selected solution is used to generate the recommendations provided in your campaign.

sol-2

Solution version ID
The selected solution version is used to generate the recommendations provided in your campaign.

7x031540
Wed, 25 Aug 2021 11:21:46 GMT

Minimum provisioned transactions per second rate
The minimum amount of throughput in transactions per second (TPS) that is provisioned for this campaign.

3

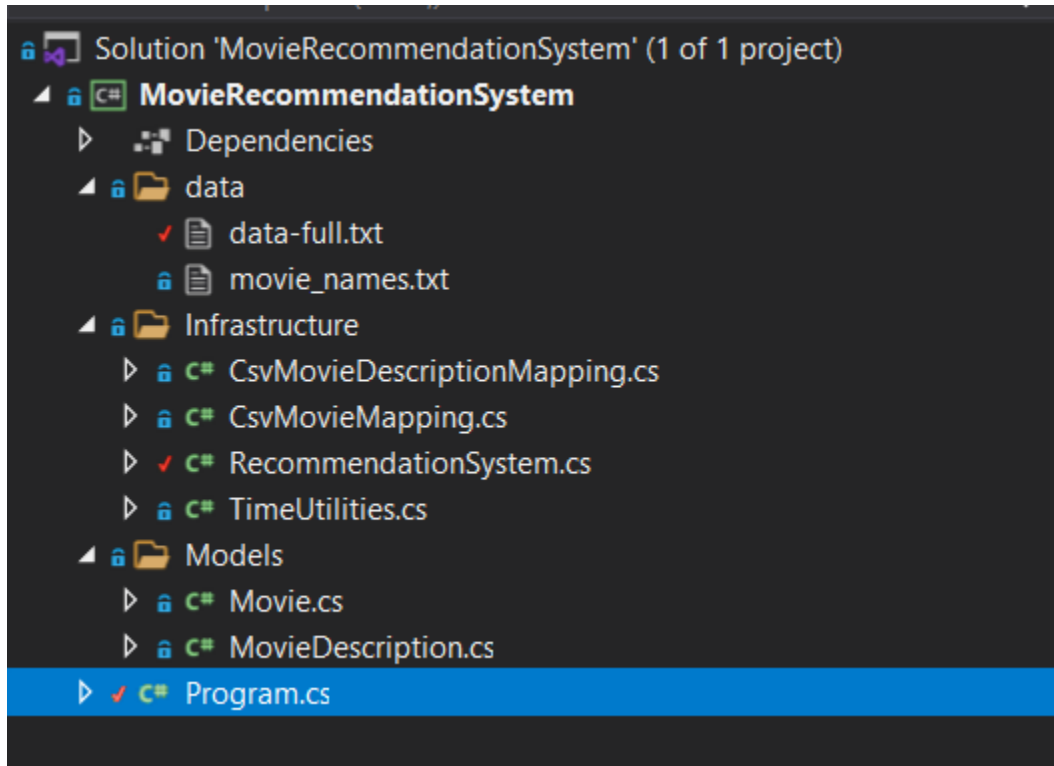
Enter a number from 0-500.

Cancel Create campaign

Hình 18 Nhập thông tin và chọn Create campaign

3. CHƯƠNG 3: CHƯƠNG TRÌNH MOVIE-RECOMMENDATION-SYSTEM

Cấu trúc chương trình (Visual Studio)



