Product Recommendation Based On Customer's Search

Bùi Nguyễn Hoàng Anh[1], Trần Hàm Dương[2], Văn Thiên Luân[3], Võ Minh Thiện[4], Đỗ Trọng Hợp[5]

[1,2,3,4,5] Trường Đại Học Công Nghệ Thông Tin, ĐHQG-TP.HCM
anhbnh.15@grad.uit.edu.vn,
duongth.15@grad.uit.edu.vn,
luanvt.15@grad.uit.edu.vn,
thienvm.15@grad.uit.edu.vn,
hopdt@uit.edu.vn

Abstract. Hiện nay nhu cầu mua sắm trực tuyến cũng như các hệ thống thương mại điện tử (E-Commerce) ngày càng phát triển. Mỗi ngày, số lượng người dùng truy cập các website bán hàng để tìm kiếm sản phẩm rất nhiều và sinh ra một lượng dữ liệu khổng lồ. Với sự phát triển của các mô hình máy học cũng như hệ thống dữ liệu lớn (big data), nhóm thực hiện mong muốn xây dựng một hệ thống gợi ý sản phẩm tự động dựa vào những gì khách hàng tìm kiếm nhằm tăng trải nghiệm của người dùng tốt hơn.

Keywords: pyspark, NLP, search, regression, relevant score.

1 Giới thiệu tổng quan

Ngày nay, thương mại điện tử đang phát triển như vũ bão dựa trên nhu cầu mua sắm không ngừng tăng lên của con người. The Home Depot là hãng bán lẻ thiết bị xây dựng nhà ở lớn nhất tại Mỹ. Khách hàng khi vào website có thể nhập tên sản phẩm mong muốn vào ô search để tìm. Hệ thống sẽ tăng trải nghiệm của khách hàng bằng cách phát triển một model có thể dự đoán cao nhất mức độ liên quan Search relevance của sản phẩm với đoạn text mà người dùng đang nhập vào ô Search từ đó đưa ra các gợi ý chính xác hơn giúp tăng trải nghiệm và giảm thời gian tìm kiếm của khách hàng.

Xuất phát từ ý tưởng của bài toán đó, nhóm chúng tôi mong muốn xây dựng nên một hệ thống tìm kiếm và gợi ý cho người dùng những sản phẩm phù hợp nhất với họ. Đầu tiên sẽ là việc xây dựng một mô hình máy học nhằm dự đoán và trả về kết quả mong muốn, kế tiếp là sẽ xây dựng hệ thống có giao diện trực quan, dễ sử dụng với phần lõi bên dưới sẽ là các mô hình dự đoán đã được huấn luyện sẵn và trả kết quả dự đoán về cho người dùng trong thời gian thực.

Các phương pháp nhóm thực hiện gồm có: phân tích dữ liệu; kỹ thuật xử lý ngôn ngữ tự nhiện cho dữ liệu text; rút trích đặc trưng dữ liệu text; huấn luyện mô hình regression với pyspark; thu thập dữ liệu hình ảnh tự động từ kho hình ảnh của google; hệ thống web-app với giao diện cho người dùng truy vấn từ khóa tìm kiếm và hiển thị kết quả, phần backend server xử lý liên kết các thành phần bao gồm: truy vấn của frontend, lưu trữ cơ sở dữ liệu, phần dự đoán của mô hình máy học,…

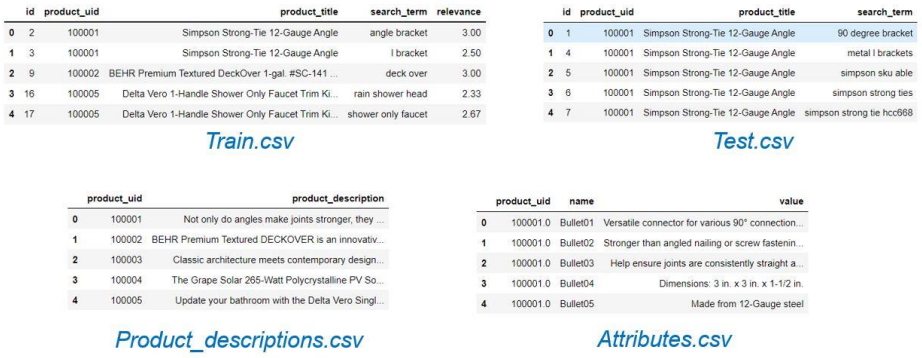
2 Xây dựng mô hình dự đoán

Ở phần này nhóm sẽ trình bày các bước để xây dựng một mô hình dự đoán "relevant score" dựa trên nguồn dữ liệu và thông tin yêu cầu từ cuộc thi "Home Depot Product Search Relevance" trên hệ thống kaggle[1]. Từ đó, áp dụng này để xây dựng hệ thống tìm kiếm mong muốn.

[1] https://www.kaggle.com/c/home-depot-product-search-relevance/

2.1 Phân tích dữ liệu

Dữ liệu cuộc thi bao gồm 2 tập: train (74067 mẫu) và test (166693 mẫu); ngoài ra còn có các file kèm thông tin bổ trợ như *product_descriptions.csv* (chứa thông tin mô tả của sản phẩm) và *attributes.csv* (chứa thông số thuộc tính của sản phẩm, một số sản phẩm không có thuộc tính này). Mô tả và một số ví dụ được mô tả trong hình 1 và bảng 1 bên dưới.

