TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

TÊN ĐỀ TÀI: GỢI Ý ĐẶT PHÒNG KHÁCH SẠN

HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN	ĐIỂM BẢO VỆ
Nguyễn Văn Quốc Hùng	19N13	
Tôn Nữ Hoàng Giang	19N13	
Văn Trung Hiếu	19N13	

ĐÀ NẴNG, 06/2022

TÓM TẮT

- Đặt vấn đề: với sự phát triển của các website thương mại điện tử, việc đưa người dùng tiếp cận những sản phẩm phù hợp là rất cần thiết. Và gợi ý đặt phòng khách sạn ra đời để đáp ứng nhu cầu đó. Nó giúp gợi ý đặt phòng khách sạn tốt nhất cho người dùng với hai tiêu chí chính đó là giá cả thấp và chất lượng tốt nhất.

- Phương pháp giải quyết: đầu tiên, cần thu thập được dữ liệu từ các khách sạn. Từ dữ liệu đó, nhóm em sẽ trích xuất đặc trưng cần thiết và đề xuất ra hai giải pháp đó là thuật toán K-means và thuật toán Hierarchical để phân cụm dữ liệu dựa theo các đặc trưng đã chọn. Sau đó, so sánh, đánh giá bằng hai thuật toán đó là DB Index và Silhouette Index.

- Kết quả đạt được: chọn được vùng dữ liệu chứa các khách sạn có chất lượng tốt và giá cả lại thấp để đưa ra được các gợi ý cho người dùng.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá theo 3 mức (Đã hoàn thành/Chưa hoàn thành/Không triển khai)
Nguyễn Văn Quốc Hùng	Mô hình hóa dữ liệu Đánh giá mô hình Gợi ý	Đã hoàn thành Đã hoàn thành Đã hoàn thành
Tôn Nữ Hoàng Giang	Lựa chọn đặc trưng Làm sạch/ Chuẩn hóa/ Giảm chiều dữ liệu	Đã hoàn thành Đã hoàn thành
Văn Trung Hiếu	Thu thập dữ liệu Xử lý dữ liệu Mô tả dữ liệu	Đã hoàn thành Đã hoàn thành Đã hoàn thành

MỤC LỤC

1. Giới thiệu

Với vấn đề được đặt ra là gợi ý đặt phòng khách sạn tốt nhất cho người dùng với hai tiêu chí chính đó là giá cả thấp và chất lượng tốt thì những vấn đề cần giải quyết ở bài toán này đó là thu thập được dữ liệu từ các khách sạn có chứa dữ liệu cần thiết đó là: Giá phòng, Lượt đánh giá, Điểm đánh giá. Từ dữ liệu đó nhóm em đề xuất ra hai giải pháp đó là dùng thuật toán K means và thuật toán Hierarchical để phân cụm dữ liệu thu được dựa theo ba tiêu chí đó. Tiếp sau đó so sánh, đánh giá hai thuật toán đó bằng DB Index và Silhouette Index để đưa ra được kết quả tốt nhất. Cuối cùng là chọn được vùng dữ liệu chứa các khách sạn có chất lượng tốt (thể hiện ở Lượt đánh giá và Số điểm đánh giá) và giá cả lại thấp (thể hiện ở Giá phòng) để đưa ra được các gợi ý cho người dùng từ cụm dữ liệu đó.

2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

- Nguồn thu thập dữ liệu: https://www.booking.com/.

- Công cụ thu thập (thư viện của python): selenium, bs4 (BeautifulSoup), pandas.

    + Selenium: import webdriver để điều hướng trang web.

    + bs4: import BeutifulSoup để thu thập dữ liệu từ source của trang web.

- Cách sử dụng thư viện Selenium:

    + Tải thư viện: *pip install selenium*

    + Import thư viện: *from selenium import webdriver*

    + Mở trình duyệt Chrome (Web browser chúng ta sẽ dử dụng):

        driver = webdriver.Chrome()

        output: Trình duyệt Chrome được khởi chạy.

    + Điều hướng Chrome tới một đường link cụ thể (URL cụ thể):

        driver.get(URL)

        output: Trình duyệt nhảy tới URL chỉ định.

    + Lấy hết mã nguồn (page source) của trang web hiện tại:

        driver.page_source

        output: trả về page source của trang web hiện tại.

- Cách sử dụng thư viện bs4:

    + Tải thư viện: *pip install bs4*

    + Import thư viện: *from bs4 import BeutifulSoup*

+ Lấy hết mã nguồn (source) của một trang web:

page_source = BeutifulSoup(driver.page_source)

output: mã nguồn (như lúc ta bấm ctrl + u tại một trang web bất kỳ)của trang web chỉ định ở driver sẽ được lưu vào biến.

+ Tìm một (find) hoặc nhiều (find_all) element (VD: div, span, p, ...) cụ thể (thường đi kèm với class):

hotels = page_source.find_all('div', class='a826ba81c4')

output: trả về một mảng gồm tất cả các div có class là a826ba81c4.

- Cách sử dụng thư viện pandas:

+ Tải thư viện: *pip install pandas*

+ Import thư viện: *import pandas as pd*

+ Xuất dữ liệu thu được (data) ra file csv (tên rawdata.csv):

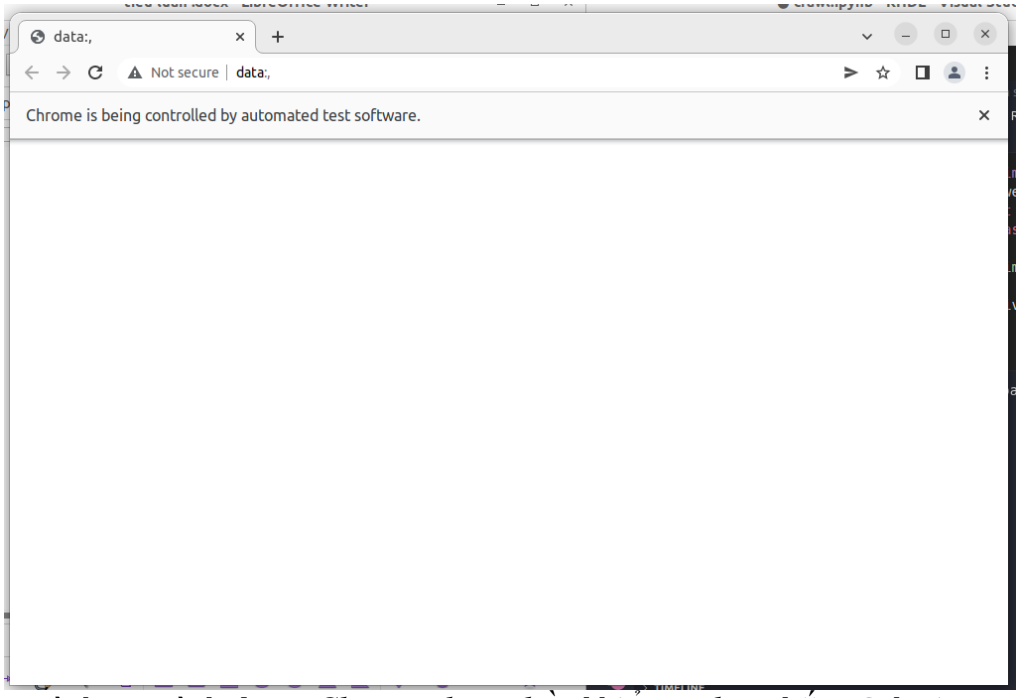
pd.DataFrame(data).to_csv('rawdata.csv', header = ['header1', 'header2', ...])

