

Trường Đại học Bách Khoa Hà Nội



Bài tập lớn môn học **Đề tài: Tích hợp dữ liệu web**

Học phần: Tích hợp dữ liệu (IT5420)

Giảng viên hướng dẫn: TS. Đỗ Bá Lâm

Nhóm 09:

Nguyễn Văn Lương	20173249
Nguyễn Đình Mạnh	20173255
Nguyễn Văn Long	20173244
Lê Trung Hoàng Long	20173242

Hà Nội, 31 tháng 5 năm 2021

Trường Đại học Bách Khoa Hà Nội	1
Mở đầu	3
1. Tổng quan	3
1.1. Tóm tắt các nội dung trong bài toán	3
1.2. Các công nghệ sử dụng	4
2. Thu thập dữ liệu	6
2.1. Phân tích bài toán	6
2.2. Trường dữ liệu trích xuất	6
2.3. Lập trình và cài đặt	7
2.4. Data sau khi crawl	9
3. Tiền xử lý	10
3.1. Loại bỏ trùng lặp	10
3.2. Chuẩn hóa	10
3.3. Chuẩn bị dữ liệu matching	10
4. Đối sánh lược đồ	12
4.1 Dựa trên tên trường thuộc tính	12
4.2 Dựa trên bản ghi (luật cú pháp)	13
5. Đối sánh dữ liệu	17
5.1. Xử lý trùng lặp	17
5.2 Tổng hợp dữ liệu	19
5.3. Đối sánh dữ liệu lần 1	20
5.4. Tối ưu trọng số	22
6. Phân tích dữ liệu	23
6.1. Tổng quan về dữ liệu	24
6.2. Thống kê dữ liệu thiếu	24
6.3 Môi quan hệ giữa giá và khu vực	24
6.4. Môi quan hệ giữa giá và số sao	25
6.5. Môi quan hệ giữa giá và điểm đánh giá	25
6.6. Độ tương quan giữa các thuộc tính	26
7. Xếp hạng khách sạn	26
8. Đề xuất khách sạn	27
9. Xây dựng website khai thác dữ liệu	29
10. Đánh giá và kết luận	31
11. Tài liệu tham khảo	31

Mở đầu

Tích hợp dữ liệu (Data integration) bao gồm việc kết hợp dữ liệu không đồng nhất trong các nguồn khác nhau vào một lược đồ duy nhất và có thể truy vấn, cung cấp cho người dùng một cái nhìn thống nhất về chúng.

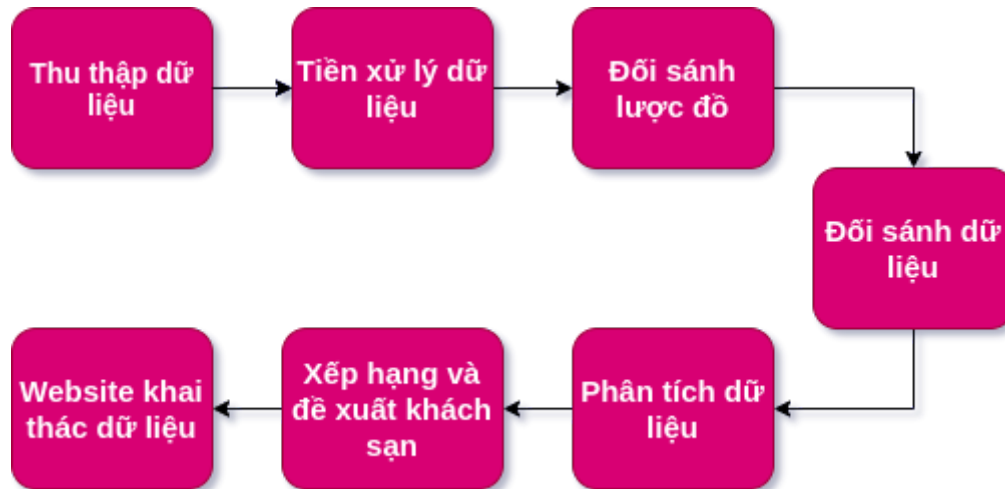
Tích hợp dữ liệu được sử dụng với tần số ngày càng nhiều khi mà khối lượng và nhu cầu chia sẻ dữ liệu hiện nay rất lớn. Để đảm bảo việc trao đổi dữ liệu trong hệ thống được hiệu quả hoặc xử lý các công việc tiếp theo theo các luồng công việc định trước như: phân tích, so sánh, thống kê, báo cáo,... vai trò của nó đang ngày một quan trọng với sự phát triển không ngừng của thông tin hiện nay.

Trong bài toán này hướng tới việc tích hợp dữ liệu khách sạn từ các nguồn thông tin được crawl trên các trang web khác nhau. Đây là một bài toán rất hay hiện tại vì như chúng ta đã biết, diễn biến dịch bệnh covid ở Việt Nam đang rất phức tạp với sự bùng phát số ca mắc và các vùng dịch bệnh mới mỗi ngày khiến nhiều địa điểm du lịch nhận được yêu cầu hủy phòng, kéo theo sự sụt giảm công suất của toàn thị trường. Vì vậy ngoài việc chính trong bài toán là tích hợp dữ liệu thì nhóm cũng sẽ có một vài phân tích, đánh giá xếp hạng, cũng như xây dựng một hệ thống đề xuất các khách sạn phù hợp.

1. Tổng quan

1.1. Tóm tắt các nội dung trong bài toán

Crawl dữ liệu khách sạn ở các OTA nhằm xây dựng website chung thuận lợi cho việc tìm kiếm và so sánh của khách hàng.



Biểu đồ luồng xử lý của bài toán

Biểu đồ trên đây thể hiện luồng xử lý, cũng như các công việc trong bài toán, về cơ bản nó gồm các bước:

- Thu thập dữ liệu: dữ liệu thu thập bằng phương pháp crawl dữ liệu trên các website đặt phòng khách sạn, đầu ra sau khi crawl là các tập tin thô ở dạng json lưu thông tin của các khách sạn khác nhau từ nhiều nguồn khác nhau.
- Tiền xử lý: sau bước đầu ta có tập dữ liệu thô sơ, do đó ở bước này sẽ có những phương pháp tiền xử lý cơ bản để làm sạch dữ liệu, loại bỏ trùng lặp và sau đó kết hợp chuẩn hóa dữ liệu để đưa về các đơn vị thống nhất cho các nguồn.
- Đối sánh lược đồ: tìm ra lược đồ chung bằng phương pháp đối sánh các lược đồ
- Đối sánh dữ liệu: đối sánh các bản ghi giữa các bộ dữ liệu với nhau nhằm tìm ra các bản ghi giống nhau giữa các tập.
- Phân tích dữ liệu: cơ bản các nguồn dữ liệu khi đến được bước này đã được làm sạch và đồng nhất về đơn vị. Ở bước này ta sẽ đưa ra những phân tích và trực quan hóa dữ liệu này.
- Xếp hạng và đề xuất khách sạn: xây dựng một hệ gợi ý - recommendation system để đề xuất các khách sạn tương ứng khi ta chọn vào một khách sạn.
- Xây dựng website khai thác dữ liệu: xây dựng một trang web đơn giản để biểu diễn trực quan hơn các dữ liệu.

1.2. Các công nghệ sử dụng

Scrapy: để thu thập dữ liệu, nhóm sử dụng scrapy: Scrapy là một framework thu thập dữ liệu web ở mức cao và nhanh chóng, được sử dụng để thu thập dữ liệu các trang web và trích xuất dữ liệu có cấu trúc từ các trang của chúng. Nó có thể được sử dụng cho nhiều mục đích khác nhau, từ khai thác dữ liệu đến giám sát và kiểm tra tự động.

Hiệu suất của nó rất nhanh và nó là một trong những thư viện mạnh nhất hiện có. Một trong những ưu điểm chính của Scrapy là nó được xây dựng dựa trên Twisted, một khung mạng không đồng bộ, có nghĩa là Scrapy sử dụng cơ chế không chặn trong khi gửi yêu cầu đến người dùng. Scrapy có nhiều ưu điểm khác để được lựa chọn cho việc thu thập dữ liệu là :

- Scrapy có hỗ trợ tích hợp để trích xuất dữ liệu từ các nguồn HTML bằng cách sử dụng biểu thức XPath và biểu thức CSS.
- Nó là một thư viện di động, tức là (được viết bằng Python và chạy trên Linux, Windows, Mac và BSD)
- Dễ dàng mở rộng.
- Nhanh hơn các thư viện thu thập hiện có khác. Nó có thể giải nén các trang web với tốc độ nhanh hơn 20 lần so với các công cụ khác. Nó tiêu tốn ít bộ nhớ và CPU hơn rất nhiều.
- Xây dựng nhanh chóng một ứng dụng mạnh mẽ và linh hoạt với nhiều chức năng.