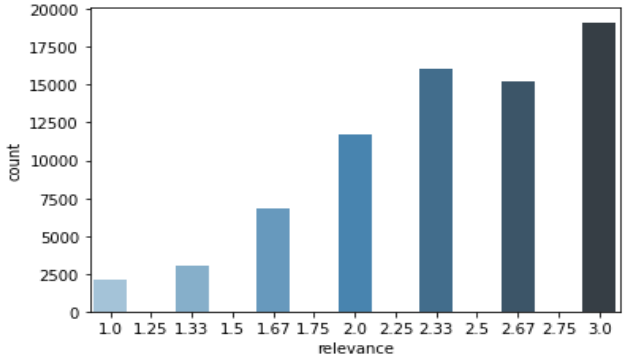


Hình 1. *Một số ví dụ về dữ liệu*

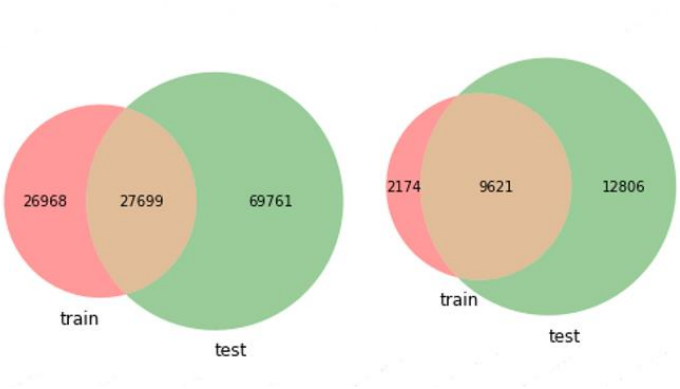
Bảng 1. *Mô tả thông tin của các trường dữ liệu*

Tên trường	Mô tả
Id	id của một cặp search_term và sản phẩm.
Product_uid	id của sản phẩm
Product_title	tiêu đề của sản phẩm
Product_description	mô tả của sản phẩm
Search_term	đoạn văn bản tìm kiếm của người dùng được ghi lại
Relevance	điểm tương thích giữa search_term và sản phẩm gợi ý
Name	tên thuộc tính
Value	giá trị thuộc tính

Nhóm nhận thấy cả trong tập train và test đều không có giá trị rỗng (NULL). Các giá trị của "relevance score" có giá trị từ 1 đến 3 và có phân bố như hình 2. Bên cạnh đó, thống kê sự phân bố id của sản phẩm (Product_uid) và văn bản tìm kiếm của người dùng (Search_term) trong 2 tập train, test được mô tả trong hình 3.



Hình 2. *Phân bố của relevance score*



Hình 3. *Số lượng của các product_uid (ẢNH TRÁI) và search_term (ẢNH PHẢI) xuất hiện ở 2 tập train và test*

2.2 Làm sạch dữ liệu

Nhóm nhận thấy phần lớn dữ liệu có ở dạng text nên cần phải áp dụng các kỹ thuật xử lý trong xử lý ngôn ngữ tự nhiên (NLP) cũng như cần làm dữ liệu kỹ lưỡng.

Trước tiên, nhóm xử lý dữ liệu ở file attributes.csv, nhóm nhận thấy mỗi sản phẩm có chứa khá nhiều thuộc tính mà tên thì chỉ khác nhau một vài từ phía sau ví dụ như trường name của tập này chứa các giá trị là Bullet0x với x = 0, 1, 2, 3, 4, 5, 6, 7, 8. và ứng với mỗi name như này thì các thuộc tính lại khác nhau. Do đó, ta sẽ gộp các cột có cùng id đấy thành một và các thuộc tính khác nhau 1 chút thì chỉ lấy 1 phần chung lớn nhất và các phần khác đi kèm phía sau.

Kế tiếp, nhóm sẽ hợp giữa file *attributes.csv* đã xử lý này và file *product_description.csv* thành một file như nhất dựa vào phần chung product_uid của sản phẩm như Hình 4. Ngay sau đó, gộp cả hai thông tin attributes và product description thành một thông tin sản phẩm duy nhất như Hình 5 . Ta cũng làm tương tự cho tập test.

	product_uid	product_description	product_attributes
0	100001	not only do angles make joints stronger, they ...	Bullet Versatile connector for various 90° con...
1	100002	behr premium textured deckover is an innovativ...	Application Method Brush,Roller,Spray Assemble...
2	100003	classic architecture meets contemporary design...	Built-in flange Yes Bullet Slightly narrower f...
3	100004	the grape solar 265-watt polycrystalline pv so...	Amperage (amps) 8.56 Bullet Positive power tol...
4	100005	update your bathroom with the delta vero singl...	Bath Faucet Type Combo Tub and Shower Built-in...

Hình 4. *Dữ liệu sau khi kết hợp product_descriptions.csv và attributes.csv*

	product_uid	product_description_attributes
0	100001	not only do angles make joints stronger, they ...
1	100002	behr premium textured deckover is an innovativ...
2	100003	classic architecture meets contemporary design...
3	100004	the grape solar 265-watt polycrystalline pv so...
4	100005	update your bathroom with the delta vero singl...