- Quá trình thu thập:

a) Import các thư viện cần thiết sau đó mở trình duyệt Chrome:

Chạy dòng lệnh: driver = webdriver.Chrome()

Trình duyệt Chrome được mở lên (Hình 1).



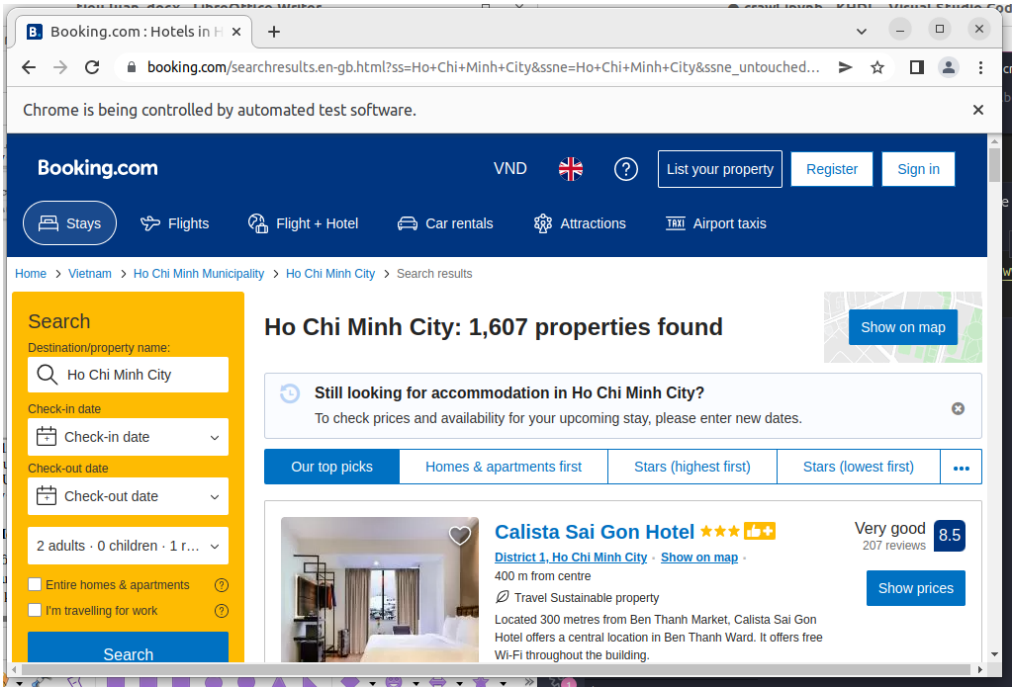
Hình 1: Trình duyệt Chrome được điều khiển tự động bằng Selenium

b) Điều hướng Chrome đến trang web cần thu thập dữ liệu

Lấy URL của trang web có dữ liệu cần thu thập.

Chạy dòng lệnh: driver.get(url)

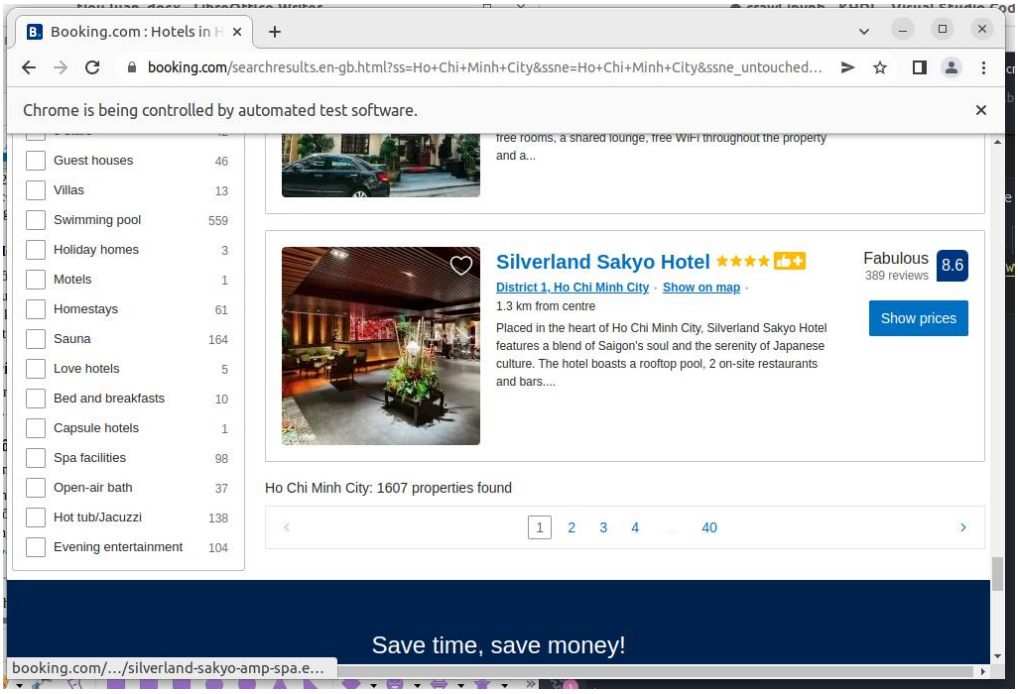
Trình duyệt sẽ mở trang web theo url nhập vào (Hình 2)



Hình 2: Trình duyệt đã được chuyển tới URL truyền vào.