Selenium: do các trang website có cả các dynamic page mà scrapy không xử lý được nên nhóm kết hợp với selenium để crawl dữ liệu. Selenium là một trong những công cụ kiểm thử phần mềm tự động mã nguồn mở mạnh nhất hiện nay cho việc kiểm thử ứng dụng Web. Selenium script có thể chạy được trên hầu hết các trình duyệt như IE, Mozilla FireFox, Chrome, Safari, Opera; và hầu hết các hệ điều hành như Windows, Mac, Linux. Về cơ bản mà nói, quá trình scraping cũng tương tự như quá trình kiểm thử ứng dụng tự động bởi chúng đều thực hiện một chuỗi thao tác tương tác với các trang web một cách tự động và liên tục. Bởi vậy, Selenium thường xuyên được sử dụng nhất là khi cần thu thập từ các trang web SPA - thứ mà khó có thể thu thập được dữ liệu từ nó nếu như phần mã JavaScript của chúng không được thực thi.

Python + Jupyter notebook: nhóm sử dụng ngôn ngữ python kết hợp jupyter notebook để thực hiện phần lớn các tác vụ thao tác và xử lý với dữ liệu. Python là một ngôn ngữ rất phổ biến, nổi tiếng vì cú pháp đơn giản, dễ học, bộ thư viện hỗ trợ rộng lớn và đặc biệt là ngôn ngữ hỗ trợ rất mạnh để làm việc với dữ liệu như phân tích, trực quan hóa, machine learning,... với Jupyter notebook, đây là một công cụ mã nguồn mở miễn phí với mục đích nhắm đến khoa học dữ liệu và giáo dục. Công cụ này cho phép bạn đưa cả code Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức... vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác

trên đó. Các file notebook này có thể được chia sẻ với mọi người và có thể thực hiện lại các công đoạn một cách nhanh chóng và chính xác như những gì tác giả của file này tạo ra.

2. Thu thập dữ liệu

2.1. Phân tích bài toán

Thu thập dữ liệu là bước đầu tiên và cũng rất quan trọng trong bài toán lần này. Để đảm bảo tính khách quan các cho bước ở sau, dữ liệu cần thu thập được từ nhiều nguồn, và có kích thước đủ lớn.

Các trang web đặt khách sạn thường là những trang web rất lớn, phức tạp và không chỉ có thông tin về các khách sạn mà còn chứa thông tin về các dịch vụ du lịch nói chung. Dữ liệu về khách sạn thường được chia theo các thành phố, có thành phố có ít khách sạn chỉ từ khoảng một vài trăm khách sạn, có thành phố lại có nhiều khách sạn lên đến hàng nghìn khách sạn ví dụ như Hà Nội, Hồ Chí Minh, Đà Lạt, Vũng Tàu, Quảng Ninh, Nha Trang, Phú Quốc là những thành phố nổi tiếng về du lịch ở Việt Nam. Nên để đảm bảo số lượng khách sạn ở mỗi thành phố ở mức tương đối, không quá ít thì ta sẽ chỉ crawl ở một số thành phố có lượng khách sạn cao.

Trong mỗi khách sạn lại có rất nhiều trường dữ liệu cần quan tâm, ngoài những thứ ta thấy ngay như địa điểm, số sao, đánh giá của khách hàng,... thì có những trường dữ liệu phức tạp hơn trong việc thu thập như giá phòng, các tiện nghi, các địa điểm gần kề,... và thông tin các khách sạn ở trang web này có thể liên kết đến trang web đặt phòng khách sạn khác. Ngoài ra giá của khách sạn phụ thuộc vào nhiều yếu tố: ngày check in, ngày check out, số lượng khách lưu trú, nên ta cần hạn chế lượng thông tin này lại. Ở bài toán này sẽ lấy thông tin đặt phòng dành cho 1 người lớn + ở trong một đêm.

Các website cho phép đặt phòng khách sạn ở riêng tại Việt Nam có rất nhiều nhưng chỉ một số trang web tương đối lớn với mạng lưới khách sạn rộng khắp, đủ thông tin để cho phép ta crawl, nên trong bài toán lần này, nhóm sử dụng 4 nguồn dữ liệu từ 4 website về đặt phòng khách sạn, du lịch có thể nói là lớn và uy tín nhất ở Việt Nam hiện nay bao gồm:

- booking.com
- expedia.com.vn
- agoda.com
- ebooking.com

2.2. Trường dữ liệu trích xuất

Trong 4 tên miền trên, các thông tin đăng về khách sạn đều có chứa các trường

thông tin gần như giống nhau và có thêm một vài trường khác nhau giữa các website. Để tối đa hóa nguồn dữ liệu có được phục vụ cho quá trình xử lý phân tích dữ liệu, 14 trường dữ liệu sẽ được trích xuất tương ứng với các thông tin cơ bản của một khách sạn. Các trường được trích xuất bao gồm:

- **City:** Tên thành phố nơi khách sạn được đặt.
- **Hotel name:** Tên của khách sạn.
- **Url:** Đường dẫn đến trang thông tin chi tiết của khách sạn.
- **Address:** Địa chỉ của khách sạn (thường có số đường, quận/huyện, thành phố)
- **Longitude:** Kinh độ của khách sạn
- **Latitude:** Vĩ độ của khách sạn
- **Stars:** Số sao của khách sạn (cao nhất là 5 sao)
- **Price:** Giá phòng rẻ nhất của khách sạn
- **Rating:** Điểm rating trung bình của các du khách đã từng đặt và trải nghiệm ở khách sạn
- **Number of reviews:** Số lượng reviews dành cho khách sạn (reviews là những bình luận của khách hàng)
- **Reviews:** Một vài dòng bình luận thu thập được dành cho khách sạn
- **Facilities:** Các cơ sở vật chất, tiện nghi của khách sạn
- **Description:** Dòng mô tả về khách sạn
- **Nearby places:** Các địa điểm du lịch, nhà hàng, ăn uống, sân bay,... xung quanh khách sạn.

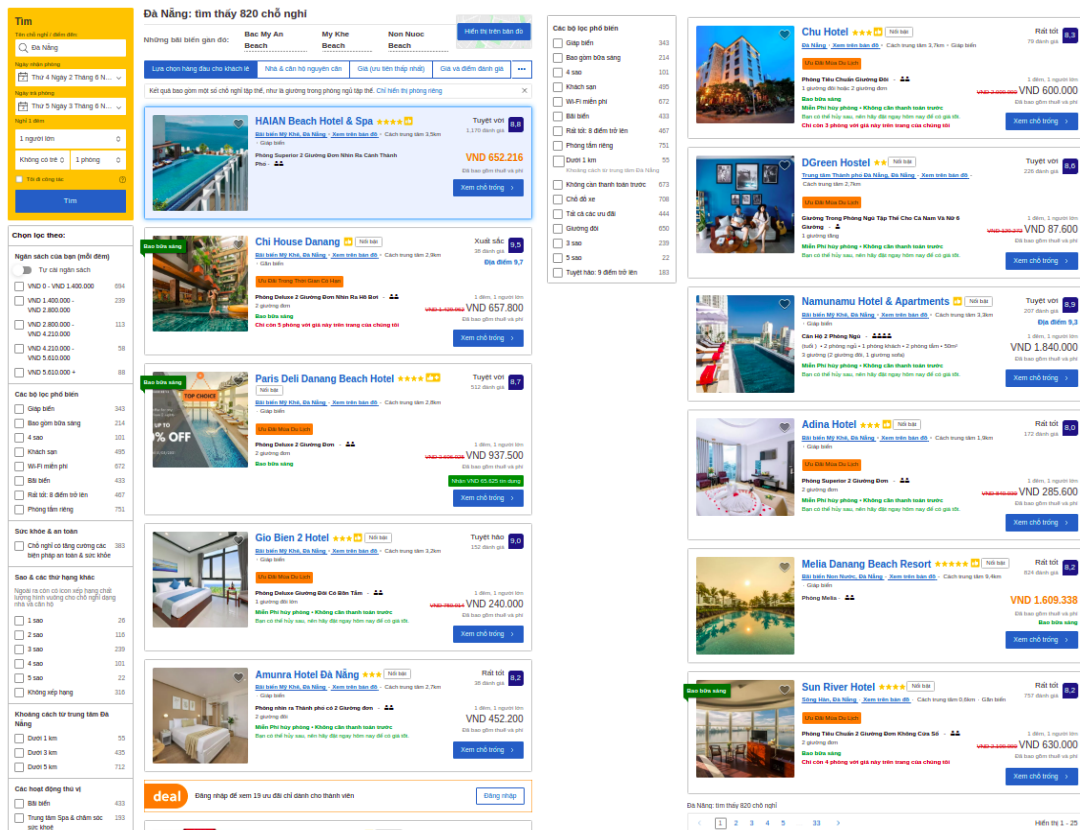
Như đã phân tích ở phần trên, ta sẽ chỉ crawl ở mỗi tên miền thông tin các khách sạn của 8 thành phố lớn và nổi tiếng về du lịch tại Việt Nam là: Hà Nội, Hồ Chí Minh, Đà Lạt, Đà Nẵng, Nha Trang, Phú Quốc, Vũng Tàu, Quảng Ninh. Và thông tin đặt phòng là: một người lớn + thuê trong một ngày một đêm

2.3. Lập trình và cài đặt

Nhóm sử dụng framework scrapy và selenium dành cho ngôn ngữ python, và phần mềm google chrome version 91 để crawl dữ liệu.

