XÂY DỰNG VÀ SO SÁNH MỘT SỐ KỸ THUẬT TRONG HỆ KHUYẾN NGHỊ SÁCH

Võ Đinh Ngọc Huyền và Nguyễn Vũ Sao Mai

Trường Đại học Công nghệ Thông tin, khu phố 6, phường Linh Trung, Thành phố Thủ Đức, Thành phố Hồ Chí Minh

Tóm tắt. *Trong lĩnh vực dữ liệu lớn, hệ thống gợi ý ngày càng trở nên phổ biến. Khi số lượng người mua và người bán trực tuyến ngày càng tăng, các kỹ thuật kinh doanh hiệu quả cần được áp dụng để xử lý lượng lớn dữ liệu được tạo ra mỗi ngày. Hệ thống gợi ý đóng một vai trò quan trọng trong việc lọc dữ liệu và cung cấp đầy đủ thông tin cho người dùng. Các kỹ thuật khác nhau như Lọc cộng tác, Dựa trên nội dung và Nhân khẩu học đã được áp dụng để đề xuất nhưng vẫn tồn tại có một số nhược điểm khiến các kỹ thuật này không đưa ra được đề xuất hiệu quả. Trong bài báo này, chúng tôi thực nghiệm xây dựng một số các hệ gợi ý và so sánh xem hệ thống nào nhanh và chính xác hơn.*

Từ khóa: *Big data, Apache Spark, Recommendation System....*

1 Giới thiệu

Recommender System – Trong tiếng Việt được gọi là một hệ thống gợi ý hoặc hệ thống khuyến nghị, là một lớp con của hệ thống lọc thông tin. Hệ thống này sẽ tìm cách dự đoán "xếp hạng" hay "sở thích" một người sử dụng đối với một đối tượng cụ thể. Các hệ thống gợi ý được sử dụng trong nhiều lĩnh vực bao gồm quảng cáo, phim ảnh, âm nhạc, tin tức, sách, bài báo nghiên cứu, truy vấn tìm kiếm, xã hội và sản phẩm nói chung. Ngoài ra hệ thống này còn dùng cho các chuyên gia, cộng tác viên, nhà hàng, hàng may mặc, dịch vụ tài chính, bảo hiểm nhân thọ, các ứng dụng hẹn hò trực tuyến và các trang Twitter.

Một loạt các kỹ thuật đã được đề xuất làm cơ sở cho các hệ thống khuyến nghị như: Các kỹ thuật lọc cộng tác (collaborative), dựa trên nội dung (content-based), dựa trên kiến thức (knowledge-based) và nhân khẩu học (demographic tech-niques). Mỗi kỹ thuật này đều có những thiếu sót, như vấn đề Cold Start đối với các hệ thống lọc cộng tác và dựa trên nội dung (phải làm gì với người dùng mới với ít xếp hạng), tắc nghẽn tri thức (knowledge engineering bottle-neck) trong các phương pháp dựa trên tri thức. Một hệ thống khuyến nghị lai là một hệ thống trong đó kết hợp nhiều kỹ thuật với nhau để đạt được một số sức mạnh tổng hợp giữa chúng.

2 Recommendation System

2.1 Các khái niệm chính

Trong RS, thông thường người ta quan tâm đến ba thông tin chính là người dùng (user), mục tin (item, item có thể là sản phẩm, bộ phim, bài hát, bài báo,.. tùy hệ thống), và phản hồi (feedback) của người dùng trên mục tin đó (thường là các xếp hạng/đánh giá – rating biểu diễn mức độ thích/quan tâm của họ). Các thông tin này được biểu diễn thông qua một ma trận. Ở đó, mỗi dòng là một user, mỗi cột là một item, và mỗi ô là một giá trị phản hồi (ví dụ, xếp hạng) biểu diễn "mức độ thích" của user trên item tương ứng. Các ô có giá trị là những item mà các user đã xếp hạng trong quá khứ. Những ô trống là những item chưa được xếp hạng (điều đáng lưu ý là mỗi user chỉ xếp hạng cho một vài item trong quá khứ, do vậy có rất nhiều ô trống trong ma trận này – còn gọi là ma trận thưa – sparse matrix).