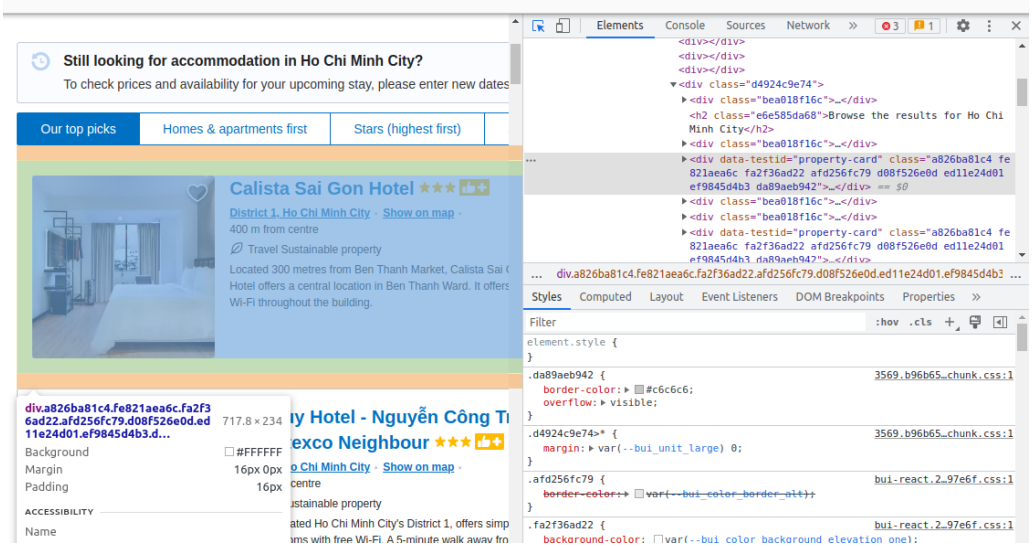
c) Xác định các thông tin cần thiết

Trang web chúng ta cần thu thập có 40 trang, 1 trang có 25 khách sạn (tổng là 1000 khách sạn) (Hình 3).



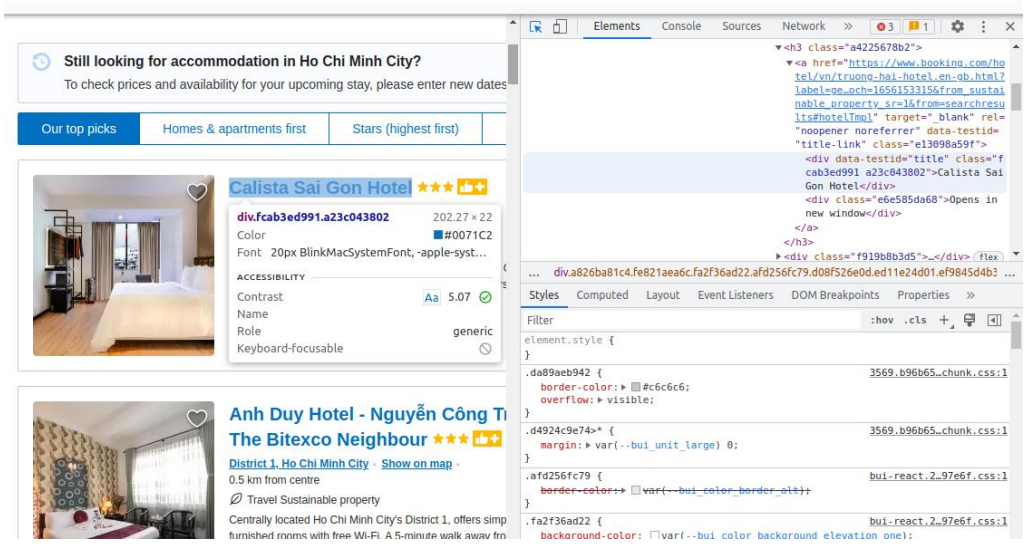
Hình 3: Số lượng trang có dữ liệu cần thu thập.

Xác định vùng có chưa các dữ liệu cần thiết cho việc thu thập (Hình 4) ở bài này vùng có chưa dữ liệu là element div có class là ‘a826ba81c4 fe821aea6c fa2f36ad22 afd256fc79 d08f526e0d ed11e24d01 ef9845d4b3 da89aeb942’.



Hình 4: Thông tin của vùng dữ liệu cần thu thập.

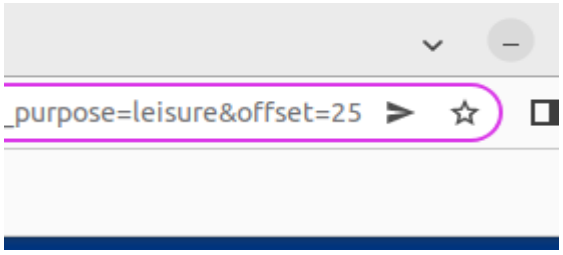
Ở trong vùng dữ liệu đó xác định các dữ liệu cần thu thập ở bài này chúng ta cần thu thập các thông tin (Tên khách sạn (Hình 5), giá phòng, số lượt đánh giá, số điểm đánh giá, địa điểm):



Hình 5: Thông tin của dữ liệu cần thu thập (VD ở đây là tên khách sạn).

d) Thực hiện thu thập dữ liệu bằng BeutifulSoup từ các thông tin từ bước c

Lặp 40 lần (tương ứng số trang web chứa dữ liệu), cuối URL của mỗi trang có một thông tin gọi là offset (Hình 6) trang thứ 1 có offset=0, trang thứ 2 có offset=25 từ đó suy ra mỗi trang sẽ có giá trị offset=số trang x 25(Hình 7).



Hình 6: Giá trị offset của trang thứ 2.

```
NUMBER_OF_PAGES = 40

def crawlData():
  res = []

  for i in range(NUMBER_OF_PAGES):
    offset = i*25
    url = URL_HCM + '&offset=' + str(offset)
    driver.get(url)
```

Hình 7: Cách lấy URL của 40 trang dữ liệu.

Ở mỗi trang dữ liệu có 25 khách sạn tương đương với 25 vùng dữ liệu (Hình 4) nên ta sẽ lấy tất cả các vùng chứa dữ liệu đó lưu vào biến hotels:

hotels = page_source.find_all('div', class_='a826ba81c4 fe821aea6c fa2f36ad22 afd256fc79 d08f526e0d ed11e24d01 ef9845d4b3 da89aeb942')

Lặp 25 lần tương ứng với 25 vùng chứa dữ liệu (Hình 8), lấy các thông tin cần thu thập của vùng đó (Hình 9).

```
    for hotel in hotels:
      hotel_name = getName(hotel)
      hotel_price = getPrice(hotel)
      hotel_rating = getRating(hotel)
      hotel_review = getReviews(hotel)
      hotel_locate = getLocation(hotel)
```

Hình 8: Lặp hết các khách sạn (25 khách sạn) và gọi các hàm để lấy thông tin.

```
def getName(hotel):
  HOTEL_NAME = hotel.find('div', class_='fcab3ed991 a23c043802')
  if(HOTEL_NAME == None):
    return '-'
  else:
    return HOTEL_NAME.text
```

Hình 9: Hàm thu thập thông tin từ bể dữ liệu (VD ở đây là lấy tên khách sạn).

e) Xuất dữ liệu thu thập được ra file csv (bằng pandas)

Xuất dữ liệu thu thập được (hotels_data) thành file csv có tên rawdata.csv:

pd.DataFrame(hotels_data).to_csv('rawdata.csv', header = ['Name', 'Price', 'Rating', 'Reviews', 'Location'])

2.2. Mô tả dữ liệu

- Tập dữ liệu sau khi thu thập được gồm có 1000 mẫu và mỗi mẫu có 5 đặc trưng đó là:

+ Name: Tên khách sạn.