Về thông số cài đặt scrapy và quá trình crawl trên các trang website. Nhận thấy các website này có cấu trúc tương đối giống nhau. Cơ bản sẽ có 2 loại trang thông tin chính.

- Trang web chứa danh sách các khách sạn:



Chứa danh sách khách sạn theo thành phố, ngoài ra thông tin đặt phòng về ngày check in, check out, đặt cho bao người cũng được cài đặt ở trang này và thông tin đặt phòng này sẽ ảnh hưởng đến thông tin khách sạn.

Thông tin khách sạn được liệt kê ở các trang web này bao gồm một vài trường thông tin chính như: nguồn ảnh, tên khách sạn, rating, số sao, giá khởi điểm và đặc biệt là thông tin về đường url dẫn đến trang web chứa thông tin chi tiết của khách sạn.

Trang web dạng này là loại dynamic web page tức là nội dung động và hay thay đổi, mà scrapy không hỗ trợ crawl với dynamic content nên gây rất nhiều khó khăn trong việc thu thập, một phương pháp nhóm áp dụng để làm việc với trang web động là sử dụng framework selenium: với cả 4 website, nhóm sử dụng selenium để thao tác trên trang web dạng này với các thao tác như cuộn xuống để tải thông tin, chuyển trang; lấy dữ liệu về tên thành phố, giá khởi điểm và đặc biệt là các đường dẫn url đến các trang thông tin chi tiết của khách sạn bằng phương thức:

find_elements_by_css_selector(). Các url lấy được này sẽ được đưa vào hàng đợi để tiến hành nạp và thu thập dữ liệu khách sạn.

- Trang web chứa thông tin chi tiết của mỗi khách sạn: chứa các thông tin chi tiết của mỗi khách sạn, mỗi khách sạn có một trang chi tiết như vậy chứa tất cả thông tin mà ta cần về khách sạn. Nhóm sẽ trích xuất phần lớn trường dữ liệu của khách sạn tại các trang web này bằng cách tìm css selector chứa trường dữ liệu bằng bộ công cụ phát triển web tích hợp của google chrome và sử dụng phương thức của scrapy: response.css(selector) để lấy dữ liệu.

Mỗi website nhóm sẽ sử dụng một con spider tương ứng với một lớp Spider riêng để crawl dữ liệu. Khởi tạo seed urls là các url dẫn đến các trang web chứa danh sách khách sạn của mỗi thành phố, trong bài tập lần này là 8 urls tương ứng với 8 trang web chứa danh sách hotel của 8 thành phố.

2.4. Data sau khi crawl

Sau khi thực hiện việc crawl dữ liệu từ trang web dao động trong khoảng thời gian từ 30 - 40 phút chạy để hoàn thành việc lấy hết dữ liệu thì ta có được bộ dữ liệu gốc ban đầu được lưu trong 4 file json với những thông kê như sau:

Ebooking: 9031 records

url	img	name	address	rating	facilities	price	star	n_review	nearby	lat	lot	city
https://ho	https://exj	Khách sạn 65 Nguyễn		8.6	Áp dụng	204622	3.5	40	Tên khu v	21.03282	105.8545	Hà Nội
https://ho	https://exj	Khách sạn 77 Hàng L		8.8	Chỗ ở	306011	3	110	Tên khu v	21.03682	105.8486	Hà Nội
https://ho	https://exj	Khách sạn 15 Nguyễn T		8.8	Áp dụng	433900	3	120	Tên khu v	21.03387	105.8525	Hà Nội
https://ho	https://exj	O'Gallery F 122 Hàng		9.4	Áp dụng	505794	4	859	Tên khu v	21.02966	105.8458	Hà Nội
https://ho	https://exj	Khách sạn 29 Hàng N		9.6	Áp dụng	319376	3.5	96	Tên khu v	21.03165	105.8484	Hà Nội
https://ho	https://exj	Midori Bn 43 Triết		8	Áp dụng	218909	3	2	Tên khu v	21.01639	105.8504	Hà Nội
https://ho	https://exj	Imperial H 44 Hàng H		9.4	Áp dụng	603266	4	166	Tên khu v	21.03113	105.85	Hà Nội
https://ho	https://exj	Golden Ro 17 Hàng C		9	Áp dụng	289420	3	107	Tên khu v	21.03587	105.8472	Hà Nội
https://ho	https://exj	Little Hanc 1 Yên Th		9.2	Áp dụng	396570	3.5	184	Tên khu v	21.03198	105.8477	Hà Nội

Expedia: 4068 records

city	hotel name	url	address	stars	price	rating	n_reviews	facilities	description	nearby place	images	lat	lot
Hà Nội	Chân	Wink Hotel	https://wv 75 Nguyễn	4 out of 5	807.316	4,5/5	Tùy 2 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.79128	106.7005
Hà Nội	Chân	Khách sạn	https://wv 76 Lê Lai	5 out of 5	1.488.000	4,1/5	Rất 994 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.77058	106.6952
Hà Nội	Chân	Khách sạn	https://wv 261C Nguyễn	4 out of 5	1.052.814	5,0/5	Ng 4 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.79754	106.672
Hà Nội	Chân	Airport Sài	https://wv 34 Ng 3 Th	3 out of 5	458.366	4,1/5	Rất 191 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.79575	106.6632
Hà Nội	Chân	Grand Hotel	https://wv 8 Lê Lai	5 out of 5	846.831	4,3/5	Xuất 998 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.77368	106.7056
Hà Nội	Chân	Hotel Nikk	https://wv 235 Nguyễn	5 out of 5	1.386.707	4,6/5	Tùy 1.000 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.76376	106.6823
Hà Nội	Chân	Sofitel Plaza	https://wv 17 Lê Duẩn	5 out of 5	1.632.000	4,4/5	Xuất 659 nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	10.78436	106.7023

Agoda: 9031 records

city	hotel name	url	address	stars	price	rating	n_reviews	image	reviews	facilities	nearby place	lat	lot
Nha Trang	Nhà	Dã Cn	https://wv 7 Sao Bi	5	81.818	9,8	93 Nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	12.27728	109.2005
Nha Trang	Căn	Hà Nội	https://wv 02 Nguyễn	5	445.886	9,6	127 Nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	12.23939	109.1961
Nha Trang	Khách	Sán	https://wv 65/07 Nguyễn	Thiet T	194.567	9,6	1.806 Nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	12.2392	109.1939
Nha Trang	Bán	f b	https://www.agoda.c	0		10,0	44 Nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	12.23901	109.1954
Nha Trang	Khách	Sán	https://wv 87 Bán	4	370.31	9,1	344 Nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	12.24002	109.1921
Nha Trang	LeMore	H	https://wv 33A Tân	Hi	436.506	9,0	899 Nhà	Áp dụng	Áp dụng	Áp dụng	Áp dụng	12.23941	109.1923

Booking: 5495 records

address	city	descriptor	facilities	hotel name	images	n_reviews	nearby place	price	rating	reviews	stars	url	lat	lon
6, Đinh Th	Ha	N	RedDoorz	Ch	RedDoorz	57	B	VND 439.27,3			3	https://wv	21.05213	105.8367
29 Liá...u	Ha	N	Ná±m c	2 há»" b/ 22housing		17	C	VND 1.2159,4			4	https://wv	21.03624	105.8477
112 Cau G	Ha	N	Ná±m c	S	SkyLake Hi	35	N	VND 455.09,1			0	https://wv	21.01826	105.7764
57 Lo Su S	Ha	N	Ná±m tr	N	ha Hanoi La S	1,030	1900 Le 1	VND 303.39,6			3	https://wv	21.00625	105.8387
40 Hang D	Ha	N	Tá»a lá»c	Ch	Antique Ar	897	1900 Le 1	VND 440.09,2			3	https://wv	21.00339	105.7805
156 Á»nh	Ha	N	Ná±m c	S	Homey Ap	37	Diplomat	VND 405.08,4			3	https://wv	21.06485	105.8232
24 ng	Áu 4	Ha	N	Tá»a lá»c	Aiá»u h	56	Há»" Ho	VND 247.45,8			2	https://wv	21.00653	105.7821

Tổng cộng tất cả các bản ghi của cả 4 tập dữ liệu này là 23586

Dựa trên sự đầy đủ của các trường dữ liệu và bản ghi thì ta đưa ra một thứ tự ưu tiên như sau: booking > agoda > ebooking > expedia.