Hình 19: Cấu trúc chương trình

Danh sách data

Gồm 17770 bộ phim:

```

1 1,2003,Dinosaur Planet
2 2,2004,Isle of Man TT 2004 Review
3 3,1997,Character
4 4,1994,Paula Abdul's Get Up & Dance
5 5,2004,The Rise and Fall of ECW
6 6,1997,Sick
7 7,1992,8 Man
8 8,2004,What the #$! Do We Know!?
9 9,1991,Class of Nuke 'Em High 2
10 10,2001,Fighter
11 11,1999,Full Frame: Documentary Shorts
12 12,1947,My Favorite Brunette
13 13,2003,Lord of the Rings: The Return of the King: Extended Edition: Bonus Material
14 14,1982,Nature: Antarctica
15 15,1980,Neil Diamond: Greatest Hits Live
16 16,1996,Screamers
17 17,2005,7 Seconds
18 18,1994,Immortal Beloved
19 19,2000,By Dawn's Early Light
20 20,1972,Seeta Aur Geeta
21 21,2002,Strange Relations
22 22,2000,Chump Change
23 23,2001,Clifford: Clifford Saves the Day! / Clifford's Fluffiest Friend Cleo
24 24,1981,My Bloody Valentine
25 25,1997,Inspector Morse 31: Death Is Now My Neighbour
26 26,2004,Never Die Alone
27 27,1962,Sesame Street: Elmo's World: The Street We Live On
28 28,2002,Lilo and Stitch
29 29,2001,Boycott
30 30,2003,Something's Gotta Give
31 31,1999,Classic Albums: Meat Loaf: Bat Out of Hell
32 32,2004,ABC Primetime: Mel Gibson's The Passion of the Christ
33 33,2000,Aqua Teen Hunger Force: Vol. 1
34 34,2003,Ashtanga Yoga: Beginner's Practice with Nicki Doane
35 35,2000,Fergully 2: The Magical Rescue
36 36,1992,Lady Chatterley

```

Hình 20: Dữ liệu danh sách movie từ đầu đến 36

```

17736 17736,1999,Reptilian
17737 17737,1985,Frank Zappa: Does Humor Belong in Music?
17738 17738,2004,Henry Rollins: Live at Luna Park
17739 17739,1960,Dracula / The Strange Case of Dr. Jekyll and Mr. Hyde
17740 17740,1960,Shoot the Piano Player
17741 17741,2004,Ginger Snaps 2: Unleashed
17742 17742,1995,Catherine the Great
17743 17743,2003,Better Luck Tomorrow
17744 17744,2004,NASCAR: Tony Stewart Smoke
17745 17745,2002,Russell Simmons Presents Def Poetry: Season 1
17746 17746,1991,Godzilla & Mothra: Battle for Earth / Vs. King Ghidora
17747 17747,1991,Eric Clapton: 24 Nights
17748 17748,2005,Dog the Bounty Hunter: The Best of Season 1
17749 17749,1985,No End
17750 17750,2005,The Hee Haw Collection: Vol. 4
17751 17751,1993,Highlander: Season 2
17752 17752,2003,Out of Order
17753 17753,1997,Maslin Beach
17754 17754,1999,On the Ropes
17755 17755,2003,L/R: Licensed by Royalty
17756 17756,1935,The 39 Steps
17757 17757,2002,Ulysses 5. Grant: Warrior / President: American Experience
17758 17758,1979,Prophecy
17759 17759,1972,The Big Bird Cage
17760 17760,2004,Lightning Bug
17761 17761,2003,Levity
17762 17762,1997,Gattaca
17763 17763,1978,Interiors
17764 17764,1990,Shakespeare in Love
17765 17765,1969,Godzilla's Revenge
17766 17766,2002,Where the Wild Things Are and Other Maurice Sendak Stories
17767 17767,2004,Fidel Castro: American Experience
17768 17768,2000,Epoch
17769 17769,2003,The Company
17770 17770,2003,Allen Hunter