		Items					
		1	*2*	...	*i*	...	*m*
Users	*1*	5	3		1	2	
	2		2				4
	:			5			
	u	3	4		2	1	
	:					4	
	n			3	2		

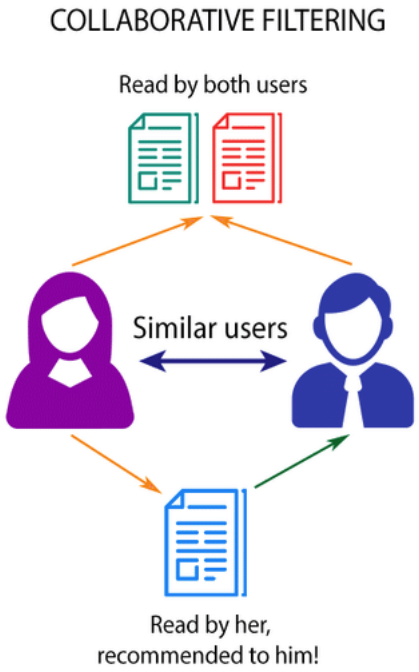
Hình 1: Ma trận Sparse user-item

2.2 Các kỹ thuật chính

Hiện tại, trong RS có rất nhiều giải thuật được đề xuất, tuy nhiên có thể gom chúng vào trong các nhóm chính:

Nhóm giải thuật lọc cộng tác (Collaborative Filtering): Lọc cộng tác là phương pháp khai thác những khía cạnh liên quan đến thói quen sử dụng sản phẩm của cộng đồng người dùng có cùng sở thích trong quá khứ để đưa ra dự đoán các sản phẩm mới phù hợp với người dùng hiện thời. Như vậy, thông tin đầu vào của hệ tư vấn dựa vào phương pháp lọc cộng tác chính là các phản hồi của người dùng về các sản phẩm trong hệ thống. Youtube là hệ thống cho phép người dùng theo dõi, chia sẻ và tư vấn các video trực tuyến cá nhân hóa tới ngườii dùng. Tính năng tư vấn của Youtube được xây dựng sử dụng phương pháp lọc cộng tác dựa vào mô hình để đưa ra tư vấn các video phù hợp với nhu cầu của người dùng. Mô hình được sử dụng ở đây là mạng nơ ron sâu (Deep Neural Networks). Netflix là hệ thống hệ tư vấn về phim rất lớn sử dụng phương pháp lọc cộng tác dựa vào mô hình thừa số hóa ma trận để đa ra tư vấn các sản phẩm phù hợp cho người dùng hiện thời. Amazon là hệ tư vấn thương mại điện tử rất nổi tiếng sử dụng phương pháp

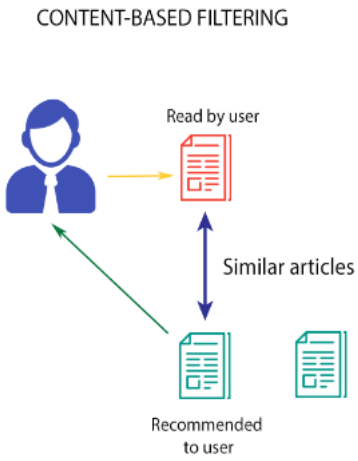
lọc cộng tác theo bộ nhớ dựa vào sản phẩm. Hệ tư vấn này khai thác thông tin đầu và là ma trận đánh giá thu được thông qua các đánh giá tường minh của người dùng với sản phẩm để huấn luyện và đưa ra tư vấn.