Hình 5. *Dữ liệu cuối cùng sau khi kết hợp các thông tin lại một lần nữa*

Kế tiếp, nhóm sẽ tiến hành làm sạch dữ liệu trên một bảng thông tin sản phẩm tổng hợp cuối cùng này. Các phương pháp được sử dụng như: xóa hết các ký tự đặc biệt; các chữ cái đứng một mình không mang ngữ nghĩa cũng sẽ bị gỡ bỏ; các giá trị ở trường mô tả khá dài kể cả khi chưa được nối với bảng thuộc tính và vì nó mô tả chi tiết sản phẩm nên trong đoạn text của nó chứa nhiều stopword không liên quan trực tiếp tới sản phẩm ta có thể xóa các stopword ở trường này; đưa các tokens về nghĩa gốc của nó dễ dàng hơn trong việc so sánh các tokens sau này;…

Có thể sử dụng 1 trong 2 (hoặc cả hai) kỹ thuật đưa một từ về dạng gốc của nó đó là stemmer hoặc Lemmatization. Về Stemmer : thực hiện nhanh vì nó không so khớp với từ điển chỉ đơn giản lược bỏ các hậu tố phía sau từ vì thế mà không thể áp dụng cho các biến thể là quá khứ phân từ được. Về Lemmatization : thực hiện bằng cách so khớp với từ điển nên thời gian thực hiện lâu hơn.

Đến đây dữ liệu cơ bản đã được làm sạch và có thể tiến tới rút trích các đặc trưng cần thiết để đưa vào mô hình huấn luyện.

2.3 Rút trích đặc trưng

Nhiệm vụ của bài toán là dự đoán relevance score của các cặp search_term và product_id, do vậy nên ta sẽ tập trung khai thác các đặc trưng từ độ liên quan của search_term là chủ yếu.

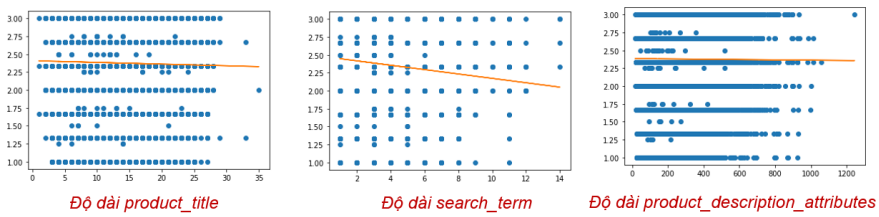
Các đặc trưng đầu tiên sẽ là độ dài của lần lượt các trường *product_title*, *search_term* và *product_description_attributes*. nếu cả 3 đều thỏa mãn là có xu hướng ảnh hưởng tới relevance. Ở 3 đặc trưng đầu này có vấn đề đó là ta cần phải đánh giá được độ liên quan của 3 trường trên đối với relevance và tất nhiên là sẽ dùng tập train để đánh giá nhưng mà trường *product_description_attributes* được tạo ra sau khi nối bảng còn 2 trường trên thì tồn tại ngay từ tập train ban đầu. Vì vậy mà ta cần đánh giá 3 trường trên ở 2 tập dữ liệu khác nhau, product_title và search_term ở tập train ban đầu và product_description_attributes sau khi nối.

Khi xây dựng công cụ tìm kiếm và khi lấy mẫu đánh giá thực nghiệm relevance thì chắc chắn chúng ta sẽ dựa vào sự xuất hiện của các từ trong trường search_term ở các trường khác để đưa ra kết quả liên quan do đó mà các đặc trưng tiếp theo sẽ lấy từ việc đếm độ xuất hiện của các từ trong search_term ở các trường khác:

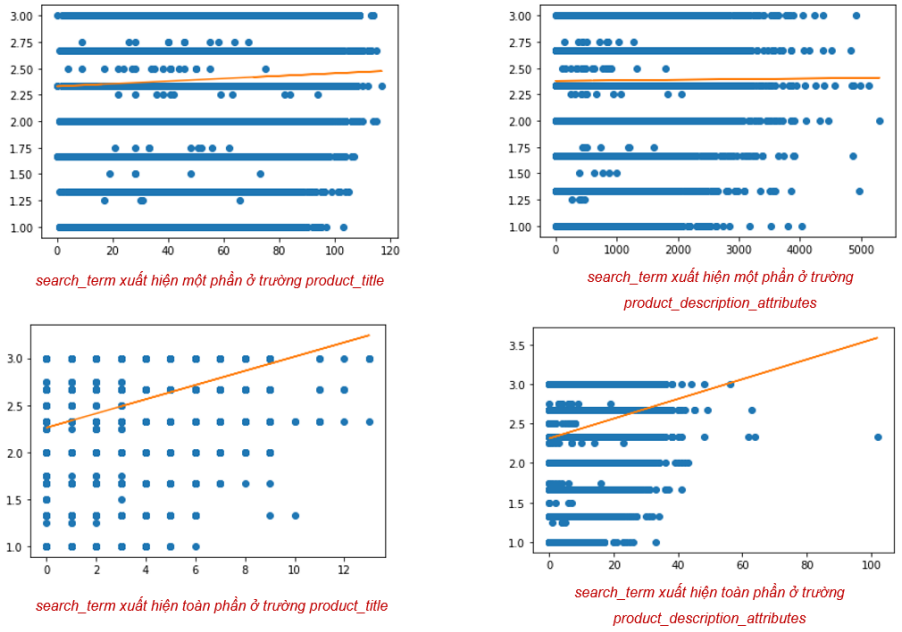
- search_term xuất hiện một phần trong trường product_title.
- search-term xuất hiện một phần trong trường product_description_attributes.
- search-term xuất hiện (tất cả) trong trường product_title.
- search-term xuất hiện (tất cả) trong trường product_description_attributes.