```

Hình 21: Dữ liệu danh sách movie từ 17736 đến 17770

Data-full.txt là danh sách ID bộ phim, ID người xem và đánh giá (rating) của họ

```
1 28,1302773,4.0
2 28,1990901,5.0
3 28,765331,3.0
4 28,1987434,4.0
5 28,2193455,4.0
6 28,1468812,4.0
7 28,41412,4.0
8 28,2400678,4.0
9 28,160077,4.0
10 28,098229,4.0
11 28,30878,3.0
12 28,577307,5.0
13 28,2615678,5.0
14 28,1276913,5.0
15 28,1663216,4.0
16 28,2196888,5.0
17 28,1744889,1.0
18 28,2569320,3.0
19 28,1174530,4.0
20 28,1644758,3.0
21 28,1753259,3.0
22 28,339448,4.0
23 28,1778851,5.0
24 28,1000399,4.0
25 28,542786,4.0
26 28,2639583,3.0
27 28,2433426,4.0
28 28,048367,5.0
29 28,488477,3.0
30 28,1765193,4.0
31 28,368495,5.0
32 28,305151,4.0
33 28,780936,4.0
34 28,557167,4.0
35 28,1296163,3.0
36 28,951709,5.0
```

Hình 22: Dữ liệu danh sách data-full

Code Models

```
public class MovieDescription
{
    3 references
    public int MovieId { get; set; }

    1 reference
    public int YearOfRelease { get; set; }

    3 references
    public string Title { get; set; }
}
```

Hình 23: MovieDescription.cs

```

3 references
public class Movie
{
    2 references
    public int MovieId { get; set; }

    2 references
    public int UserId { get; set; }

    2 references
    public double Rating { get; set; }
}

```

Hình 24: Movie.cs

CsvMovieDescriptionMapping.cs

```

2 references
public class CsvMovieDescriptionMapping : CsvMapping<MovieDescription>
{
    1 reference
    public CsvMovieDescriptionMapping() : base()
    {
        MapProperty(0, x => x.MovieId);
        MapProperty(1, x => x.YearOfRelease);
        MapProperty(2, x => x.Title);
    }
}

```

Hình 25: CsvMovieDescriptionMapping.cs

CsvMovieMapping.cs

```

2 references
public class CsvMovieMapping : CsvMapping<Movie>
{
    1 reference
    public CsvMovieMapping() : base()
    {
        MapProperty(0, x => x.MovieId);
        MapProperty(1, x => x.UserId);
        MapProperty(2, x => x.Rating);
    }
}

```

Hình 26: CsvMovieMapping.cs

RecommendationSystem.cs

```
1 //reference
public RecommendationSystem(Func<T, int> itemIdFunc, Func<T, int> userIdFunc, Func<T, double> ratingFunc, Func<int, string> itemNameFunc = null)
{
    _itemIdFunc = itemIdFunc;
    _userIdFunc = userIdFunc;
    _ratingFunc = ratingFunc;
    _itemNameFunc = itemNameFunc;
}

1 //reference
public void LoadModel(IEnumerable<T> itemRatings)
{
    _itemRatings = itemRatings;
    _itemSimilarityMatrix = new Dictionary<(int X, int Y), double>();
    var distinctItem = itemRatings.Select(_itemIdFunc).Distinct().ToList();

    int currentProgress = 0, totalCount = distinctItem.Count();
    Console.WriteLine($"Building model based off of {totalCount} items...");
    Parallel.ForEach(distinctItem, firstItemId =>
    {
        var itemName = _itemNameFunc == null ? firstItemId.ToString() : _itemNameFunc(firstItemId);
        var firstItemRatings = _itemRatings.Where(x => _itemIdFunc(x) == firstItemId);
        var firstItemAverageRating = firstItemRatings.Sum(_ratingFunc) / firstItemRatings.Count();
        Interlocked.Increment(ref currentProgress);
        //Console.WriteLine($"[{currentProgress}/{totalCount}] Finished work on item \"{itemName}\" in {elapsed.TotalMinutes} minute(s).");
        Console.WriteLine($"[{currentProgress}/{totalCount}] {itemName} Average: {firstItemAverageRating}");
        //Console.WriteLine($"[{currentProgress}/{totalCount}] {itemName}");
    });
}
```