3. Tiền xử lý

3.1. Loại bỏ trùng lặp

Các bộ dữ liệu khi crawl về đều có nhiều bản ghi bị trùng có thể do các lý do: một khách sạn có nhiều loại phòng nên đăng nhiều lần, khu khách sạn liền kề, website thêm khách sạn đó nhiều lần hoặc lỗi trong quá trình crawl điều này sẽ ảnh hưởng tới quá trình matching sau này. Do đó nhóm sẽ sử dụng hàm drop_duplicates của pandas để lọc trùng hoàn toàn, do mỗi khách sạn có thể có nhiều chi nhánh nên việc lọc này sẽ dựa vào 2 thuộc tính là name và city.

Lượng bản ghi còn lại sau khi lọc lần 1: 20576 (loại bỏ 3010 bản ghi)

3.2. Chuẩn hóa

Một số trường dữ liệu cần được chuẩn hóa để đưa về đơn vị thống nhất:

- **“Price”** : Loại bỏ VND, đ , lấy dữ liệu số từ chuỗi này và đưa về dạng số nguyên, fillna = -1..
VD: 1.250.658 VND → 1250658
- **“star”**: thay ‘,’ bằng ‘.’ đưa về dạng float, fillna = -1. Tương tự với rating.
VD: 3,5 → 3.5
- **“number reviews”** : loại bỏ string thừa như ‘nhận xét’..... đưa về dạng số nguyên, fillna = 0
VD: 300 nhận xét → 300

3.3. Chuẩn bị dữ liệu matching

- **“hotel name”**: đối với tên khách sạn nhóm sẽ loại bỏ các substring hay gặp để mang lại hiệu quả cho việc tính tương đồng giữa các tên này. Để làm điều này đầu tiên tên khách sạn đưa về dạng chữ thường không dấu. Sau đó loại bỏ các

substring như: hotel, khách sạn,...Đối với khách sạn có cả tên tiếng Anh và tiếng Việt thì nhóm chỉ giữ lại 1 tên.

Ví dụ kết quả sau khi xử lý:

```
Hanoi Chic Boutique Hotel ==> hanoichicboutique
Diamond Legend Hotel ==> diamondlegend
Khách sạn Imperial Hotel & Spa Hà Nội
(Imperial Hotel & Spa) ==> imperial&spahanoi

Dal Vostro Hotel & Spa ==> dalvostro&spa
O'gallery Premier Hotel & Spa ==> o'gallerypremier&spa
Millennium Hanoi Hotel ==> millenniumhanoi
Khách sạn Daewoo Hà Nội
(Hanoi Daewoo Hotel) ==> daewoohanoi
```

- **“address”**: Với địa chỉ cũng sẽ được xử lý tương tự. Ví dụ: liệt kê một số substring bị loại bỏ là: “street”, “district”, “ward”, “phuong”, “duong”, “quan”, “thanh pho”, “tp”, “ha noi”,....


Ví dụ kết quả sau khi xử lý:

```
12 ngõ Bảo Khánh, Phường Hàng Trống, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 12ngobaokhanhhangtronghoankiem
122 Hàng Bông, Hàng Bông, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 122hangbonghangbonghoankiem
246B Hàng Bông, Cửa Nam, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 246bhangbongcuonanamhoankiem
360 Kim Mã, Ngọc Khánh, Quận Ba Đình, Hà Nội, Việt Nam, 100000 ==> 360kimmangockhanhbadinh
45 Hàng Đồng, Hàng Bồ, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 45hangdonghangbohoankiem
46 Nguyễn Trường Tộ, Trúc Bạch, Quận Ba Đình, Hà Nội, Việt Nam, 100000 ==> 46nguyentruongtotrucbachbadinh
4A-4B Bảo Khánh, Phường Hàng Trống, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 4a-4bbaokhanhhangtronghoankiem
11 Hàng Rươi, Hàng Mã, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 11hangruoihangmahoankiem
67 Hàng Thiếc, Hàng Gai, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 67hangthiechanggaihoankiem
22 Phố Bảo Khánh, Phường Hàng Trống, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 100000 ==> 22phobaokhanhhangtronghoankiem
```

- **“longitude”** và **“latitude”**: với những tập dữ liệu hay bản ghi còn thiếu hai trường tọa độ này, nhóm xây dựng một con spider để crawl hai trường longitude và latitude dựa theo địa chỉ “address”:

 <https://www.google.com/maps/search/hồ hoàn kiếm>

Sau khi chạy sẽ thu được kết quả:

 [google.com/maps/place/Hồ+Hoàn+Kiếm/@21.0287748,105.8523647,17z/data=!3m1!4b1!4m5!3m4!1s0x3135a](https://www.google.com/maps/place/Hồ+Hoàn+Kiếm/@21.0287748,105.8523647,17z/data=!3m1!4b1!4m5!3m4!1s0x3135a)

và nhóm sẽ dựa theo url này để trích xuất ra latitude và longitude đằng sau ký tự “@” và đưa vào dataframe

nages	lat	lot
024x...	105.823386	21.067211
024x...	106.720364	10.795147

4. Đối sánh lược đồ

Trong 4 dataset nhóm sẽ lựa chọn booking làm bộ dữ liệu chuẩn, để đối sánh lược đồ do bộ dữ liệu này có độ hoàn thiện cao nhất.

4.1 Dựa trên tên trường thuộc tính

Nhóm sử dụng khoảng cách Levenshtein để tính độ tương đồng giữa tên của các trường dữ liệu.

```
In [31]: booking_list=booking.columns.values
ebooking_list=ebooking.columns.values
rs={}
for i in booking_list:
    x={}
    for j in ebooking_list:
        x[j]=levenshtein(j,i)
    x={k: v for k, v in sorted(x.items(), key=lambda item: item[1])}
    rs[i]=x
```

```
In [32]: rs
        'rating': 0,
        'n_review': 7,
        'facilities': 9},
        'address': {'address': 0,
        'url': 6,
        'name': 6,
        'price': 6,
        'star': 6,
        'nearby': 6,
        'img': 7,
        'rating': 7,
        'n_review': 7,
        'city': 7,
        'facilities': 8},
        'stars': {'star': 1,
```

Sau đó nhóm sẽ lấy ra các bản ghi ở mỗi trường để tiếp tục đối sánh lược đồ dựa trên bản ghi (dùng luật cú pháp).

4.2 Dựa trên bản ghi (luật cú pháp)

Dựa vào booking dataset thì nhóm sẽ có các tập luật để đối sánh giá trị của từng record:

- “**url**”: Phải chứa tên của domain.com VD: agoda phải chứa agoda.com... (trường image url không chứa giá trị này)

```
|: import re
count=0
number=0
for i in ebooking['url']:
    if i==i:
        number+=1
        if re.search('ebooking.com',i):
            count+=1
print(count/number*100,end=' ')
print('%')
```

100.0 %

- “images”: Phải chứa ‘.jpg’

```
: count=0
number=0
for i in booking['images']:
    if i==i:
        number+=1
        if re.search('.jpg',i):
            count+=1
print(count/number*100,end=' ')
print('%')
```

99.94500458295143 %

- “star”: giá trị kiểu float trong khoảng từ 0-5, ngoại trừ giá trị -1. Tương tự rating: giá trị kiểu float giá trị trong khoảng 0-10, ngoại trừ giá trị -1

```

count=0
number=0
for i in expedia['stars']:
    if i !=-1:
        number+=1
        if i<=5.0:
            count+=1
print(count/number*100,end=' ')
print('%')

```

100.0 %

- “**longitude**”: trong khoảng 102-110, tương tự với vĩ độ từ 9-23

```

count=0
number=0
for i in agoda['lot']:
    if i==i:
        number+=1
        if i>102 and i<110:
            count+=1
print(count/number*100,end=' ')
print('%')

```

99.95426829268293 %

- “**city**”: chỉ gồm một trong các giá trị ‘Hà Nội’, ‘Hồ Chí Minh’, ‘Đà Nẵng’, ‘Đà Lạt’, ‘Quảng Ninh’, ‘Vũng Tàu’, ‘Nha Trang’, ‘Phú Quốc’

```

: count=0
number=0
for i in expedia['city']:
    if i==i:
        number+=1

        if re.search('^(Hà Nội|Hồ Chí Minh|Đà Lạt|Đà Nẵng|Nha Trang|Quảng Ninh|Phú Quốc|Vũng Tàu)$',i):
            count+=1
print(count/number*100,end=' ')
print('%')