Đặc trưng cuối cùng là từ việc tính toán độ giống nhau của các cặp câu trong search_term và product_title, đặc trưng này sẽ được tính bằng thư viện Levanshtein.

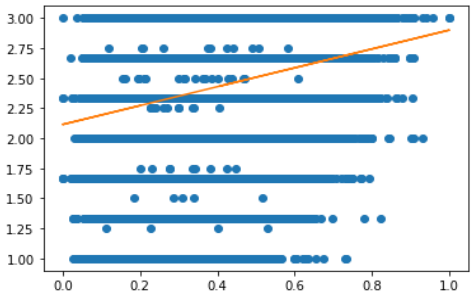
Các đặc trưng sau khi được rút trích sẽ được đánh giá lại bằng hàm tương quan một lần nữa. Nhóm sẽ xây dựng một biểu đồ với trục x tương ứng là độ dài của các trường được xét đến, trục y là điểm search_relevance, một đường regression sẽ được vẽ để có thể quan sát và cho kết quả trực quan như các Hình 6,7,8. Nếu như đường regression càng lệch ra khỏi các điểm trên đồ thị thì ta kết luận nó càng có ảnh hưởng tới điểm relevance và nếu nó càng trùng với các điểm thì ta sẽ kết luận ngược lại.



Hình 6. *Độ tương quan của các đặc trưng 1,2,3*



Hình 7. *Độ tương quan của các đặc trưng 4,5,6,7*



Hình 8. *Độ tương quan của đặc trưng độ giống nhau của các cặp câu trong search_term và product_title (Levanshtein score)*

Dựa vào việc tính toán các tương quan nhóm lựa chọn lại các đặc trưng có ý nghĩa sau: độ dài của search_term, search_term xuất hiện một phần trong trường product_title, search_term xuất hiện (tất cả) trong trường product_title, search-term xuất hiện (tất cả) trong trường product_description_attributes và độ giống nhau của các cặp câu trong search_term và product_title (Levanshtein score). Tổng cộng có 5 đặc trưng.

2.4 Huấn luyện mô hình

Do đây là bài toán hồi quy nên nhóm thực hiện 2 mô hình quen thuộc là Random Forest Regressor và Gradient Boosting Regressor (đều được hỗ trợ bởi thư viện pyspark), đồng thời thực hiện phương pháp kết hợp ensemble của 2 mô hình để cho ra kết quả relevance score cuối cùng là tốt nhất (bằng cách lấy trung bình kết quả của 2 mô hình lại với nhau). Kết quả đạt được như bảng 2.

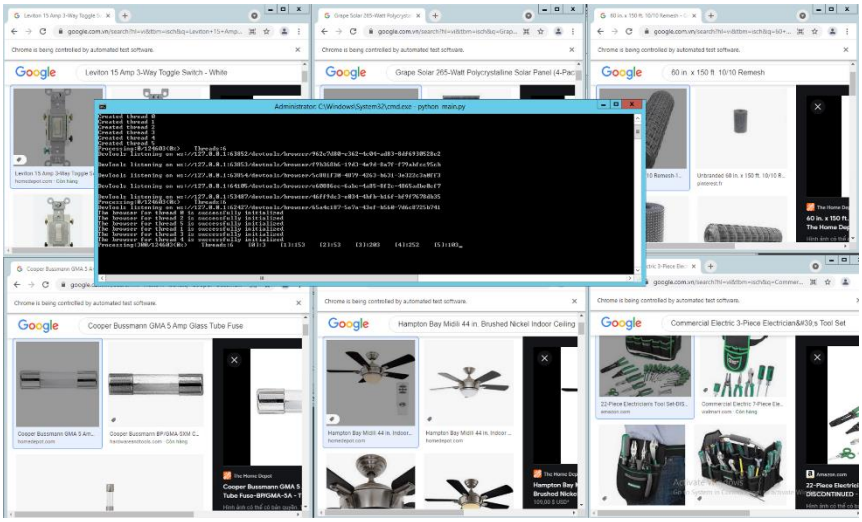
Bảng 2. *Kết quả các mô hình dự đoán relevance score*

Mô hình	**Kết quả (độ đo: RMSE)**
Random Forest Regressor	0.49047
Gradient Boosting Regressor	0.49433
Ensemble 2 mô hình	**0.49015** **(top-1 cuộc thi là 0.43192)**

Mô hình ensemble này sẽ được lưu trữ lại các trọng số (weight) và được đưa vào phần backend của hệ thống tìm kiếm nhằm dự đoán relevant score dựa vào search_term của người dùng. Một điểm lưu ý mô hình và cách triển khai bài toán của cuộc thi cần điều chỉnh bổ sung một vài phần để phục vụ cho hệ thống tìm kiếm mong muốn.