Hình 27: LoadModel trong RecommendationSystem.cs

```
1 //reference
public double PredictUserRating(int userId, int itemId)
{
    if (_itemRatings == null || _itemSimilarityMatrix == null)
        throw new Exception("There is no data to generate recommendations from.");
    var existingRating = _itemRatings.FirstOrDefault(x => _itemIdFunc(x) == itemId && _userIdFunc(x) == userId);
    if (existingRating != null)
        return _ratingFunc(existingRating);

    var firstItemRatings = _itemRatings.Where(x => _itemIdFunc(x) == itemId);
    var firstItemAverageRating = firstItemRatings.Sum(_ratingFunc) / firstItemRatings.Count();

    var ratings = _itemRatings.Where(x => _userIdFunc(x) == userId);
    var distinctRatedItems = ratings.Select(_itemIdFunc).Distinct();

    IEnumerable<(double WeightedRating, double AbsoluteSimilarity)> itemData = distinctRatedItems.Select(secondItemId =>
    {
        var secondItemRatings = _itemRatings.Where(x => _itemIdFunc(x) == secondItemId);
        var secondItemAverageRating = secondItemRatings.Sum(_ratingFunc) / secondItemRatings.Count();
        return !_itemSimilarityMatrix.TryGetValue((itemId, secondItemId), out var similarity) ? (0, 0) :
            ((_ratingFunc(ratings.First(x => _itemIdFunc(x) == secondItemId)) - secondItemAverageRating) * similarity, Math.Abs(similarity));
    });
    return firstItemAverageRating + itemData.Sum(x => x.WeightedRating) / itemData.Sum(x => x.AbsoluteSimilarity);
}
```

Hình 28: Tìm đánh giá của Id người dùng

Program.cs


```

Console.WriteLine("Reading movie data in from \"data/data-full.txt\"...");
var elapsed = TimUtilities.MeasureDuration(() => movieParser.ReadFromFile("data/data-full.txt", Encoding.ASCII).ToList(), out var data);
var movies = data.Where(x => x.IsValid).Select(x => x.Result).ToList();
//Console.WriteLine($"Data loaded in {elapsed.TotalSeconds} second(s).");

ListMovieDescription> descriptions = new ListMovieDescription();
if (File.Exists("data/movie_names.txt"))
{
    Console.WriteLine("Reading optional movie description data in from \"data/movie_names.txt\"...");
    elapsed = TimUtilities.MeasureDuration(() => descriptionParser.ReadFromFile("data/movie_names.txt", Encoding.ASCII).ToList(), out var descriptionData);
    descriptions.AddRange(descriptionData.Where(x => x.IsValid).Select(x => x.Result));
    //Console.WriteLine($"Data loaded in {elapsed.TotalSeconds} second(s).");
}

Console.WriteLine("Training the recommendation system...");
var recommendationSystem = new RecommendationSystem(x => x.MovieId, x => x.UserId, x => x.Rating,
    x => descriptions.FirstOrDefault(y => y.MovieId == x)?.Title ?? x.InString());
elapsed = TimUtilities.MeasureDuration(() => recommendationSystem.LoadModel(movies));
//Console.WriteLine($"Recommendation system trained in {elapsed.TotalMinutes} minute(s).");

var continueLoop = true;
while (continueLoop)
{
    Console.WriteLine("Enter a user ID to predict a rating for:");
    var userParseSuccess = int.TryParse(Console.ReadLine(), out var userId);
    Console.WriteLine("Enter a movie ID to predict a rating for:");
    var movieParseSuccess = int.TryParse(Console.ReadLine(), out var movieId);

    var rating = recommendationSystem.PredictUserRating(userId, movieId);
    Console.WriteLine($"User \"{userId}\" would most likely rate the movie " +
        $"{descriptions.FirstOrDefault(x => x.MovieId == movieId)?.Title ?? movieId.ToString()}\" {Math.Round(rating, 2)} out of 5. ");

    Console.WriteLine("Enter another? (Y/n)");
    continueLoop = Console.ReadLine().Trim().ToUpperInvariant() == "Y";
}
Console.WriteLine("Press any key to exit...");
Console.ReadKey();