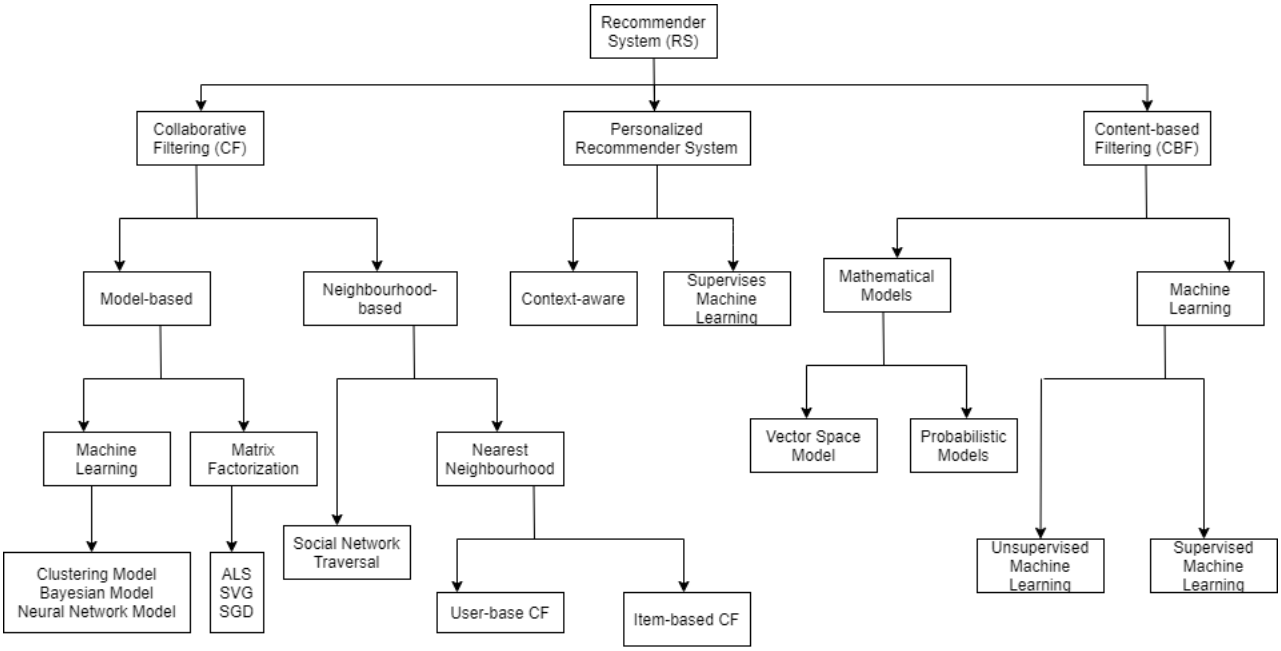
Hình 2: Phương pháp lọc cộng tác

Nhóm giải thuật lọc trên nội dung (Content-based Filtering): Lọc theo nội dung dựa vào mô hình là phương pháp sử dụng toàn bộ tập hồ sơ sản phẩm hoặc tập hồ sơ người dùng để thực hiện huấn luyện. Kết quả của mô hình huấn luyện sẽ sử dụng trong mô hình dự đoán để sinh ra tư vấn cho người dùng. Trong cách tiếp cận này, lọc theo nội dung có thể sử dụng các kỹ thuật thống kê, học máy như: mạng Bayes, phân cụm, cây quyết định, mạng nơ ron nhân tạo... để sinh dự đoán cho người dùng. Pazzani và Billsus [12] sử dụng bộ phân loại Bayes dựa trên những đánh giá "thích" hoặc "không thích" của người dùng để phân loại các sản phẩm. Solombo [13] đề xuất mô hình lọc thích nghi, trong đó chú trọng đến việc quan sát mức độ phù hợp của tất cả các sản phẩm. Zhang [14] đề xuất mô hình tối ưu tập các sản phẩm tương tự dựa vào giá trị ngưỡng. Trong đó, giá trị ngưỡng được ước lượng dựa trên tập sản phẩm thích hợp và tập sản phẩm không thích hợp với mỗi hồ sơ người dùng.

Nhóm lai ghép: Hybrid Filtering là sự kết hợp của hai giải thuật Content-based Filtering và Collabrative Filtering: Hybrid Fitering được sử dụng mềm dẻo khi hệ thống Collabrative Filtering không có các hành vi (ratings), khi đó hệ thống sẽ sử dụng Content-based Filtering và ngược lại, khi Content-based Filtering không có các feature cần thiết trong việc đánh giá thì hệ thống sẽ sử dụng Collaborative Fitering để thay thế.



Hình 3: Phương pháp lọc dựa trên nội dung



Hình 4: Tổng quan một số kỹ thuật sử dụng trong Recommender System

3 Cài đặt thực nghiệm

Apache spark là công cụ xử lý dữ liệu quy mô lớn. Chúng tôi đã triển khai các thuật toán là lọc cộng tác và ước tính mức độ phổ biến bằng cách sử dụng khung công tác apache spark. MLlib là thư viện học máy có thể mở rộng của Apache Spark. Spark đưa dữ liệu vào RAM và sau đó xử lý nó không giống như Hadoop, nơi phần lớn thời gian dành cho việc ghi dữ liệu từ đĩa vào bộ nhớ.

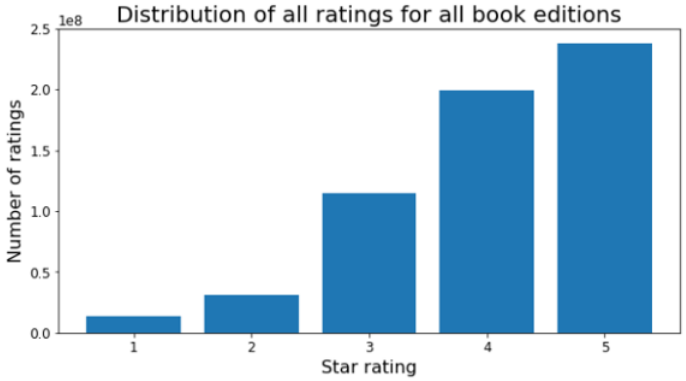
3.1 Bộ dữ liệu

Chúng tôi đã sử dụng tập dữ liệu Goodbooks-10k từ trang web: https://www.kaggle.com/zygmunt/goodbooks-10k. Datatsets chứa 6 triệu xếp hạng cho 10.000 sách phổ biến trên Goodreads từ hơn 50.000 người dùng. Tập dữ liệu chứa 5 tệp CSV:

- ratings.csv: chứa các xếp hạng được sắp xếp theo thời gian.
- to-read.csv: cung cấp ID của những cuốn sách được mỗi người dùng đánh dấu là 'to-read'.
- books.csv: có chứa dữ liệu của từng cuốn sách.
- book-tags.csv: chứa tags/shelves/genres được người dùng gán cho sách.
- tags.csv: tên của các thẻ.