2.5 Xây dựng cơ sở dữ liệu hình ảnh

Để tăng phần trực quan, sinh động khi hiển thị sản phẩm trên website, chúng tôi đã xây dựng cơ sở dữ liệu hình ảnh về sản phẩm. Các hình ảnh được thu thập dựa vào kho hình ảnh của Google, tìm kiếm dựa trên thuộc tính tên sản phẩm product_title và lưu trữ định danh bằng product_uid (với khoảng 54.000 ảnh). Chúng tôi đã kết hợp Python và Selenium để xây dựng công cụ tự động tìm kiếm và tải xuống ảnh phù hợp. Công cụ đã thực hiện tự động hóa các thao tác điền từ khóa tìm kiếm, gửi yêu cầu tìm kiếm hình ảnh, xác định vị trí hình ảnh đầu tiên trong danh sách các ảnh trả về, lưu hình ảnh xuống cơ sở dữ liệu.



Hình 9. *Quá trình thu thập hình ảnh sản phẩm*

3 Xây dựng hệ thống tìm kiếm

Trong phần này chúng tôi trình bày về việc hiện thực hóa mô hình bài toán khuyến nghị sản phẩm vào thực tế. Quá trình hiện thực được mô tả thành 2 giai đoạn. Giai đoạn 1 là giai đoạn huấn luyện mô hình. Kết thúc giai đoạn 1 là một mô hình (tập hợp các trọng số đã được huấn luyện/ điều chỉnh) sẽ là tiền đề hệ thống tìm kiếm ứng dụng. Thông tin chi tiết về hệ thống tìm kiếm như sau:

3.1 Kiến trúc tổng quan

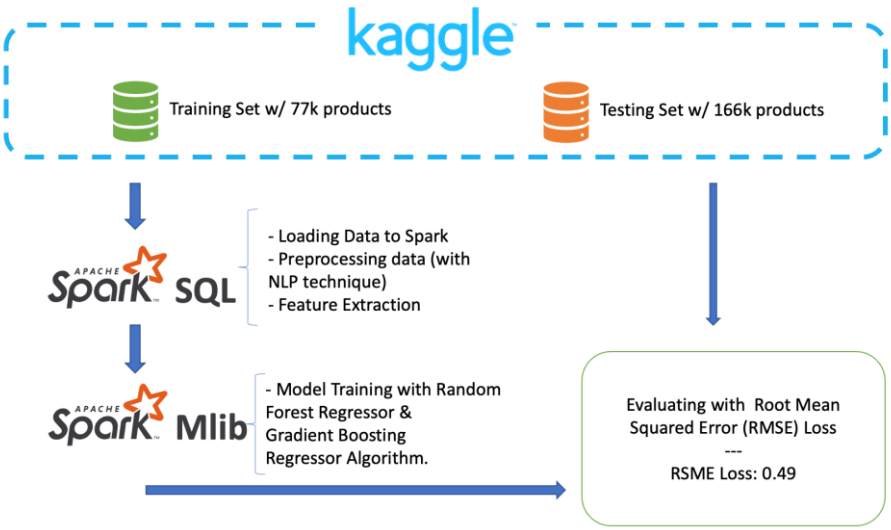
Giai đoạn 1: Huấn luyện mô hình

Với dữ liệu đã được công bố từ Kaggle, tập dữ liệu huấn luyện được tải vào các dataframe của Apache Spark. Dữ liệu huấn luyện này được xử lý thông qua Spark gồm các giai đoạn sau, và minh hoạ qua hình 10.

- Tiền xử lý dữ liệu: Vì dữ liệu là ngôn ngữ tiếng Anh, nên trước khi xử lí cần đưa dữ liệu qua các bước Stemming và Lemmatization. Mục tiêu của bước này là gom nhóm các biến thể khác nhau của từ lại (giảm các biến thể).
- Trích xuất đặc trưng: biến đổi các từ ngữ thành dạng các trọng số phù hợp để chuẩn bị cho bước huấn luyện. Chi tiết tại bước này đã được trình bày tại chương xây dựng mô hình dự đoán nêu trên.
- Huấn luyện mô hình: Sử dụng thư viện Spark Mlib để xây dựng mô hình khuyến

nghị, sử dụng lần lượt giải thuật Forest Regressor, Gradient Boosting Regressor và giải thuật kết hợp hai phương pháp trên. Hàm loss được sử dụng là hàm Root Mean Squared Error (RMSE).

Sau khi thực hiện huấn luyện và kiểm thử điểm RMSE trên tập test, thu được kết quả tốt nhất ở phương pháp kết hợp. Mô hình sau đó được lưu lại để hệ thống tìm kiếm.