+ Price: Giá phòng.

+ Rating: Điểm đánh giá.

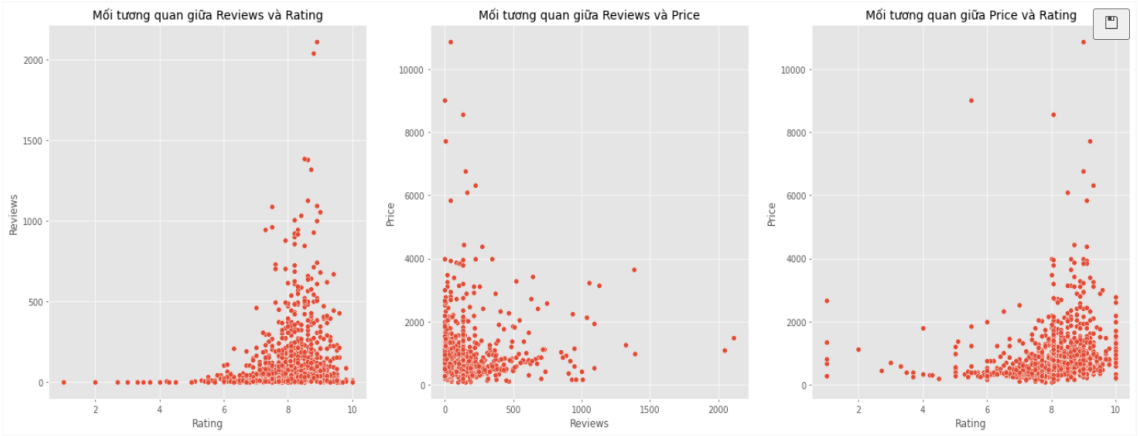
+ Reviews: Lượt đánh giá.

+ Location: Địa điểm của khách sạn.

- Tất cả đặc trưng của các mẫu thu được đều là kiểu dữ liệu *object* tuy nhiên nhóm em sẽ chuyển đổi kiểu dữ liệu của ba đặc trưng: Price, Rating, Reviews thành kiểu dữ liệu float để có thể dễ dàng xử lý dữ liệu.

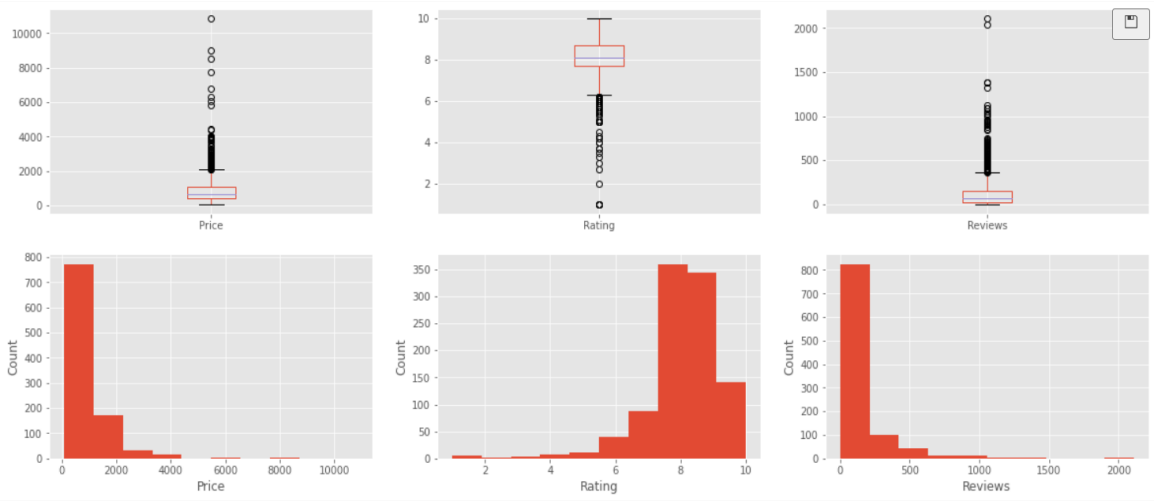
- Đặc trưng Rating có 72 dữ liệu trống (Các khách sạn chưa được đánh giá), đặc trưng ‘Reviews’ có 65 dữ liệu trống (Các khách sạn chưa được đánh giá), còn lại các đặc trưng khác đều không có giá trị trống.

- Mối tương quan giữa các đặc trưng Price, Rating, Reviews (Hình 10) cho thấy 3 đặc trưng này ít tương quan với nhau do các mẫu tập trung chủ yếu ở các góc của biểu đồ scatter:



Hình 10: Biểu đồ thể hiện mối tương quan giữa các đặc trưng Price, Rating, Reviews.

- Sự phân bố dữ liệu của các đặc trưng Price, Rating, Reviews được thể hiện ở biểu đồ Histogram và Boxplot(Hình 11) làm rõ hơn cho việc dữ liệu tập trung ở các góc của biểu đồ trên (Hình 10).

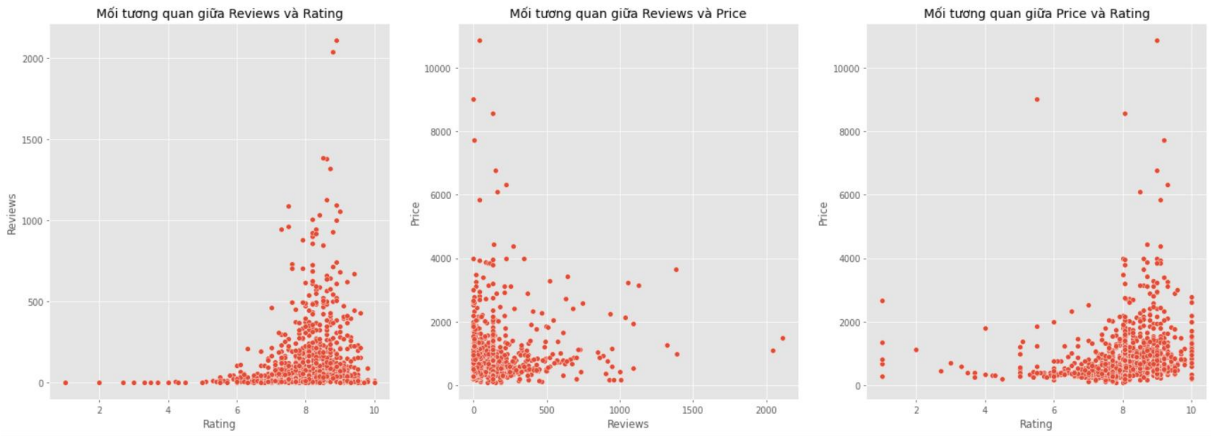


Hình 11: Biểu đồ thể hiện sự phân bố dữ liệu của các đặc trưng Price, Rating, Reviews.