Books (books.csv)

Có 10.000 cuốn sách phổ biến nhất trong tập dữ liệu trong đó mức độ phổ biến được đo lường bằng cách nhiều xếp hạng mà một cuốn sách đã nhận được. Dữ liệu sách chứa số nhận dạng sách, tên sách, tác giả, ISBN, ngày xuất bản, liên kết đến hình ảnh. Ngoài ra, mỗi cuốn sách có một số xếp hạng và các bài đánh giá mà cuốn sách đã nhận được và tổng số tất cả các xếp hạng mà tất cả các ấn bản của cuốn sách nhận được đã được chia nhỏ xếp hạng từ 1 sao đến 5 sao.

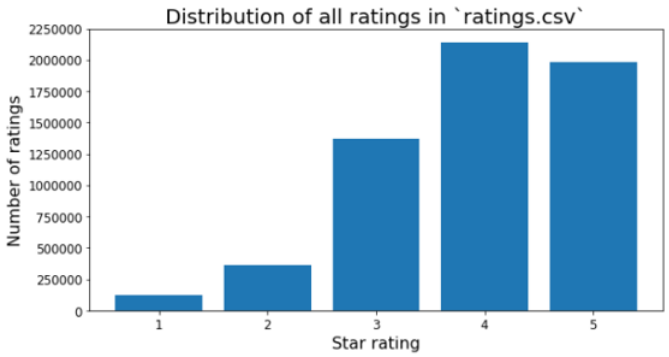


Hình 5: Biểu đồ thể hiện sự phân bố các mức đánh giá trong books.csv

Số lượng xếp hạng có chiều hướng tích cực tăng dần vì: Tập dữ liệu chứa những cuốn sách phổ biến nhất trên Goodreads, sách phổ biến nhất là những sách mọi người thích đọc, vì vậy chúng được đánh giá cao; Nhiều người bị thu hút bởi những cuốn sách mà họ có thể thích, hoặc thông qua truyền miệng hoặc bằng cách đọc các đề xuất biên tập khác nhau, những cuốn sách đó có nhiều khả năng nhận được một đánh giá tích cực sau đó nếu một người đã chọn ngẫu nhiên; Theo thói quen, một số người dùng thường xếp hạng tốt cho tất cả mọi thứ trước khi quan tâm đến nội dung.

Book ratings (ratings.csv)

Có 5,976,479 xếp hạng trong tập dữ liệu. Mỗi xếp hạng bao gồm user-id, book-id và một điểm xếp hạng từ 1 đến 5. Số lượng xếp hạng cho mỗi cuốn sách từ 8 đến 22.806 và xếp hạng theo phạm vi người dùng từ 19 đến 200. Chúng tôi nhận thấy phân phối xếp hạng khác nhau: có nhiều xếp hạng 4 sao hơn 5 sao trong tập dữ liệu so với biểu đồ phân phối được trình bày trước đó. Điều này cho thấy ratings.cvs chỉ chứa một tập con xếp hạng trong books.csv.



Hình 6: Biểu đồ thể hiện sự phân bố các mức đánh giá trong ratings.csv

Book to read (to-read.csv)

Có 912.705 cặp user-book trong tập dữ liệu này. Sách to-read có thể là một bổ sung mạnh mẽ báo hiệu xếp hạng vì 99,86 (%) những tựa sách này nằm trong danh sách đọc của user khác và 91,48 (%) user sau khi thêm vào mục "to-read" thì đều đọc sách đó..

Tags (tags.csv và book-tags.csv)

Có 34.252 thẻ do người dùng chỉ định trong tập dữ liệu này. Có hàng chục hàng nghìn thẻ và nhiều thẻ có nội dung tương tự. Ví dụ: Tag 681 có "to-read" trong tên của chúng mà một số ít được chọn ngẫu nhiên là “want-to-read”, “to-read-maybe-someday” "to-read-memoir" và "to-read-cookbook." Ngoài ra, các thẻ cũng khác nhau về mặt ngôn ngữ. Các thẻ có cùng ý nghĩa nhưng cách viết khác nhau nên được coi là tương tự. Một số ví dụ về tiếng Đức, tiếng Pháp là “noch-zu-lesen, “101-à-lire-dans-sa-vie” tương ứng.

3.2 Phương pháp thực nghiệm

Hai cách tiếp cận đề xuất thường được sử dụng là Collaborative Filtering (uuCF và ALS) và Content Based.