Hình 10. *Minh hoạ luồng hoạt động của giai đoạn huấn luyện mô hình*

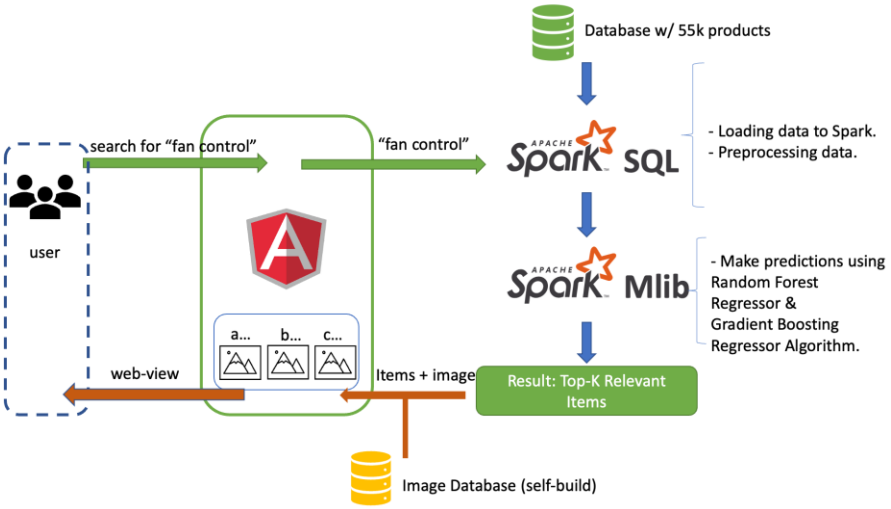
Giai đoạn 2: Hệ thống tìm kiếm

Với mô hình huấn luyện đã được chuẩn bị ở giai đoạn 1, chúng tôi trình bày về cách hiện thực hệ thống tìm kiếm như sau, và minh hoạ qua hình 12:

- Phía người dùng: người dùng truy cập vào giao diện tìm kiếm, điền từ khóa cần tìm kiếm vào "search box". Từ khoá sau đó được frontend gọi vào API của hệ thống tìm kiếm.
- Với kỹ thuật tạo tập dữ liệu input: một dataframe, tương tự như giai đoạn 1 được tạo ra và nạp thông tin của hơn 55 ngàn sản phẩm vào dataframe. Cột từ khóa (nhận từ người dùng) được thêm vào sau đó (minh họa bởi hình 11).
- Cả dataframe được xử lý theo quy trình tiền xử lý và rút trích đặc trưng tương tự như giai đoạn 1. Sau đó dataframe được đưa qua mô hình khuyến nghị (đã được lưu từ trước) và tiến hành tính toán điểm số search_relevant.
- Cuối cùng, dựa trên đầu ra điểm số search_relevant của mô hình, chọn lựa ra TOP-K sản phẩm và gửi về cho API Frontend, từ đó hiển thị kết quả cho người dùng.

	product_uid	product_title	product_description_attributes	search_term
0	100001	simpson strongti 12gaug angl	angl make joint stronger also provid consist s...	fan control
1	100002	behr premium textur deckov 1gal sc141 tugboat ...	behr premium textur deckov innov solid color c...	fan control
2	100005	delta vero 1handl shower onli faucet trim kit ...	updat bathroom delta vero singlehandl shower f...	fan control
3	100006	whirlpool 1 9 foot over the rang convect micro...	achiev delici result almost effortless whirlpo...	fan control
4	100007	lithonia light quantum 2light black led emerg ...	quantum adjust 2light led black emerg light un...	fan control
...	...	...	...	...
54676	206638	atlant windowpan 576 cd or 192 dvd bluray or g...	atlant inc 94835722 uniqu design maximum mediu...	fan control
54677	206639	philip 40watt halogen r20 flood light bulb 12pack	philip energi advantag lamp use le energi main...	fan control
54678	206641	schlage camelot inact age bronz handleset with...	schlage camelot inact age onz handleset leftha...	fan control
54679	206648	plastec 11 inch 24 inch rose garden wall decor...	rose garden inspir popular earli 20th centuri ...	fan control
54680	206650	lichtenberg pool blue no 918 millenni ryan hea...	918 millenni ryan heather textur semish curtai...	fan control

Hình 11. *Ví dụ minh họa về tập test input giả lập được xây dựng dựa trên từ tìm kiếm của người dùng.*