3. Trích xuất đặc trưng

3.1 Lựa chọn đặc trưng

- Sự phân tán dữ liệu theo các cặp: Rating và Reviews, Rating và Price, Reviews và Price được thể hiện như sau (Hình 12).



Hình 12: Sự phân bố giữa các cặp dữ liệu.

- Trên thực tế, việc lựa chọn đặt phòng khách sạn thường dựa trên hai tiêu chí: giá phòng thấp và kèm theo đó là chất lượng tốt được thể hiện qua điểm đánh giá. Tuy nhiên, số lượng lượt đánh giá cũng ảnh hưởng không nhỏ đến chất lượng cũng như độ tin cậy của điểm đánh giá.

- Dựa vào biểu đồ trên (Hình 12), ta thấy số điểm đánh giá tập trung phần lớn trong khoảng 6 – 10. Trong khoảng đó, số lượt đánh giá lại phân bố không đồng đều. Do đó để tăng độ tin cậy cho điểm đánh giá, nhóm chúng em quyết định sẽ gộp hai cột Rating và Reviews lại thành cột Rating_Reviews.

- Vì vậy, bài toán sẽ được giả quyết dựa trên hai đặc trưng đó là Price và Rating_Reviews.

3.2 Giảm chiều dữ liệu

- Tập dữ liệu sẽ được xóa đi hai cột Rating và Reviews, mà thay vào đó sẽ them vào cột Rating_Reviews được gộp lại bởi 2 cột Rating và Reviews bằng cách: Rating_Reviews = Rating * Reviews.

- Tập dữ liệu sẽ trở thành như sau (Hình 13).

	Name	Price	Location	Rating_Reviews
0	Lucky Star Hotel 266 De Tham	165.000	District 1, Ho Chi Minh City	8232.800000
1	Eden Garden Hotel	160.000	District 1, Ho Chi Minh City	3016.000000
2	Ngan Ha Hotel	516.000	District 1, Ho Chi Minh City	915.300000
3	Anh Duy Hotel - Nguyễn Công Trứ The Bitexco Ne...	90.000	District 1, Ho Chi Minh City	1216.000000
4	Calista Sai Gon Hotel	748.000	District 1, Ho Chi Minh City	1642.600000
...	...	...	...	...
995	Angela Home - A cozy 2 bedroom apartment with ...	1600.000	District 2, Ho Chi Minh City	1092.884243
996	Hotel An Nhi	290.407	Ho Chi Minh City	10.000000
997	KIMI HOST AIRBNB	1200.000	Binh Thanh, Ho Chi Minh City	1092.884243
998	Sunrise Cityview Apt *****	1840.000	Ho Chi Minh City	11.000000
999	Căn Hộ Vinhomes Bình Thạnh View Đẹp	1120.000	Binh Thanh, Ho Chi Minh City	1092.884243

1000 rows × 4 columns

Hình 13: Bảng dữ liệu sau khi giảm chiều dữ liệu.

3.3 Làm sạch dữ liệu

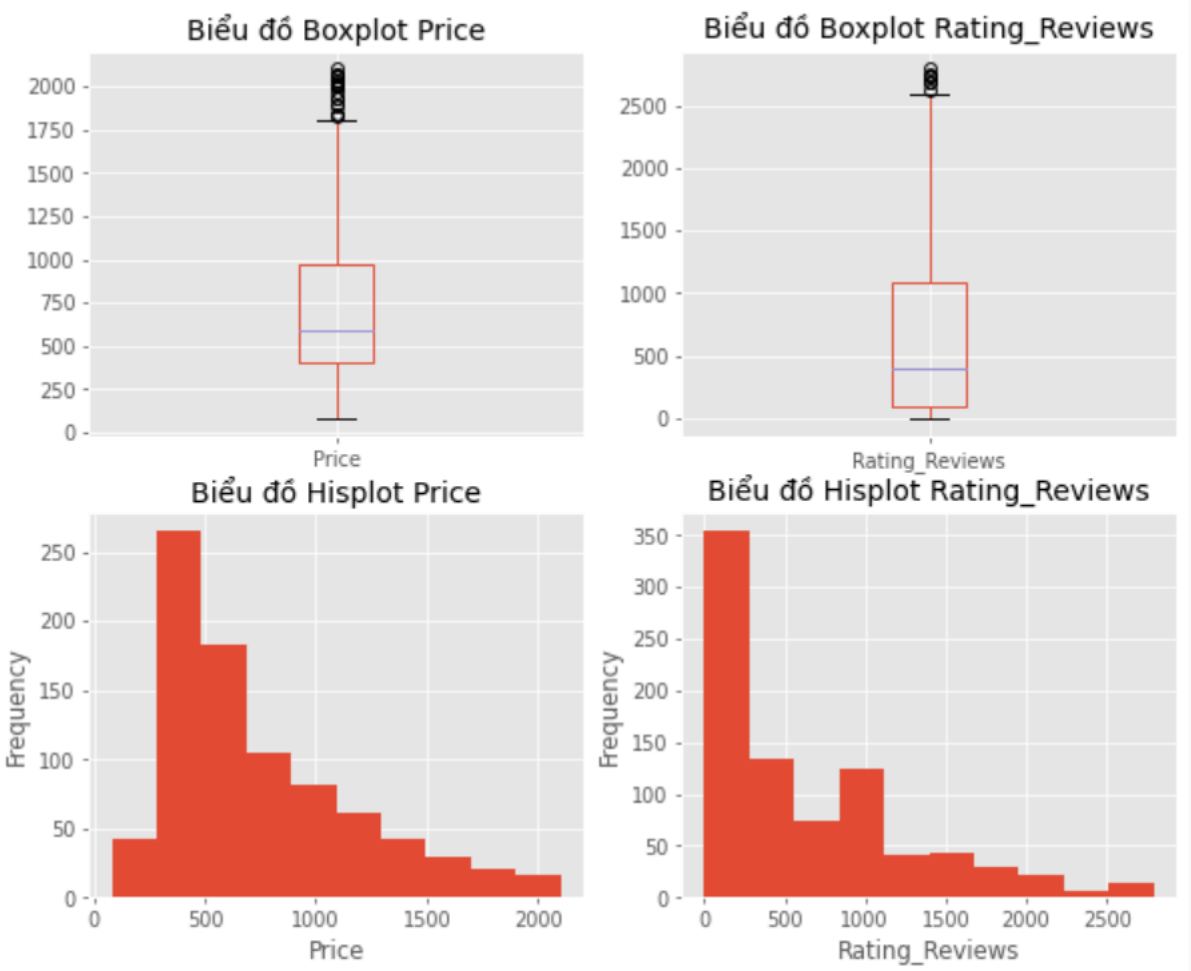
- Đầu tiên, kiểm tra và xóa các dữ liệu trùng lặp trong tập dữ liệu.