Collaborative Filtering sử dụng Alternating Least Squares (ALS)

Alternating Least Squares (ALS): Trong phương pháp này, chúng ta có một ma trận lớn và chúng ta nhân nó thành các ma trận nhỏ hơn thông qua các bình phương nhỏ nhất xen kẽ. Chúng tôi kết thúc với hai hoặc nhiều ma trận chiều thấp hơn có tích bằng ma trận ban đầu. ALS có sẵn trong Apache Spark. Để xử lý trước dữ liệu, các cột productID và reviewerID phải được chuyển đổi từ chuỗi sang số nguyên bằng cách sử dụng trình chỉ mục chuỗi để chuẩn bị cho chúng cho hồi quy ALS. Trong khi xây dựng mô hình đề xuất bằng cách sử dụng ALS trên dữ liệu đào tạo, 'chiến lược bắt đầu lạnh' đã bị loại bỏ để tránh nhận được giá trị NaN trong các chỉ số đánh giá. Bắt đầu nguội xảy ra khi nó cố gắng dự đoán xếp hạng cho một cặp người dùng-mục nhưng không có xếp hạng nào cho người dùng / mục này trong tập huấn luyện. Việc loại bỏ chiến lược khởi động lạnh chỉ đơn giản là loại bỏ các hàng / cột đó khỏi các dự đoán và khỏi tập thử nghiệm. Do đó, kết quả sẽ chỉ chứa các số hợp lệ có thể được sử dụng để đánh giá.

```
# define parameters
als = ALS(maxIter=20,
          regParam=0.01,
          userCol="user_id",
          itemCol="book_id",
          ratingCol="rating",
          coldStartStrategy="drop")
#fit the model to the ratings
model = als.fit(ratings_df)
```

Hình 7: Các thông số đầu vào cho mô hình ALS

Collaborative Filtering sử dụng User-based

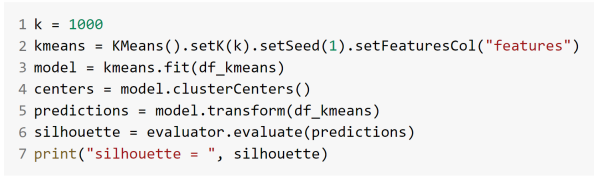
User-based CF: Giả sử trong quá khứ của người dùng A, người này thích giày thể thao hiệu XYZ, mũ hiệu UVT, kính râm hiệu MNP. Cũng tương tự trong quá khứ của B, B cũng thích giày thể thao hiệu XYZ, mũ hiệu UVT. Ta nhận thấy rằng sở thích của hai người này khá là giống nhau, do A có thích thêm kính râm hiệu MNP, có khả năng cao rằng B cũng thích kính râm hiệu MNP này, ta sẽ gợi ý cho B mua sản phẩm này. .

Content Based Filtering sử dụng K-Means

K-Means: K- mean là một thuật toán học không giám sát được sử dụng phổ biến. Đối với phương pháp K-means, chúng tôi sẽ sử dụng file books.csv. Trong đó, chúng ta chỉ sử dụng hai cột là "authors" và "average-rating" dùng làm thuộc tính để tiến hành gom cụm, như vậy, những user nào cùng thích một số

Hình 8: Sử dụng Suprise Library để xây dựng model

tác giả nào đó thì cũng sẽ thích những sách mà user khác có cùng sở thích đã đọc. Trước khi tiến hành xây dựng thuật toán gom cụm, các cột có type là string cần được chuyển thành dạng numerical. Tiếp theo, nhóm sinh viên tiến hành tạo feature để là đầu vào cho thuật toán thông qua hàm VectorAssembler của thư viện pyspark.ml.feature, sau đó tiến hành sử dụng thuật toán k-means để xây dựng model và đánh giá kết quả. Ban đầu số lượng cụm thử nghiệm là từ 2 đến 50. Nhóm sinh viên đã sử dụng thang đo silhouette để đánh giá mô hình k-means, theo đó, giá trị silhouette giảm dần khi số lượng cụm tăng lên đến 100 và theo dõi kết quả.



Hình 9: Code thực hiện

3.3 Phương pháp đánh giá

Có nhiều phương pháp khác nhau có thể được sử dụng để đánh giá giải thuật như: F-Measure, Area Under the ROC Curve (AUC)... mỗi phương pháp đánh giá sẽ thích hợp cho từng lĩnh vực cụ thể. Trong RS, độ đo lỗi RMSE (Root Mean Squared Error) là độ đo phổ biến mà cộng đồng người dùng trong lĩnh vực RS thường sử dụng. RMSE hay được dùng cho bài toán dự đoán xếp hạng (Rating Prediction) còn MAE hay được dùng cho dự đoán mục tin (Item Prediction) (Guy and Asela, 2011). RMSE và MAE được xác định bằng các công thức sau:

$$RMSEP = \frac{\sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}}{\bar{y}} \quad (1)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|e_i| \quad (2)$$