```

100.0 %

- “**address**”: sau khi bắt được city thì address là trường có tỉ lệ chứa 1 trong các string ‘Hà Nội’, ‘Hồ Chí Minh’, ‘Đà Nẵng’, ‘Đà Lạt’, ‘Quảng Ninh’, ‘Vũng Tàu’, ‘Nha Trang’, ‘Phú Quốc’ lớn nhất.

```

count=0
number=0
for i in agoda['address']:
    if i==i:
        number+=1

        if re.search('(Hà Nội|Hồ Chí Minh|Đà Lạt|Đà Nẵng|Nha Trang|Quảng Ninh|Phú Quốc|Vũng Tàu)',i):
            count+=1
print(count/number*100,end=' ')
print('%')

```

89.86146095717883 %

- “**hotel name**”: riêng đối với trường trường này không có quy luật chung, hơn nữa nhóm đã bắt được url ở phần trên do vậy chúng đưa về chữ thường không dấu và dùng luật: tỉ lệ chứa từ ‘hotel’, ‘khách sạn’, ‘chung cư’, ‘căn hộ’ cao nhất

```

count=0
number=0
for i in ebooking['name']:
    if i==i:
        number+=1
        i=unicode.decode(str.lower(i))
        if re.search('hotel|khách sạn|chung cư|căn hộ',i):
            count+=1
print(count/number*100,end=' ')
print('%')

```

53.402607986960064 %

```

count=0
number=0
for i in ebooking['address']:
    if i==i:
        number+=1
        i=unicode.decode(str.lower(i))
        if re.search('hotel|khách sạn|chung cư|căn hộ',i):
            count+=1
print(count/number*100,end=' ')
print('%')

```

0.14262428687856563 %

5. Đối sánh dữ liệu

5.1. Xử lý trùng lặp

Xử lý trùng lặp ở mỗi domain là một điều tối quan trọng trước khi matching, bước này nhóm sẽ sử dụng các thuật toán để loại bỏ các dữ liệu bị trùng lặp.

Ở đây nhóm sử dụng thư viện recordlinkage để xử lý trùng lặp: Các tham số của recordlinkage sau khi tối ưu, ở đây có một hàm mới là Geographic, thì hàm này chính là so sánh dựa trên tọa độ, tham số có nghĩa là trong bán kính 4km thì sẽ được điểm tối đa, ra ngoài 4km cứ mỗi 2km điểm sẽ giảm theo hàm Gauss

```
from recordlinkage.compare import Geographic

compare = recordlinkage.Compare()
compare.add(Geographic('lot', 'lat', 'lot', 'lat', offset=4, scale=2, method='gauss', label='distance'))

compare.string('new_add',
              'new_add',
              method='jarowinkler',
              threshold=0.75,

              label='add')

compare.string('new_name',
              'new_name',
              method='levenshtein',
              threshold=0.8,

              label='name')
```

Do đã thống nhất cách đặt tên city nên để giảm bớt lượng dữ liệu candidate nhóm block city lại. Như hình bên dưới đây là cùng một hotel và đăng 2 phòng khác nhau, nhóm không muốn điều này xảy ra và loại bỏ phòng có giá cao hơn.

```
|: ebooking.loc[929]

|: url          https://hotels.ebooking.com/ho1457207552/?pa=9...
   img          https://exp.cdn-hotels.com/hotels/46000000/455...
   name         Luxury Apartment 3BR Vinhomes Metropolis
   address      29 Liễu Giai, Trúc Bạch, Ba Đình, Hà Nội, 1000...
   rating       8
   facilities    Hồ bơiCó bãi đậu xeBồn tắmBếp nhỏMáy điều hòa ...
   price        3.53249e+06
   star         3
   n_review     1
   nearby       Tại khu vực Quận Ba Đình,Cách Lăng Bác 33 phút...
   lat          21.0318
   lot          105.814
   city         Hà Nội
   new_add      29lieugiaitrucbachbadinh
   new_name     luxuryapartment3brvinhomesmetropolis
   Name: 929, dtype: object
```

```
|: ebooking.loc[296]

|: url          https://hotels.ebooking.com/ho1078334688/?pa=2...
   img          https://exp.cdn-hotels.com/hotels/34000000/336...
   name         Luxury Apartment in Vinhomes Metropolis
   address      29 Liễu Giai, Ba Đình, Hà Nội, 100000, Việt Nam
   rating       10
   facilities    Có bãi đậu xeBồn tắmBếpMáy điều hòa nhiệt độ
   price        1.20446e+06
   star         3.5
   n_review     10
   nearby       Tại khu vực Quận Ba Đình,Cách Lăng Bác 33 phút...
   lat          21.0316
   lot          105.814
   city         Hà Nội
   new_add      29lieugiaibadinh
   new_name     luxuryapartmentinvinhomesmetropolis
```

Tại bước này mỗi dataset sẽ giảm thêm khoảng vài chục đến hơn 100 record nữa.

Riêng đối với agoda thì ngoại lệ, agoda có rất nhiều dạng chung cư, căn hộ cho thuê, dữ liệu này là cá biệt chỉ domain này có như vậy record dạng này sẽ không cần thiết trong Project, do vậy nhóm sẽ không xử lý chúng, do đó làm các record này sẽ vượt qua được threshold về name 1 cách dễ dàng từ đó sẽ bị trùng lặp và bị loại bỏ.

```
: city                    Nha Trang
hotel name               Chung cư 52 m2 2 phòng ngủ, 2 phòng tắm riêng ...
url                     https://www.agoda.com/vi-vn/hoty-s-house/hotel...
address                 Vĩnh Phước, Nha Trang, Việt Nam
stars                   5
price                   388636
rating                  9.2
n_reviews               7
image                   https://pix5.agoda.net/hotelImages/8101685/0/5...
reviews                 []
facilities               ['Thang máy', 'Điều hòa', 'Đồ vải lanh']
nearby places            ['Viện Hải dương học - 7,44 km', 'Đại học Nha ...
lat                     12.2733
lot                     109.202
new_add                 vinhphuoc
new_name                 chungcu52m22phongngu,2phongtamriengovinhphuoc
Name: 167, dtype: object
```

```
: agoda.loc[71]
```

```
: city                    Nha Trang
hotel name               Chung cư 50 m2 2 phòng ngủ, 1 phòng tắm riêng ...
url                     https://www.agoda.com/vi-vn/livin-homestay-nha...
address                 Vĩnh Phước, Nha Trang, Việt Nam
stars                   5
price                   950000
rating                  9.7
n_reviews               15
image                   https://pix6.agoda.net/hotelImages/agoda-homes
```

Sau bước lọc thứ 2 này bộ dữ liệu còn khoảng 18k bản ghi, (agoda chiếm khoảng 90% lượng bản ghi bị loại bỏ)

5.2 Tổng hợp dữ liệu

Để không phải đối sánh nhiều lần thì nhóm gộp các dữ liệu lại thành 1 tập khi đó thì việc đối sánh dữ liệu chính là việc lọc trùng tập hợp này.

Việc này sinh ra khó khăn là sau này lấy dữ liệu ra chính xác kiểu gì thì khi gộp, nhóm sẽ tạo ra một tập data tổng hợp mà chỉ có các trường dùng để đối sánh kèm id của từng domain ở các tập con (id1, id2, id3, id4).

	address	city	hotel name	id1	id2	id3	id4	lat	lot	new_add	new_name
0	6, Dinh Thon, My Dinh, Tu Lien, Hà Nội, Việ...	Hà Nội	RedDoorz near My Dinh Bus Station	0.0	NaN	NaN	NaN	21.052125	105.836715	6dinhthonmydinhhtuliem	reddoorznearmydinhbusstation
1	29 Liễu Giai, Quận Ba Đình, Hà Nội, Việt Nam	Hà Nội	22housing Luxury Apartment Vinhomes Metropolis	1.0	NaN	NaN	NaN	21.036235	105.847726	29lieugiaibadinh	22housingluxuryapartmentvinhomesmetropolis
2	112 Cau Go Street, Hoan Kiem, Hà Nội, Quận Hoàn...	Hà Nội	SkyLake Homestay	2.0	NaN	NaN	NaN	21.018257	105.776398	112caugohoankiemhoankiem	skylakehomestay

5.3. Đối sánh dữ liệu lần 1

Tiếp đến là sau khi matching thì nhóm cần lấy ra một bộ dataset nếu match ở 2 domain thì giữ lại record chung đó và xóa 2 record đơn kia đi. Và dưới đây là một hàm để giải quyết vấn đề này sử dụng set của python.

```
match=eb_matches['level_0'].unique()
num=len(match)
tick=[1]*max(eb_matches['level_0'].max(),eb_matches['level_1'].max()+[1]
ls=[]
for i in match:
    #if tick[i]:

        a=set()

        b=set()
        b.add(i)
        while b:
            val=b.pop()
            if tick[val]:
                a.add(val)
                tick[val]=0
                x=eb_matches.loc[eb_matches['level_0']==val,'level_1'].to_list()
                x=x+eb_matches.loc[eb_matches['level_1']==val,'level_0'].to_list()
                for j in x:
                    if j:
                        b.add(j)
                        a.add(j)

        if a:
            ls.append(a)
```

Nhóm vẫn sử dụng bộ Compare như bước trên và giảm bớt threshold để nới lỏng điều kiện, mục đích trong lần chạy này là có một dataset chuẩn dùng để tối ưu tham số.