- Biểu đồ Boxplot và Hisplot thể hiện các phân phối dữ liệu và tần suất xuất hiện dữ liệu theo dạng cột như sau (Hình 14).

Hình 14: Biểu đồ Boxplot và Hisplot trước khi xử lý ngoại lệ

- Dựa vào biểu đồ trên, ta thấy được dữ liệu còn xuất hiện nhiều ngoại lệ cũng với dữ liệu phân bố không đồng đều với sự chênh lệch tần suất lớn.

- Xử lý ngoại lệ bằng phương pháp IQR Outlier. Sau khi xử lý ngoại lệ, ta thu được dữ tập dữ liệu được thể hiện bởi biểu đồ Boxplot và Hisplot như sau (Hình 15).



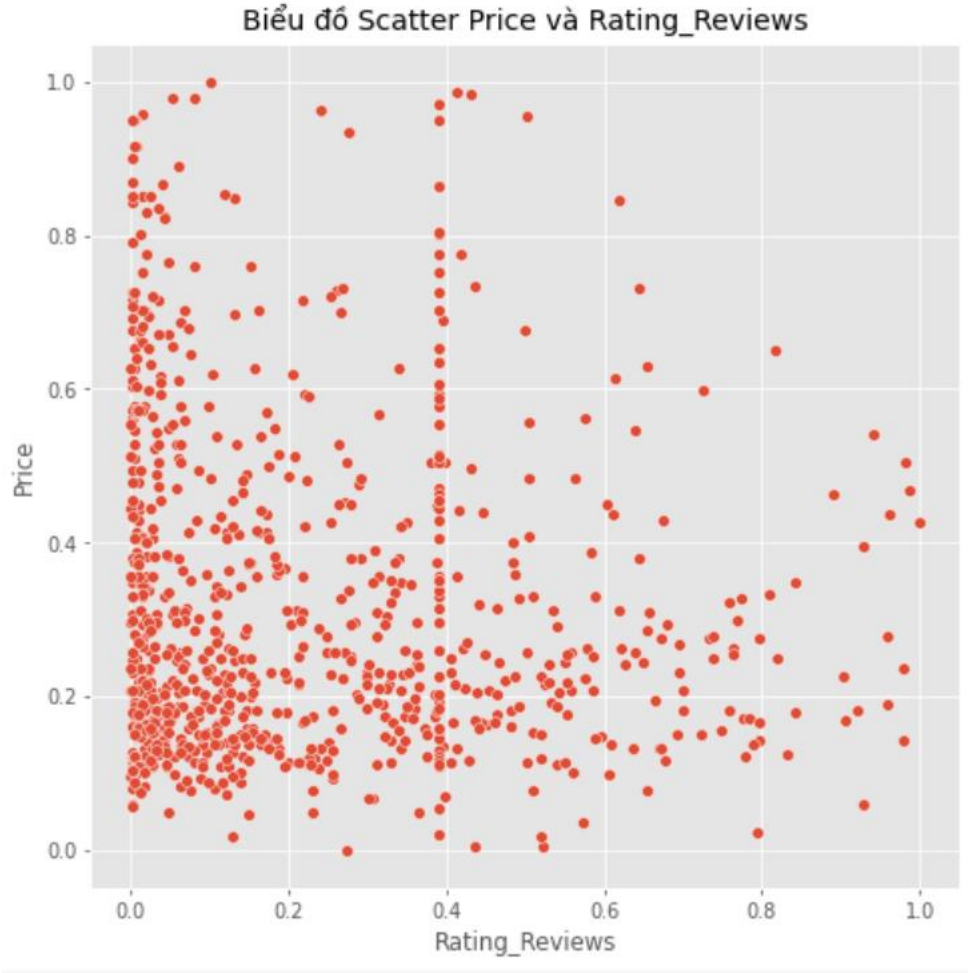
Hình 15: Biểu đồ Boxplot và Hisplot sau khi xử lý ngoại lệ.

- Dựa vào biểu đồ trên, ta thấy số lượng điểm ngoại lệ giảm đáng kể và sự chênh lệch tần suất xuất hiện của dữ liệu cũng giảm đi.

3.4 Chuẩn hóa dữ liệu

- Trong bài toán này, nhóm chúng em lựa chọn phương pháp chuẩn hóa MinMax để áp dụng cho tập dữ liệu của mình.

- Sau khi chuẩn hóa, sự phân tán dữ liệu được thể hiện qua biểu đồ Satter như sau (Hình 16).



Hình 16: Biểu đồ Scatter Price và rating_Reviews.

- Chuẩn hóa MinMax giúp tăng lên sự đồng đều của dữ liệu và độ quan trọng giữa các đặc trưng.

4. Mô hình hóa dữ liệu

4.1. Lựa chọn mô hình

- Đối với đề tài "Gợi ý đặt phòng khách sạn" này, nhóm chúng em lựa chọn 2 mô hình học không giám sát là: K-means clustering và Hierarchical clustering.

- Ý tưởng của nhóm em khi lựa chọn 2 mô hình này là để phân cụm những khách sạn theo 2 tiêu chí giá cả và đánh giá từ khách hàng (bao gồm reviews và rating), sau đó lựa chọn cụm giá trị có giá cả thấp và đánh giá cao để gợi ý cho khách hàng.

4.2. Mô tả mô hình

4.2.1. K-means clustering

- Trong thuật toán K-means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.

- Thuật toán này yêu cầu biết trước số cụm (số k) cần phân để tiến hành.

- Các bước của thuật toán:

B1: Khởi tạo ngẫu nhiên k tâm cụm.

B2: Lặp lại quá trình sau cho tới khi dừng:

+ Gán nhãn cho mỗi điểm dữ liệu bằng với nhãn của tâm cụm gần nhất.

+ Dịch chuyển dần dần tâm cụm tới trung bình của những điểm dữ liệu mà được phân về.

- API: KMeans(n_clusters=3, random_state=0)

+ n_clusters: là số cụm k.