Hệ số hình bóng (silhouette) cũng là một phương pháp để tìm số tối ưu của các cụm và giải thích và xác nhận tính nhất quán trong các cụm dữ liệu. Phương pháp hình bóng tính toán các hệ số hình bóng của mỗi điểm đo bao nhiêu điểm tương tự với cụm của chính nó so với các cụm khác. bằng cách cung cấp một biểu diễn đồ họa ngắn gọn về mức độ phân loại của từng đối tượng.

$$\mathbf{s(i)} = \frac{\mathbf{b(i)} - \mathbf{a(i)}}{\mathbf{max(b(i), a(i))}}$$

4 Kết quả thực nghiệm

Trong số các mô hình được triển khai, ALS cho chúng ta kết quả tốt hơn và hệ thống khuyến nghị chính xác hơn. ALS theo dõi sở thích của người dùng và lịch sử trước đây của người dùng để tùy chỉnh dự đoán sản phẩm tốt hơn cho từng người dùng. Hơn nữa, ALS có thể được tạo bằng cách sử dụng cả người dùng-người dùng và mục-item. Ngược lại, K-means tốt khi dữ liệu người dùng không có sẵn và chúng tôi vẫn mong muốn cung cấp các đề xuất cho người dùng. K-means ban đầu có thể được sử dụng bởi các nhà bán lẻ mới để cung cấp các đề xuất cho người dùng của họ vì họ không có nhiều dữ liệu khách hàng.

	RMSE
ALS	0,473
User-based	0,842

Bảng 1: Độ tin cậy của giải thuật

	Cold Start	**Thiếu dữ liệu**
ALS	Có	Không
K-means	Không	Không
User-based	Có	Có

Bảng 2: Các kết quả khác

4.1 User-based

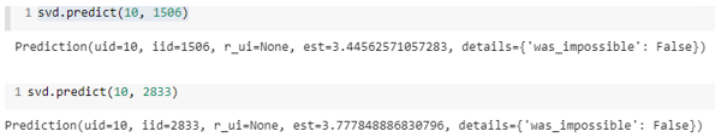
Trong quá trình thực nghiệm, chúng tôi sẽ không triển khai Lọc cộng tác từ đầu. Thay vào đó, chúng tôi sẽ sử dụng thư viện Surprise để sử dụng các thuật toán như Phân hủy giá trị đơn lẻ (SVD) để giảm thiểu

RMSE (Root Mean Square Error) và đưa ra các khuyến nghị tốt nhất. Chúng tôi nhận được Root Mean

	RMSE	**MAE**
User-based	0.842	0,695

Bảng 3: Kết quả của User-based

Square Error khoảng 0,8419, một kết quả khá tương đồng so với những nghiên cứu trước đó. Chúng tôi đã chọn ra user có ID=10 và kiểm tra kết quả mà họ đưa ra sau khi được khuyến nghị.



Hình 10: Kết quả kiểm tra

Đối với cuốn sách có ID 1506, chúng tôi nhận được dự đoán ước tính là 3,44, đối với cuốn sách có ID 2833, chúng tôi nhận được dự đoán ước tính là 3,77. Hệ gợi ý này hoàn toàn dựa trên ID sách được chỉ định và cố gắng dự đoán xếp hạng dựa trên cách những người dùng khác đã dự đoán sách.

	book_id	user_id	rating	title
150478	1506	10	4	The Zahir
282986	2833	10	4	The Prisoner of Heaven (The Cemetery of Forgot...
340448	3409	10	5	The Winner Stands Alone
393966	3946	10	5	Matterhorn
452158	4531	10	4	The Joke
506878	5084	10	2	The Sheltering Sky
588312	5907	10	4	Our Mutual Friend
590191	5926	10	2	The Night Watch
610487	6131	10	2	The Longest Day
696035	7002	10	5	A Mercy
743400	7486	10	4	Great House
759424	7651	10	4	All the Names
855593	8653	10	4	The End of Mr. Y
911432	9240	10	3	Arthur & George

Hình 11: Chọn user có ID=10 để kiểm tra

4.2 ALS

ALS cho chúng ta kết quả tốt và chính xác nhất. ALS theo dõi sở thích của người dùng và lịch sử trước đây của người dùng để tùy chỉnh dự đoán sản phẩm tốt hơn cho từng người dùng. Hơn nữa, ALS có thể được tạo bằng cách sử dụng cả user-user và item-item. Chúng tôi nhận được Root Mean Square Error khoảng