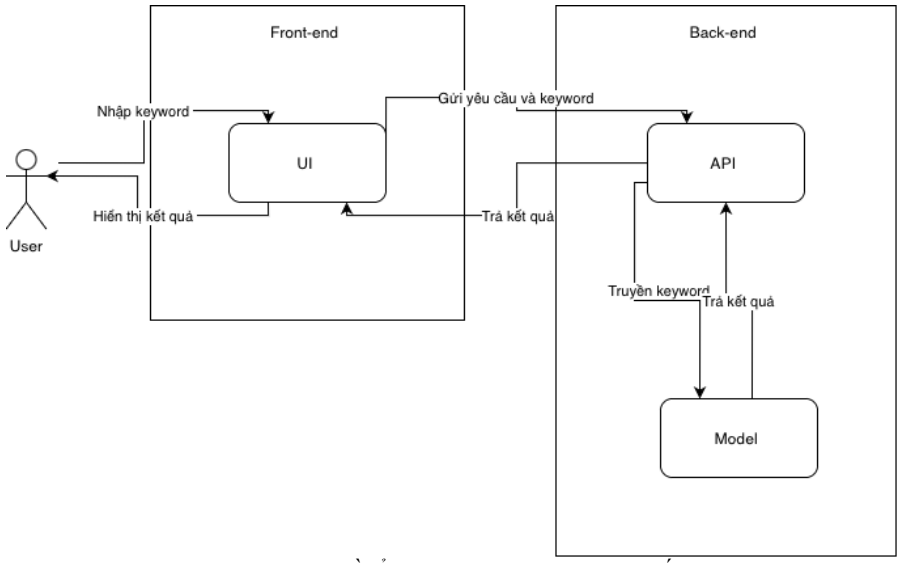


Hình 12. *Minh hoạ luồng hoạt động của hệ thống tìm kiếm*

3.2 Xây dựng ứng dụng web

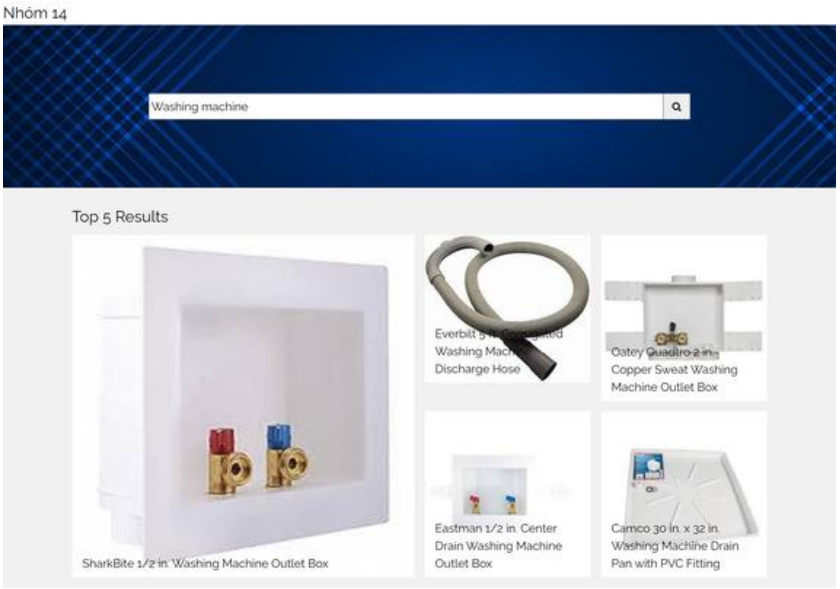
Ứng dụng web được xây dựng dựa trên mô hình đã được huấn luyện nhằm đưa ra gợi ý cho người dùng tương ứng với từ khoá mà người dùng nhập vào.

Sau khi thực hiện tính điểm liên quan của sản phẩm so với từ khoá mà người dùng nhập vào, hệ thống sẽ trả về kết quả là các sản phẩm liên quan. Các sản phẩm được hiển thị với các thông tin như tên sản phẩm, hình ảnh sản phẩm và điểm liên quan của sản phẩm so với từ khoá.

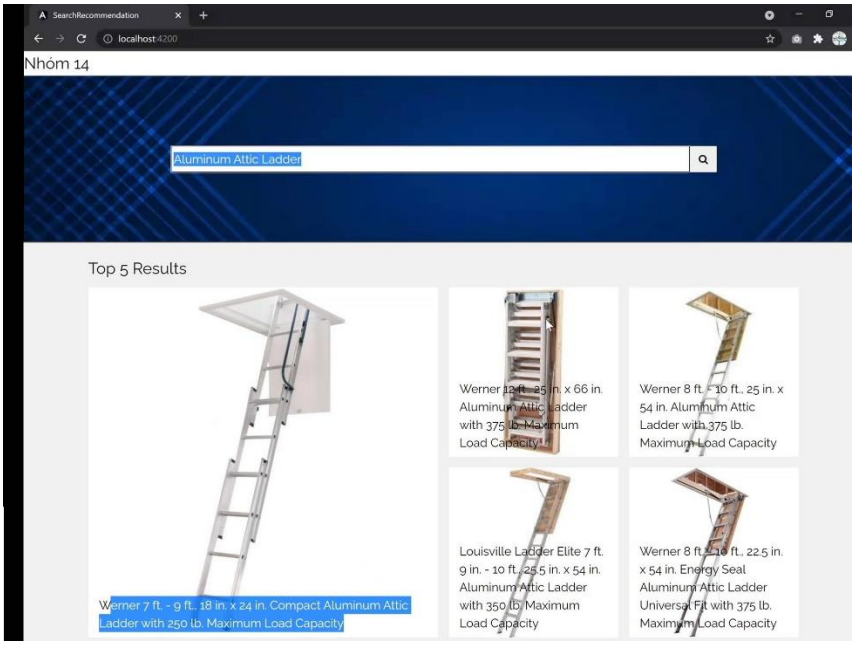


Hình 13. *Sơ đồ tổng quan user story của hệ thống*

Cụ thể, khi người dùng nhập từ khoá và tiến hành tìm kiếm thì ứng dụng sẽ gửi một yêu cầu tới Back-end để thực hiện tính toán. Sau đó, Back-end sẽ trả về tất cả các sản phẩm liên quan với đầy đủ thông tin sản phẩm và điểm liên quan đã tính toán được. Dựa vào kết quả đó Front-end sẽ lấy ra top n sản phẩm để hiển thị lên giao diện cho người dùng. Trong đó, n có thể thay đổi tuỳ theo mong muốn, trong ứng dụng này n = 5.