Ví dụ kết quả sau khi chạy lần 1:

```

{1114, 147}
{493, 1159}
{640, 3973, 3017, 3511, 1209}
{350, 1358}
{1392, 3107, 125, 3439}
{84, 1407}
{1409, 3289, 502, 3676}
{1546, 594}
{3560, 987, 2974, 1591}

```

Mỗi set trong tập hợp là index của record trong tập tổng hợp trên, và các phần tử trong set là các record trùng ở các domain (hoặc chỉ là 1 domain do đã nói lỏng threshold)

Từ đây nhóm lấy ra khoảng hơn 400 record thủ công tìm kiếm, xác minh để đảm bảo **trùng** đúng không thừa không thiếu.

```

for i in range(a,b):
    c=ls[i]
    checked_ls.append(c)

```

```
checked_ls.append(ls[2040:2033])
```

```

for i in range(1000,2500):
    print(i)
    print(new_data.loc[list(ls[i]),['id1','id2','id3','id4','hotel name']])
    print('\n')

```

```

2410
      id1 id2      id3      id4      hotel name
13482 NaN NaN  4222.0      NaN Bondi Backpackers Nha Trang Hostel
16076 NaN NaN      NaN  1945.0 Bondi Backpackers Nha Trang Hostel

```

```

2411
      id1 id2      id3      id4      hotel name
16078 NaN NaN      NaN  1947.0 Vinpearl Luxury Nha Trang
13311 NaN NaN  4050.0      NaN Vinpearl Luxury Nha Trang

```

```

2412
      id1 id2      id3      id4      hotel name
13485 NaN NaN  4225.0      NaN Khách sạn 4H
16079 NaN NaN      NaN  1948.0 Khách sạn 4H

```

```
len(checked_ls)
```

```
402
```

5.4. Tối ưu trọng số

Như đã giới thiệu bên trên nhóm sẽ kiểm tra thủ công lấy ra hơn 400 record “chuẩn xác” từ tập match lần 1 gọi là checked_ls:

```
checked_ls
{3687, 5384, 13290, 16229},
{5301, 5386},
{5295, 5392},
{3745, 5410, 13372, 15142},
{5195, 5411, 13233, 16172},
{5163, 5429},
{3696, 5430, 13146, 15055},
{3679, 5433, 13309, 15052},
{3678, 5453, 13399, 15093},
{5101, 5454, 13164, 16240},
{5138, 5456},
{5091, 5457, 13241, 16192},
{3672, 5458, 13103, 15058},
{3787, 5463},
{5207, 5477},
{5119, 5480},
{3681, 5481, 13153, 16243}
```

“Chuẩn xác” ở đây có nghĩa là với set 2 phần tử thì khách sạn với id tương ứng đó có ở đúng 2 domain,... Từ đó nhóm xây dựng hàm đánh giá các bộ tham số, cứ chứa 1 set 2 phần tử sẽ được 1 điểm, 1 set 3 phần tử được 2 điểm, 1 set 4 phần tử được 3 điểm.

```
def eva(ls,checked_ls):
    score=0
    for i in ls:
        if i in checked_ls:
            l= len(i)
            if l==2:
                score+=1
            elif l==3:
                score+=2
            else:
                score+=4
    return score
```

```
xlist=[6,7,8,9,10,11]
ylist=[4,5,6,7,8,9]
zlist=[1,2,3,4,5,6]
```

```
score=0
fx=0
fy=0
fz=0
fls=[]
for x in xlist:
    for y in ylist:
        for z in zlist:
            t=(x+y+z)*0.93
            eb_matches = eb_ft0[eb_ft0['name'].values*x+ eb_ft0['add'].values*y+eb_ft0['distance'].values*z >=t ].reset_index()
            ls=find(eb_matches)
            ns=eva(ls,checked_ls)

            if ns>score:
                score =ns
                fx=x
                fy=y
                fz=z
                fls=ls
```

Kết quả tối ưu:

- Điểm số cao nhất: 702
- Bộ trọng số: 11 với name, 5 với address và 1 với distance
- Bộ tham số tối ưu:

```
[eb_ft0['name'].values*11+ eb_ft0['add'].values*5+eb_ft0['distance'].values*1 >=15.8
```

Kết quả cuối cùng lọc ra 3547 khách sạn có ở 2 domain trở lên để thực hiện các bước tiếp theo.

Trong đó:

- Booking : 1079
- Agoda: 1153
- Ebooking: 3250
- Expedia: 2925

6. Phân tích dữ liệu

Để hiểu hơn về dữ liệu cũng như đưa ra những phân tích đánh giá giúp hỗ trợ việc đưa ra quyết định, nhóm thực hiện một vài phân tích chính.

6.1. Tổng quan về dữ liệu

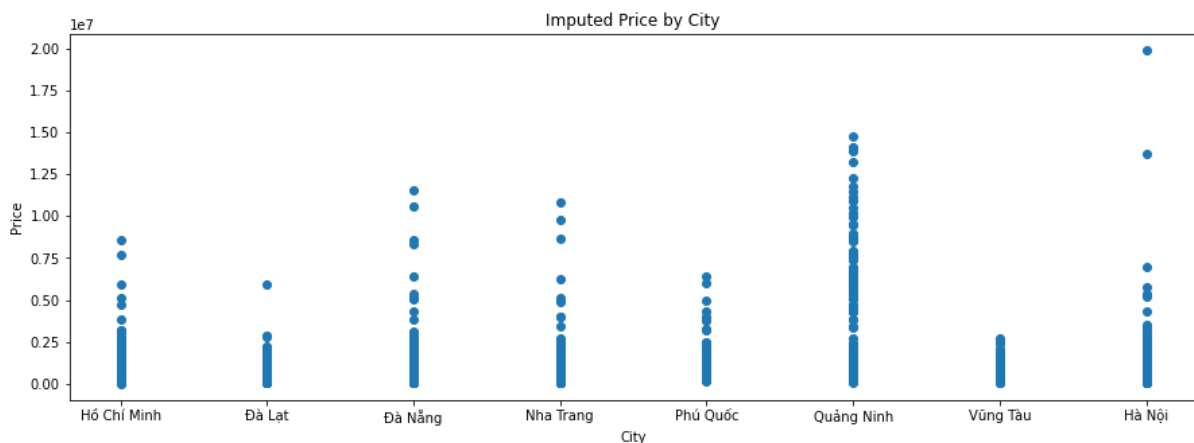
	min_price	mean_stars	mean_rating	number_review
mean	8.002679e+05	2.890696	6.734249	216.620806
std	1.310980e+06	0.854393	1.385615	603.545596
min	2.080400e+04	0.000000	2.000000	0.000000
25%	3.063025e+05	2.500000	6.000000	1.000000
50%	4.719210e+05	3.000000	6.800000	15.000000
75%	7.721710e+05	3.500000	7.500000	133.000000
max	1.987850e+07	5.000000	10.000000	8633.000000

6.2. Thống kê dữ liệu thiếu

```
name          0.000000
city          0.000000
min_price     0.000000
mean_stars    0.311086
mean_rating   21.848162
number_review 0.000000
facilities    0.000000
```

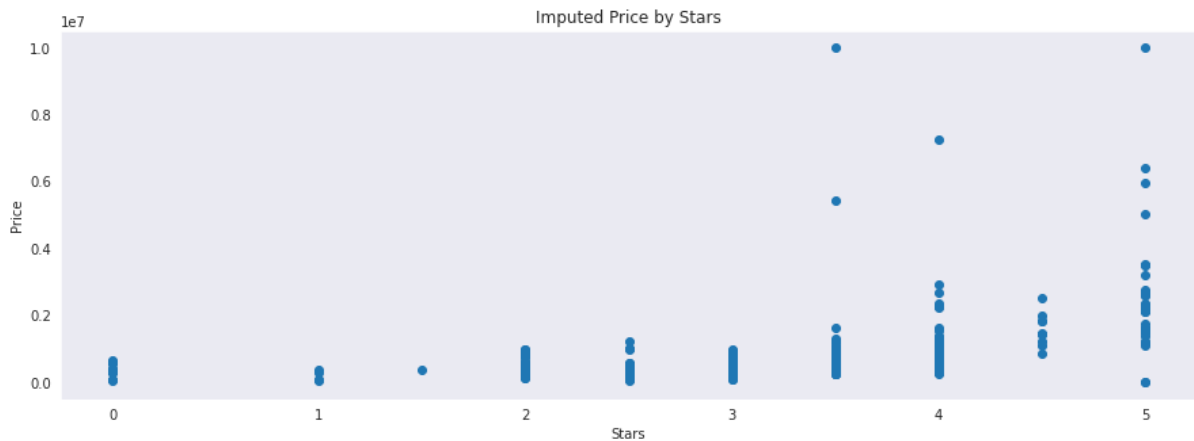
Giá trị được thể hiện theo % thiếu của dữ liệu. Ví dụ, ‘mean_stars’ thiếu 0.31%.