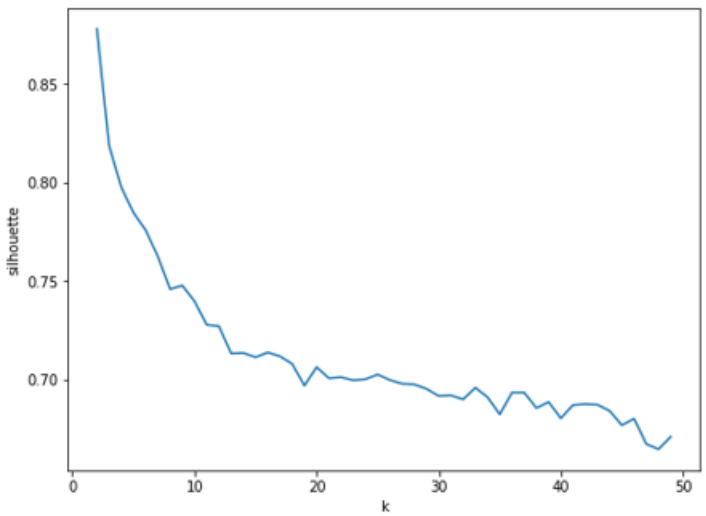
maxIter	regParam	RMSE
10	0,1	0,595
15	0,1	0,586
20	0,1	0,581
10	0,01	0,503
15	0,01	0,482
20	**0,01**	**0,473**

Bảng 4: Các kết quả của các thông số trong ALS

0,8419, một kết quả khá tương đồng so với những nghiên cứu trước đó. Chúng tôi đã chọn ra user có ID=10 và kiểm tra kết quả mà họ đưa ra sau khi được khuyến nghị.Trong quá trình thực hiện, chúng tôi chọn ra được cặp giá trị maxIter=20 và regParam=0.01 cho ra RMSE thấp nhất. Kết quả dự đoán tương đối chính xác trong tập dữ liệu này.

4.3 K-means

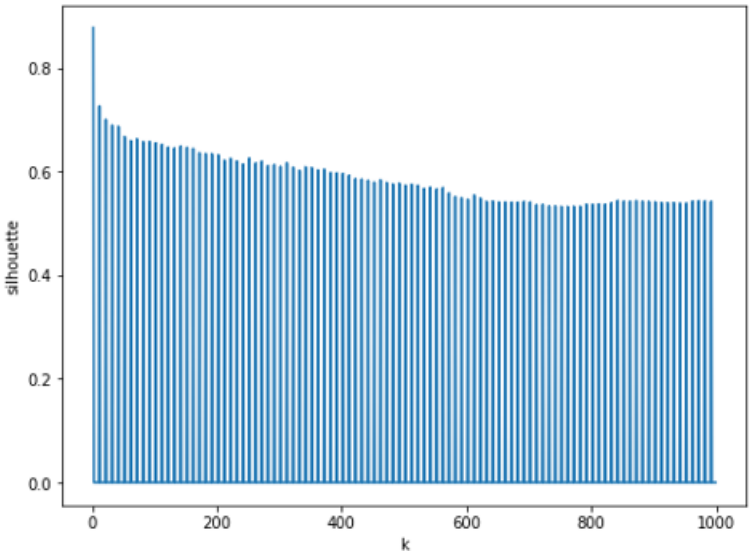
Chúng tôi đã sử dụng thang đo Silhouette để đánh giá mô hình k-means, theo đó, giá trị Silhouette giảm dần khi số lượng cụm tăng lên.



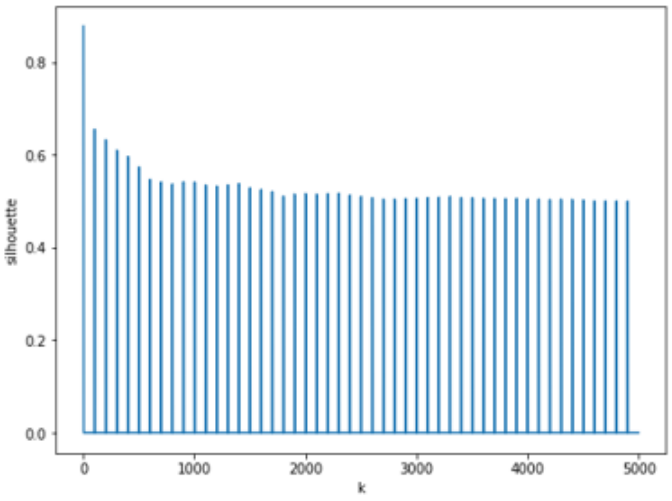
Hình 12: Kết quả biểu thị giá trị Silhouette

Sau khi nâng số lượng cụm thử nghiệm lên 1000, thu được kết quả như hình 13.

Nhìn chung, từ số lượng cụm thứ 700 thì silhouette đạt được trở nên bão hòa ở mức 0.6. Nhóm chúng



Hình 13: Kết quả sau khi nâng số lượng cụm



Hình 14: Kết quả sau khi nâng số lượng cụm

tôi cũng đã nâng số lượng cụm lên 5000 và giá trị silhouette tốt nhất có thể đạt được dừng lại ở mức 0.495.