+ random_state: là chỉ số random, khi được đặt bằng 1 số thì nó sẽ cho kết quả giống nhau với những lần chạy khác nhau.

4.2.2. Hierarchical clustering

- Theo phương pháp này, chúng tạo ra những biểu diễn phân cấp trong đó các cụm ở mỗi cấp của hệ thống phân cấp được tạo bằng cách hợp nhất các cụm ở cấp độ thấp hơn bên dưới. Ở cấp thấp nhất, mỗi cụm chứa một quan sát. Ở cấp cao nhất, chỉ có một cụm chứa tất cả dữ liệu. Các cấp của biểu diễn phân cụm được thể hiện trong đồ thị dendrogram.

- Có hai chiến lược phân chia chính phụ thuộc vào chiều di chuyển trên biểu đồ dendrogram:

+ Chiến lược hợp nhất (Agglomerative): xuất phát mỗi điểm là một cụm, việc phân cụm là thực hiện sát nhập các cụm nhỏ thành cụm to hơn (bottom–up).

+ Chiến lược phân chia (Divisive): tất cả các đối tượng/điểm là một cụm, việc phân cụm là thực hiện chia tách cụm lớn thành các cụm nhỏ hơn (top–down).

- Ở đề tài này chúng em sử dụng chiến lược hợp nhất.

- API: AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward').

+ n_clusters: số cụm.

+ affinity: công thức tính khoảng cách cho linkage.

+ linkage: tiêu chí đánh giá linkage nào được sử dụng.

4.3. Huấn luyện mô hình

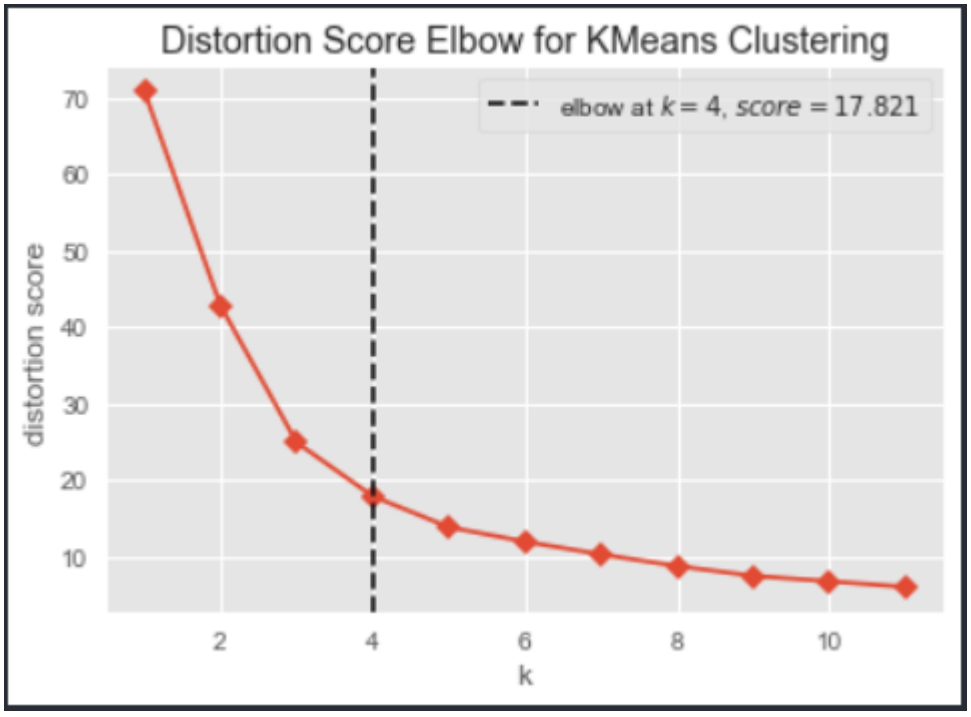
- Vì cả 2 mô hình đều thuộc loại học không có giám sát nên không thể chia tập dữ liệu thành các tập Huấn luyện/Xác thực/Kiểm thử.

4.3.1. K-means clustering

- Dữ liệu sau khi tiền xử lý sẽ được truyền vào mô hình thông qua API để tiến hành huấn luyện mô hình.

- Điều quan trọng nhất ở mô hình K-means clustering là phải xác định được số cụm k phù hợp. Đây là yếu tố quan trọng nhất ảnh hưởng đến kết quả bài toán. Có rất nhiều cách để xác định điều này, ở phạm vi dự án này chúng em sẽ sử dụng Elbow Method để đánh giá số cụm.

- Dưới đây là hình ảnh đồ thị Elbow Method thu được (Hình 17):



Hình 17: Đồ thị Elbow Method

- Từ đó ta có thể chọn được số k = 4.

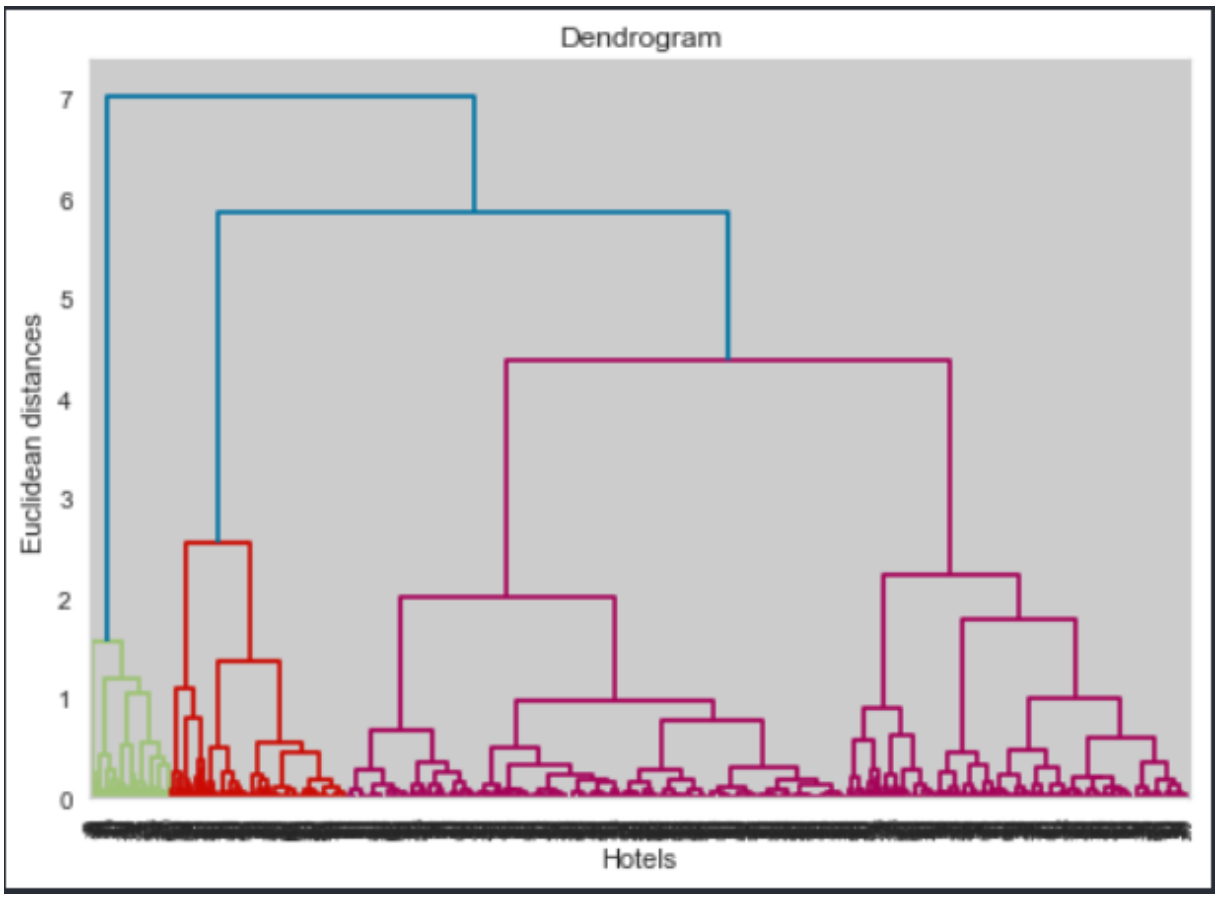
- Sau khi chọn được số k tối ưu, ta sẽ gọi API để xây dựng mô hình:

```
kmeans = KMeans(n_clusters=4, random_state=0).fit(data)
```