```

Hình 29: Program.cs

Chạy chương trình

Khi chạy chương trình nó sẽ xuất ra danh sách 92 bộ phim đã được xem và điểm trung bình đánh giá bộ phim đó

```

C:\Users\ASUS\Desktop\đồ án cntt\winform\IT-PROJECT\movie-recommendation-system\MovieRecommendationSystem\bin\Debug...
Reading movie data in from "data/data-full.txt"...
Reading optional movie description data in from "data/movie_names.txt"...
Training the recommendation system...
Building model based off of 92 items...
The Night and the Moment. Average: 2.25
Lilo and Stitch. Average: 3.7601078167115904
The Perfect Neighbor. Average: 2.3943661971830985
2002 Olympic Figure Skating Exhibition. Average: 2
Torch Song Trilogy. Average: 3.8132559264634738
Hero. Average: 3.0468497576736673
Manchild: The Complete First Season. Average: 2.5416666666666665
Madea's Family Reunion (Stage Play). Average: 3.7746870653685676
Captain Kronos: Vampire Hunter. Average: 2.8547486033519553
Paul Simon: You're the One: In Concert from Paris. Average: 2.795918367346939
Gremlins. Average: 3.418896833503575
The Secret of Roan Inish. Average: 3.677804944308612
Alien Quadrilogy: Bonus Material. Average: 3.719178082191781
Tom & Viv. Average: 3.0979133226324236
Justice League. Average: 3.6206482593037217
Farscape: The Peacekeeper Wars: Bonus Material. Average: 4.0476190476190474
Soft Skin. Average: 3.2100840336134455
Door to Door. Average: 3.825652594263616
The Murder Room. Average: 3.02
The Ranch. Average: 2.665
Songs from the Second Floor. Average: 2.7759562841530054
Shake. Average: 2.168067226890756
Dragon Ball Z: Lord Slug. Average: 3.8042813455657494
Jack. Average: 3.502688172043011
Sarafina!. Average: 3.256921373200443
Italian Stallion. Average: 1.7532467532467533

```

Hình 30: Danh sách các bộ phim và điểm trung bình của bộ phim đó

```
data-full.txt x Movie.c C:\Users\ASUS\Desktop\dô án cntt\winform\IT-PROJECT\movie-recommendation-system\MovieRecommendationSystem\bin\Debug...
1 28,1391777,4,0 Merci Docteur Rey, Average: 2.6666666666666665
2 28,1999900,5,0 The Lone Gunmen: The Complete Series, Average: 3.374485596707819
3 28,765311,1,0 Dario Argento's The Card Player, Average: 2.5925925925925926
4 28,1007836,4,0 Sharpe 13: Sharpe's Justice, Average: 3.799465248641711
5 28,2103455,4,0 Quest for Camelot, Average: 3.451359884486295
6 28,1468832,4,0 13 Going on 30, Average: 3.502146052472494
7 28,41432,4,0 Bridge of Dragons, Average: 2.484848484848485
8 28,3406678,4,0 George Washington, Average: 3.2899484536882473
9 28,160077,4,0 Kiss Toledo Goodbye, Average: 2.278688524590164
10 28,998129,4,0 Heaven Knows, Average: 3.734351145038168
11 28,30078,3,0 The Life and Times of Judge Roy Bean, Average: 3.4110050533408196
12 28,577397,5,0 Man of the House, Average: 3.4453109378124376
13 28,2615678,5,0
14 28,1276935,5,0
15 28,1663236,4,0
16 28,2196868,5,0 Enter a user ID to predict a rating for:
17 28,1744889,1,0 30878
18 28,2509320,3,0 Enter a movie ID to predict a rating for:
19 28,1174530,4,0 28
20 28,1644750,3,0 User "30878" would most likely rate the movie "Lilo and Stitch" 3 out of 5.
21 28,1753250,3,0 Enter another? (Y/n)
22 28,359446,4,0
23 28,1778851,5,0
24 28,3006799,4,0
25 28,542786,4,0
26 28,2639583,3,0
27 28,2455420,4,0
28 28,940507,5,0
29 28,480477,3,0
30 28,1765153,4,0
31 28,300405,5,0
32 28,309151,4,0
33 28,780936,4,0
34 28,557167,4,0
81 % No issues to
```