5 Kết luận và hướng phát triển

Có thể nói các hệ thống gợi ý áp dụng các giải thuật khai phá dữ liệu, học máy nhằm giúp thu thập thông tin cá nhân trên Internet, đồng thời giúp giảm bớt vấn đề quá tải thông tin với các hệ thống truy xuất thông tin và cho phép người dùng truy cập vào các sản phẩm và dịch vụ trên hệ thống. Trong số các mô hình được triển khai, ALS cho chúng ta kết quả tốt hơn và hệ thống khuyến nghị chính xác hơn. Tuy nhiên, việc thực hiện các khuyến nghị trong dự án này còn rất thô sơ và mang tính chất sơ khai và có rất nhiều cải tiến và khả năng cải thiện độ chính xác của các mô hình. Trước hết, để làm cho các dự đoán chính xác và thực tế hơn, dữ liệu có thể được thu thập trực tiếp thông qua các trang web bán lẻ bằng cách sử dụng các công cụ crawl cho các trang web. Ngoài ra, các mô hình phức tạp hơn như mô hình hóa chủ đề hoặc LDA (Phân bổ Dirichlet tiềm ẩn) có thể được triển khai để gán và xác định các từ trong các cụm K-means, trong đó các chủ đề được chỉ định và điểm số liên quan của chúng với các từ có thể hoạt động như một cơ sở logic dự đoán.

Tài liệu tham khảo

[1] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, Large-scale parallel collaborative filtering for the Netflix prize, Berlin, Heidelberg, Springer-Verlag, In AAIM '08, pages 337– 348, 2008.

[2] Sp Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing, Technical Report UCB/EECS-2011-82, EECS Department, University of California, Berkeley, 2011 Computational Linguistics, Vancouver, Canada (2017).

[3] Michael Armbrust, et al, "Spark SQL: Relational Data Processing in Spark", in Proceedings of Association for Computing Machinery, Inc. ACM 978-1-4503-2758, Pages 1383-1394, May 27, 2015.

[4] M. Isard and Y. Yu. Distributed data-parallel computing using a high-level programming language. Pages 987-994 In SIGMOD, 2009.

[5] Z.-D. Zhao and M.-S. Shang, "User-based collaborative-filtering recommendation algorithms on hadoop," in Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on. IEEE,2010, pp. 478–481.

[6] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. Journal of Information Science, vol. 32 no. 2 198-208, April, 2006.

[7] J. Jiang, J. Lu, G. Zhang, and G. Long, "Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop," in Services (SERVICES), 2011 IEEE World Congress on. IEEE, 2011,pp. 490–497.

[8] Ungar LH, Foster DP. "Clustering methods for collaborative filtering". In AAAI workshop on recommendation systems (Vol. 1, pp. 114-129), Jul 26 1998.

[9] Ungar LH, Foster DP. "Clustering methods for collaborative filtering". In AAAI workshop on recommendation systems (Vol. 1, pp. 114-129), Jul 26 1998.

[10] Ekman, P.: In: Facial expression and emotion. vol. 48, pp. 384–392. American Psychologist (1993).

[11] Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).

[12] M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The143 Identification of Interesting Web Sites," Mach. Learn. - Spec. issue multistrategy Learn., vol. 27, no. 3, pp. 313–331, 1997

[13] G. L. Somlo, A. E. Howe, G. L. Somlo, and A. E. Howe, "Adaptive Lightweight Text Filtering Adaptive Lightweight Text Filtering," in IDA 2001: Advances in Intelligent Data Analysis, 2001, vol. 2189, pp. 319–329.

[14] Y. Zhang and J. Callan, "Maximum Likelihood Estimation for Filtering Thresholds," in SIGIR 01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 01, pp. 294–302

[15] LNCS Homepage, http://www.springer.com/lncs, truy cập lần cuối ngày 27/7/2021.

[16] Hoàng Đinh, Xây dựng Content-based Filtering RS, https://viblo.asia/p/xay-dung-content-based-filtering-rs-recommender-system-co-ban-phan-2-bWrZnVovZxw, truy cập lần cuối ngày 26/7/021.

[17] snehalnair , als-system-pyspark https://github.com/snehalnair/als-recommender-pyspark/blob/master/RecommendatiEngine-MovieLens.ipynb, truy cập lần cuối ngày 28/7/2021.

[18] Reinalynn, /Building-a-Book-Recommendation-System-using-Python https://github.com/Reinalynn/Building-a-Book-Recommendation-System-using-Python/tree/master/Code, truy cập lần cuối ngày 28/7/2021.

[19] Sunith Shetty, How to build a cold-start friendly content-based recommender using Apache Spark SQL, https://hub.packtpub.com/how-to-build-a-cold-start-friendly-content-based-recommender-using-apache-spark-sql/, truy cập lần cuối ngày 27/7/2021.

[20] Snehal Nair, PySpark Collaborative Filtering with ALS, https://towardsdatascience.com/build-recommendation-system-with-pyspark-using-alternating-least-squares-als-matrix-factorisation-ebe1ad2e7679, truy cập lần cuối ngày 29/7/2021.