4.3.2. Hierarchical clustering

- Đối với mô hình Hierarchical clustering thì ta sẽ lựa chọn số k theo một cách tiếp cận khác dành riêng cho nó, được gọi là dendogram.

- Sau đây là đồ thị dendrogram thu được (Hình 18):



Hình 18: Đồ thị Dendrogram

- Ta sẽ tiến hành chọn số k bằng cách lựa chọn 1 đường thẳng dài nhất mà không có đường ngang nào cắt qua, sau đó ta sẽ đếm số đường thẳng song song với nó. Trên hình vẽ ta thu được k = 3.

- Sau đó ta sẽ tiến hành gọi API để huấn luyện mô hình:

cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')

4.4. Đánh giá mô hình

4.4.1. Silhouette Index

- Silhoutte Index là phương pháp tính toán kết hợp cả Cohesion và Separation.

Phương pháp này sẽ cho chúng ta biết các điểm dữ liệu được phân gọn trong cụm một cách tốt hay nằm ở ngoài rìa cụm. Nếu chỉ số này càng cao, tiến về 1 thì kết quả phân cụm càng chính xác và ngược lại.

- API sử dụng: hàm silhouette_score của thư viện sklearn.

- Kết quả:

+ K-means:

```
score = metrics.silhouette_score(data, kmeans.labels_, metric='euclidean')
score
```
✓ 0.1s

```
0.8632207901112803
```

+ Hierarchical:

```
score1 = metrics.silhouette_score(data1, cluster.labels_, metric='euclidean')
score1
```
✓ 0.9s

```
0.4678157865207517
```

=> Kết luận: mô hình K-means có chỉ số silhouette lớn hơn và tiến về 1 hơn so với Hierarchical (0.86… > 0.46…). Suy ra kết quả phân cụm của mô hình K-means là tốt hơn trong bài toán này.

4.4.2. DB Index

- DB Index là chỉ số giúp ta so sánh được mức độ phân tán của các cụm và độ tương đồng giữa chúng. Nếu chỉ số này càng thấp (tiến về 0) thì kết quả phân cụm càng tốt.

- API sử dụng: davies_bouldin_score của thư viện sklearn.

- Kết quả:

+ K-means:

```
db_score =  metrics.davies_bouldin_score(data, kmeans.labels_)
db_score
```
✓ 0.9s

```
0.2718518242816368
```

+ Hierarchical:

```
db_score1 =  metrics.davies_bouldin_score(data1, cluster.labels_)
db_score1
```
✓ 0.9s

```
0.7431461462580913
```

=> Kết luận: chỉ số DB Index của thuật toán K-means thấp hơn so với Hierarchical, suy ra kết quả phân cụm của K-means tốt hơn.

5. Kết luận

Bài toán đã giải quyết được vấn đề đặt ra đó là tìm ra các khách sạn phù hợp với hai tiêu chí: giá cart thấp và chất lượng tốt để đề xuất cho người dùng. Đồng thời, bài toán đã so sánh và đánh giá giữa hai thuật toán để đạt được kết quả tốt nhất.

Tuy nhiên, cần mở rộng nguồn dữ liệu cũng như cải thiện về các thuật toán trích xuất đặc trưng và thuật toán dùng để gợi ý. Đồng thời, sử dụng nhiều phương pháp so sánh, đánh giá thuật toán hơn để có được sự chính xác cao nhất và thu được kết quả tốt nhất.

6. Tài liệu tham khảo

[1] Youtube, "[Web Scraping] Lập trình Bot kéo Dữ liệu Người dùng Linkedin (Python & BeautifulSoup) cực đơn giản", https://www.youtube.com/watch?v=hfnBswCe4QE, June 3, 2022

[2] scikitLearn, "sklearn.cluster.KMeans", https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html?fbclid=IwAR2ndKamZYuohEjR3iVoYQ8vCGUKjMxKC-DL3AySeXgDQtu5iF_CP4kTjTY, June 20, 2022

[3] scikitLearn, "sklearn.cluster.AgglomerativeClustering", https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html?fbclid=IwAR3tGdvA-AafqiIS7sN612OR5lPYZjQjtzpM8gUPjE9AblsOBuCK5vw9nYU, June 20, 2022

[4] Funda, "Bài 4: K-means Clustering" https://machinelearningcoban.com/2017/01/01/kmeans/?fbclid=IwAR3ZtImRdsy6V_S_fL5-KfA0CVVpYIhHpoM-xWZjS74DLcy2Cn31prlZRJ8, Apr 20, 2022

[5] BigData, "Các phương pháp đánh giá thuật toán Clutering" https://bigdatauni.com/tin-tuc/cac-phuong-phap-danh-gia-trong-thuat-toan-clustering.html?fbclid=IwAR3AYVb7OkvWG7t7A0cfswG1nPE-IQp_UUdx_NwP6QZ9Da_xQb4D5fmCeUI, June 22, 2022

[6] BigData, "Feature Engineering – How to Detect and Remove Outliers (with Python Code)", https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/, June 18, 2022