Hình 31: Xem UserId 30878 đánh giá bộ phim với MovieId 28

```
Enter a user ID to predict a rating for:
30878
Enter a movie ID to predict a rating for:
28
User "30878" would most likely rate the movie "Lilo and Stitch" 3 out of 5.
Enter another? (Y/n)
n
Press any key to exit...

C:\Users\ASUS\Desktop\dô án cntt\winform\IT-PROJECT\movie-recommendation-system\MovieRecommendationSystem\bin\Debug\netcoreapp3.0\MovieRecommendationSystem.exe (process 1600) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Hình 32: Muốn xem tiếp đánh giá thì chọn Y và nếu không thì chọn n để kết thúc chương trình

PHẦN 3: KẾT LUẬN

Tóm lại, qua đề tài đồ án mà nhóm chúng em đã làm (tìm hiểu về dịch vụ Recommendation của AWS) đã giúp cho chúng em hiểu được dịch vụ gợi ý (đề xuất) này giúp ta như thế nào cho doanh nghiệp cũng như giúp cho chúng ta trong tương lai. Dịch vụ gợi ý (Recommendation) là hệ thống trí tuệ nhân tạo đem lại những sự gợi ý hợp lý, thích hợp, yêu thích nhất cho người dùng, giúp người dùng tìm kiếm dễ dàng và thích hợp nhất, nó được tích hợp vào nhiều ứng dụng, trang web hay trên cả thị trường thương mại mà từ đó giúp tăng doanh thu cho doanh nghiệp của họ ngày một càng phát triển. Vì đây là lần đầu chúng em làm đồ án về AWS nên sẽ có nhiều phần sai sót hoặc có thể không đúng theo điều kiện mà thầy đặt ra, chúng em mong thầy đưa ra ý kiến, nhận xét để chúng em có thể hoàn thiện hơn trong các đề tài tiếp theo mà chúng em sẽ làm trong tương lai.

Kết quả đạt được sau khi hoàn thành đề tài

Theo chúng em thấy, thì chúng em đã hiểu được cách hoạt động của Recommendation System và trên AWS. Về mặt thực nghiệm ứng dụng chúng em đã xây dựng được đồ án đề xuất ra được danh sách các đề xuất mà người dùng có thể quan tâm đến và có thể biết người dùng với ID nào đó đánh giá sản phẩm đó với bao nhiêu điểm.

Nhược điểm

Vì chưa có thể trao đổi trực tiếp với thầy, nên nhóm chúng em đã thiếu hoặc sai sót về phần đề tài này. Vì chúng em chưa có điều kiện đủ để có thể tạo được tài khoản AWS để ứng dụng nhiều vào nó, với áp lực về dịch bệnh dưới quê cũng như đề tài đồ án của các môn học khác mà chúng em đã không thể phát triển thêm một số chức năng mới mẻ hơn. Với phần đề tài thì dữ liệu quá lớn nên khi triển khai sẽ còn chậm chạp.

PHẦN 4: Danh mục tài liệu tham khảo

Trang chính AWS

<https://rpubs.com/chidungkt/634300>

<https://viblo.asia/p/gioi-thieu-ve-he-thong-goi-y-recommendation-systems-aWj53kEp56m>

Recommendation System Wikipedia

<https://www.youtube.com/watch?v=Qz9w-DdIJRg>

<https://cloud-ace.vn/dich-vu-va-giai-phap/recommendation-system/>