6.3 Mối quan hệ giữa giá và khu vực



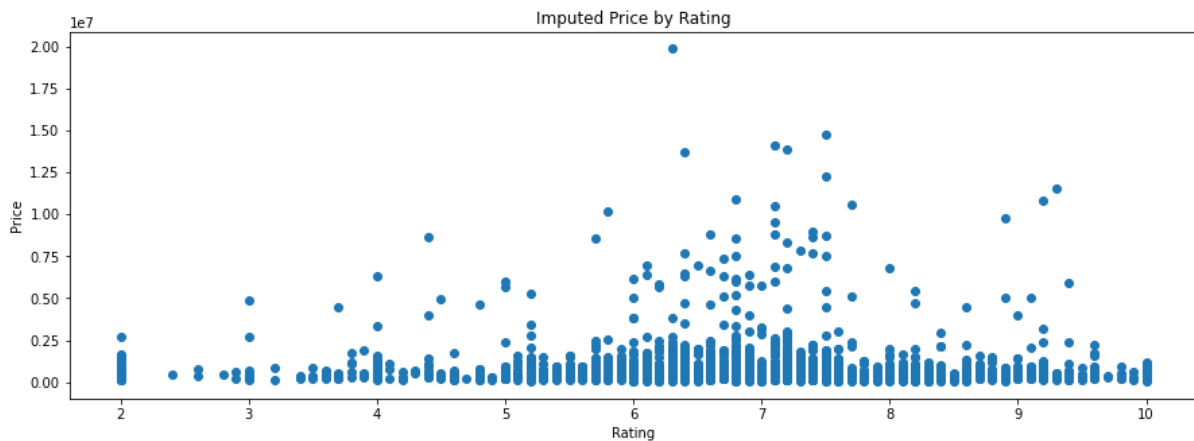
Khách sạn ở các thành phố có mức giá trung bình dưới 3.000.000 đồng.

6.4. Mối quan hệ giữa giá và số sao



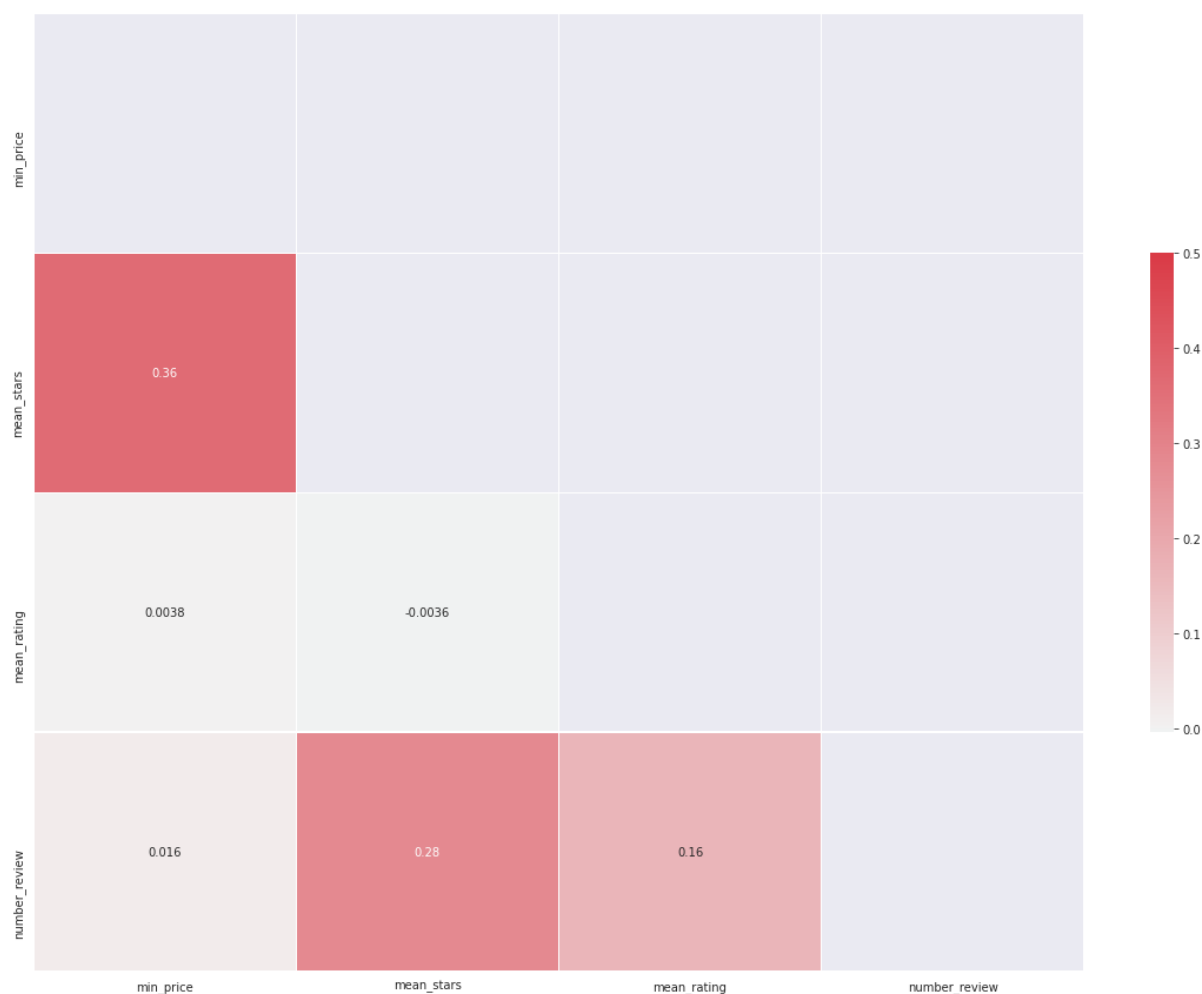
Giá được tăng dần theo số sao, số sao càng cao, khách sạn càng có giá đắt.

6.5. Mối quan hệ giữa giá và điểm đánh giá



Giá khách sạn cao thường có điểm đánh giá 7, tuy nhiên cũng có một số điểm đánh giá 9. Nhìn chung, điểm đánh giá không phụ thuộc nhiều vào giá.

6.6. Độ tương quan giữa các thuộc tính



Quan sát biểu đồ tương quan, ta có thể thấy được mối tương quan đặc biệt giữa giá và số sao, khi có điểm tương quan là 0.36, không quá cao, nhưng so với các thuộc tính khác thì rõ ràng, nó thể hiện một phần được việc, khi số sao càng cao, thì sẽ đi kèm với giá thành càng cao, điều này hoàn toàn có thể hiểu được trong thực tế.

7. Xếp hạng khách sạn

Với lượng dữ liệu lớn, hàng nghìn khách sạn mỗi thành phố, người dùng không thể xem hết lần lượt các khách sạn để tìm ra khách sạn tốt nhất mà giá phù hợp nhất. Chính vì vậy, nhiệm vụ đề ra là phải xếp hạng các khách sạn, sao cho những khách sạn tốt nhất, được đánh giá cao mà giá cả rẻ nhất sẽ được hiển thị sớm với người dùng, trên cùng của mỗi danh sách.