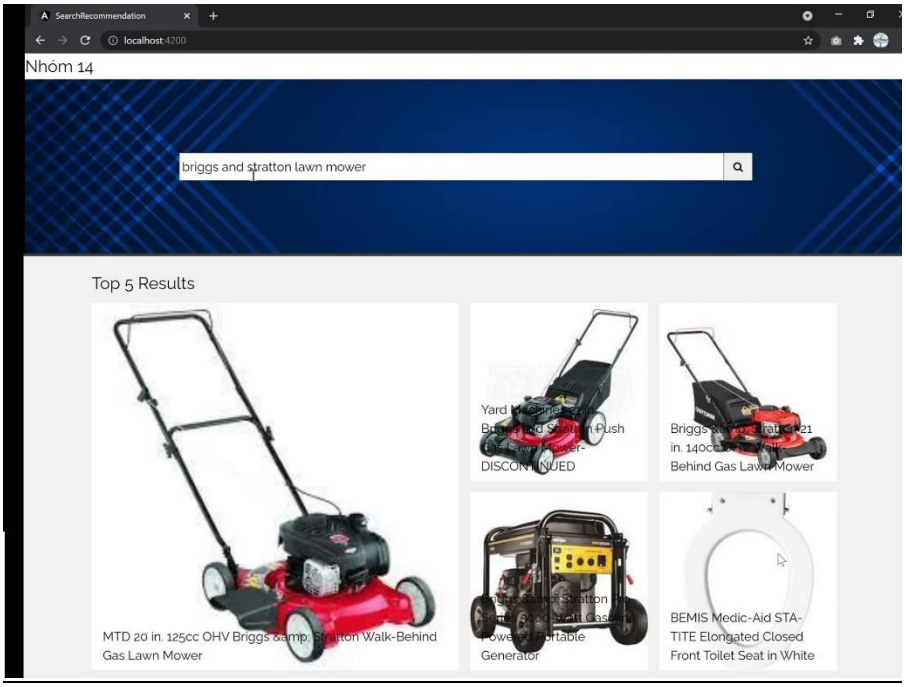


Hình 14. *Giao diện ứng dụng web*



Hình 15. *Giao diện ứng dụng web*

Hình 14 thể hiện giao diện của ứng dụng và các sản phẩm mà hệ thống gợi ý khi người dùng tìm kiếm với từ khoá "Washing machine". Giao diện gồm một ô input để người dùng nhập từ khóa cần tìm, 5 card để hiển thị sản phẩm (gồm 1 card lớn cho sản phẩm có điểm liên quan cao nhất và 4 card nhỏ cho 4 sản phẩm có điểm cao tiếp theo) và khi hover vào hình ảnh của sản phẩm, điểm liên quan sẽ được hiển thị dưới dạng tooltip. Sau khi người dùng bấm tìm kiếm thì hệ thống sẽ xử lý và trả kết quả về trong khoảng trung bình là 3 – 5 giây.



Hình 16. *Giao diện ứng dụng web*

4 Nhận xét và kết luận

Nhóm chúng tôi đã xây dựng được hệ thống tìm kiếm và gợi ý sản phẩm liên quan nhất cho người dùng với input đầu vào từ tìm kiếm của người dùng và ouput kết quả trả về là danh sách các sản phẩm liên quan nhất.

Nhóm đã vận dụng được kiến thức đã học từ môn Xử lý dữ liệu lớn và đặc biệt là phần pyspark sql và pyspark machine learning để áp dụng vào hệ thống. Trước tiên là vận dụng các kỹ thuật phân tích dữ liệu và xử lý ngôn ngữ tự nhiên để xử lý làm sạch, rút trích đặc trưng đối với dữ liệu dạng text. Nhóm sử dụng kết hợp 2 mô hình regression quen thuộc là Random Forest Regressor và Gradient Boosting Regressor với độ chính xác trên tập test của cuộc thi đạt **RMSE** là **0.49015**. Kế tiếp, hệ thống xử lý và trả về kết quả cho người dùng trong thời gian 3 – 5 giây / truy vấn.

Nhóm nhận định hệ thống còn mặt hạn chế và có thể cải thiện thêm như sử dụng các các rút trích đặc trưng từ text phức tạp hơn như tf-idf, bag of words, word2vec và đặc biệt là các mô hình học sâu, tuy nhiên sẽ phải đánh đổi về thời gian xử lý. Ngoài ra, cần đánh giá thêm về hiệu suất hệ thống với một lượng truy vấn của nhiều người dùng cùng lúc thay vì chỉ một truy vấn như hiện tại.

Tài liệu tham khảo

1. Đỗ Trọng Hợp, Big Data lecturers and slides.
2. A. Géron, Hands-on Machine Learning With Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. Sebastopol, CA, USA: O'Reilly, 2017
3. Spark Apache. https://spark.apache.org/docs/latest/ml-classification-regression
4. Home Depot Product Search Relevance challenge. https://www.kaggle.com/c/home-depot-product-search-relevance