Để thực hiện được việc này, nhóm đã dựa vào những điểm đánh giá của khách sạn và dữ liệu giá của khách sạn đó để tính ra được điểm đánh giá tổng quan (overall_score) cho từng khách sạn. Công thức được thiết kế như sau:

$$\text{overall_score} = a * \text{rating} + b * \text{price} + c * \text{number_review}$$

Trong đó:

“**rating**” là điểm đánh giá trung bình từ các OTAs

“**price**” là giá thấp nhất của khách sạn

“**number_review**” là số lượng phản hồi từ phía người dùng

	price	rating	number_review
count	4.700000e+02	467.000000	470.000000
mean	8.458169e+05	8.248608	215.912766
std	1.296243e+06	0.785308	290.799858
min	4.553700e+04	5.000000	0.000000
25%	3.501360e+05	7.800000	19.000000
50%	5.244210e+05	8.500000	89.000000
75%	8.098810e+05	8.800000	295.750000
max	1.000000e+07	9.900000	1836.000000

Dựa vào mô tả của các giá trị, nhóm lựa chọn và căn chỉnh tham số cho phù hợp. Bộ tham số được lựa chọn cuối cùng là $\{a, b, c\} = \{50000, -1, 2500\}$

Sau khi xếp hạng, ta được một vài kết quả như sau:

rank	hotel_id	name	star	price	rating	number_review
1	109	Rosa Hotel & Spa	3.0	268398.0	7.9	1836.0
2	31	Chez Mimosa Boutique Corner	4.0	314701.0	9.1	1516.0
3	185	Richico Apartment and Hotel	3.0	496970.0	8.1	1482.0
4	30	Khách sạn Dyn Opera	3.5	351450.0	8.7	1274.0
5	34	RedDoorz tại phố đi bộ Bùi Viện (RedDoorz @ Bu...	2.0	261818.0	7.2	1206.0
...
466	284	InterContinental Đà Nẵng Sun Peninsula Resort	5.0	10000000.0	8.8	15.0
467	341	Adela Boutique Hotel	3.5	10000000.0	7.0	0.0

8. Đề xuất khách sạn

Để tăng sự tiện lợi, tiết kiệm thời gian tìm kiếm và đưa ra cho người dùng thêm một số lựa chọn, nhóm đã thực hiện việc đề xuất ra những khách sạn tương đồng với khách sạn hiện tại người dùng đang xem.

Do không có dữ liệu người dùng, chỉ có các thông tin sẵn có của các khách sạn, nên việc áp dụng những hệ gợi ý thông thường như lọc cộng tác, gợi ý dựa theo phiên, ... là chưa khả thi. Từ đó, ý tưởng của nhóm là tìm ra những khách sạn tương đồng dựa vào các thông tin sẵn có của nó.

Bằng cách sử dụng hàm NearestNeighbors của thư viện sklearn, áp dụng vào bộ dữ liệu của mình, có thể tìm ra nhanh được những bản ghi hay khách sạn có khoảng cách gần hay được coi như tương đồng với khách sạn hiện tại.

Khách sạn đang xem

hotel_id		name
106	10	Khách sạn Hà My (Ha My Hotel)

Chẳng hạn như khi người dùng đang xem về “Khách sạn Hà My (Ha My Hotel)”, người dùng sẽ được đề xuất ra một số khách sạn tương đồng như “Khách sạn Oscar Sài Gòn”, “Golden Central Hotel Sài Gòn” , ...

Các khách sạn tương đồng

hotel_id		name
97	87	Khách sạn Oscar Sài Gòn
131	172	Cozrum Homes Rivera Corner
201	106	RedDoorz near Saigon Train Station 2
232	5	Khách sạn Winston Riverside (Hotel Winston Riv...
278	77	Golden Central Hotel Sài Gòn

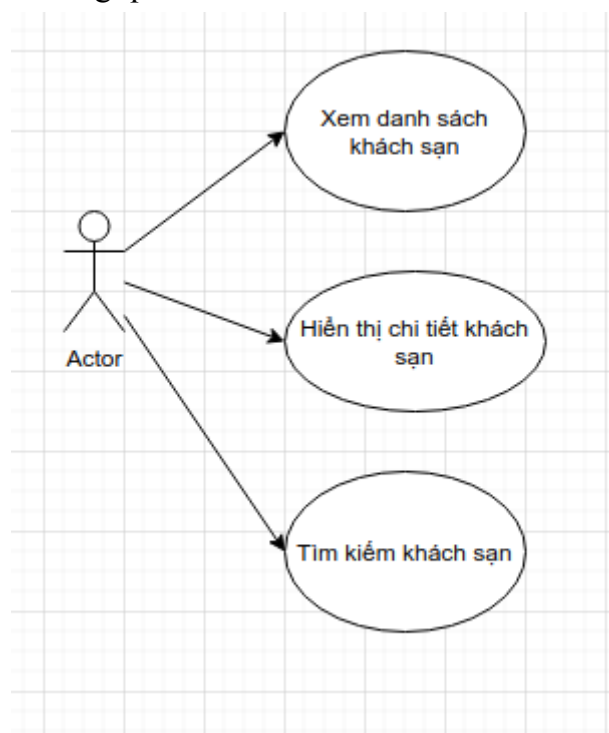
Cuối cùng, để dễ cho việc truy vấn, nhóm thực hiện việc chuyển đổi và đưa ra danh sách các id của khách sạn tương đồng với id của khách sạn hiện tại.

	hotel_id	suggest1	suggest2	suggest3	suggest4	suggest5
179	1	179	95	75	230	246
143	2	143	37	81	1	272
121	3	121	180	159	189	24
70	4	70	65	78	128	258
232	5	232	190	290	221	261
...
402	458	402	86	385	360	408
354	459	354	274	226	356	239

9. Xây dựng website khai thác dữ liệu

Để có thể khai thác các khách sạn thu thập được thì nhóm thực hiện xây dựng hiển thị thông tin khách sạn sử dụng Framework Laravel và thực hiện lưu trữ dữ liệu đã được xử lý vào MySQL.

Biểu đồ use case tổng quát:



- Chức năng xem danh sách khách sạn: Cho phép người dùng có thể xem được nhiều khách sạn, có thể lọc kết quả dựa theo thành phố, giá khách sạn, số sao của khách sạn.
- Chức năng hiển thị chi tiết khách sạn: Cho phép người dùng xem các trường thông tin của khách sạn và kết nối đến trang thông tin gốc của khách sạn với các miền khác nhau.

- Chức năng tìm kiếm khách sạn: Cho phép người dùng tìm kiếm khách sạn theo tên, thông tin mô tả khách sạn, theo thành phố.

Dưới đây là một số các giao diện của website:

Travel Hotels

Trang chủ

Về chúng tôi


Khách sạn

Liên lạc

Luôn luôn lắng nghe

Special Offers


Top Khách sạn dành cho bạn



Chez Mimosa Boutique Corner

314,701 VND

4 Stars 9 Rating




Richico Apartment and Hotel

496,970 VND

3 Stars 8 Rating

176-178 Nguyễn Văn




Khách sạn Dyn Opera

351,450 VND


3 Stars 8 Rating

18 Cao Ba Quát, Ben Nghe ward, District 1, Ho



RedDoorz tại phố đi bộ Bùi Viện (RedDoorz @ Bui Vien Walking

261,818 VND



Khách sạn Lê Hoàng Beach (Le Hoang Beach Hotel

875,555 VND

Travel Hotels

Trang chủ

Về chúng tôi

Khách sạn





Liên lạc

Luôn luôn lắng nghe


Lợi ích khách sạn

Đưa đón sân bay, Dọn phòng hằng ngày, Wi-Fi miễn phí trong tất cả các phòng, Đội ngoại tệ, Điều hòa

Liên kết

 Booking.com - 1,792,155 VND	 Agoda.com - 1,369,990 VND	 Ebooking.com - 1,387,189 VND	 Expedia.com.vn - 1,386,707 VND
---	---	--	--

Related Hotels



TÌM KIẾM KHÁCH SẠN

Từ khóa

Nơi đến

Tìm kiếm

SẮP XẾP KHÁCH SẠN

Sắp xếp

Khoảng giá

Giá min

Giá max

Lọc

Khách sạn Little Diamond

179,000 VND

3 Stars 8 Rating

11 Bát Đàn, Hàng Bồ, Quận Hoàn Kiếm, Hà Nội, Việt Nam, 106322

27 reviews

Hà Nội

Chi tiết

Night Hotel

555,288 VND

3 Stars 8 Rating

89 Búi Thị Xuân, Búi Thị Xuân, Quận Hai Bà Trưng, Hà Nội, Việt Nam, 100000

181 reviews

Hà Nội

Chi tiết

My Hotel 23

563,479 VND

3 Stars 8 Rating

23 Búi Thị Xuân, Búi Thị Xuân, Quận Hai Bà Trưng, Hà Nội, Việt Nam, 100000

146 reviews

Hà Nội

Chi tiết

Classy Holiday Hotel & Spa

391,250 VND

Khách sạn Amorita Boutique

368,000 VND

Khách sạn Trang Premium

407,951 VND

10. Đánh giá và kết luận

Trải qua quá trình thảo luận, đề xuất phương án và cùng nhau thực hiện ứng dụng các kiến thức của môn học Tích hợp dữ liệu về cơ bản chúng em cũng đã tạo ra được website như dự tính ban đầu.

Hướng phát triển tương lai:

- Xây dựng phần mềm khép kín bắt đầu từ bước thu thập dữ liệu
- Streaming hóa để bắt kịp các luồng dữ liệu mới nhất

11. Tài liệu tham khảo

1. <https://docs.scrapy.org/en/latest/topics/spiders.html>
2. <https://recordlinkage.readthedocs.io/en/latest/ref-compare.html>