TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

DỰ ĐOÁN GIÁ PHÒNG BOOKING DU LỊCH

Giảng viên hướng dẫn: TS. Ninh Khánh Duy

HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN	ĐIỂM BẢO VỆ
Trương Thị Mỹ Duyên	19N13	
Dương Anh Tuấn		
Phan Thị Thu Sương		

ĐÀ NẴNG, 06/2022

TÓM TẮT

Dự án chúng em lựa chọn có tên là “Dự đoán giá phòng booking du lịch”. Để giải quyết bài toàn chúng em sử dụng và so sánh hai thuật toán K-nearest neighbor và Liner Regression để dự đoán giá của một phòng khách sạn dựa trên các yếu cố như địa chỉ của khách sạn, loại khách sạn, diện tích phòng, rating, số lượng review,…Các kĩ thuật Fearture Engineering được sử dụng bao gồm phương pháp mã hóa dự liệu LabelEncoder, xử lí ngoại lệ (Outliers), chuẩn hóa dữ liệu bằng phương pháp Min-Max Scaling. Đồng thời sử dụng RMSE, MAE để lựa chọn đánh giá mô hình

Tuy nhiên trong quá trình thực hiện, để tìm kiếm được kết quả tốt nhất, chúng em đã sử dụng thêm mô hình thuật toán RandomForest và matrics Variance và và thu được kết quả khả quan như sau:

- Kết quả dự đoán chính xác lên đến 81% khi sử dụng RandomForest, cao hơn khi sử dụng K-nearest neighbor và Liner Regression

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá theo 3 mức
Trương Thị Mỹ Duyên	- Crawl dữ liệu - Mô hình hóa dữ liệu bằng Linear Regression, RandomForest	- Đã hoàn thành - Đã hoàn thành
Dương Anh Tuấn	- Crawl dữ liệu - Mô hình hóa dữ liệu bằng K-Nearest Neighbor	- Đã hoàn thành - Đã hoàn thành
Phan Thị Thu Sương	- Lựa chọn đặc trưng, làm sạch và chuẩn hóa dữ liệu, mã hóa và trực quan hoá dữ liệu	- Đã hoàn thành

MỤC LỤC

1. Giới thiệu

Bài toán đặt ra: Dự đoán giá phòng booking du lịch bằng các mô hình K-Nearest Neighbor, Linear Regression và RandomForest.

2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

a. **Nguồn dữ liệu:**
https://www.booking.com/index.vi.html?label=gen173nr-1DCAEoggI46AdIM1gEaPQBiAEBmAEquAEXyAEM2AED6AEBiAIBqAIDuAKYxuKVBsACAdICJDQ4MDBlM2RlLTNiYzUtNGQ2NS1iNzMwLTgxYWUzMWIwZTg2M9gCBOACAQ&sid=72bff0a7eb9e51985d80010fc4502590&keep_landing=1&sb_price_type=total&

b. **Ngôn ngữ thu nhập:** python

c. **Công cụ thu nhập dữ liệu:** Visual studio, Google Chorme

d. **Cách thức thu nhập:**
Sử dụng thư viện BeautifulSoup của python để crawl dữ liệu từ html. Dữ liệu sau khi được làm đẹp bằng BeautifulSoup , dùng css selector để đi tìm các element chứa dữ liệu cần thu nhập

e. **Đầu vào và đầu ra của quá trình thu nhập:** Chạy file crawlbookings.py
Đầu vào: link nguồn dữ liệu
Đầu ra: file rawdata.csv

f. Ví dụ minh họa:

Sau khi nhập link nguồn dữ liệu vào dòng 12 của file crawlbookings.py và chạy chương trình sẽ được kết quả như hình dưới. Số lượng mẫu thu nhập gồm 1079 hàng và 8 cột

```
type_hotel,location,rating,reviewer,distance,type_bed,area,price
0,hotel,"
85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, Việt Nam
","9,0", · 571 đánh giá,"
4,7 km
","
1 giường đôi lớn

",18 m²,"
VND 1.200.000
"
1,hotel,"
85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, Việt Nam
","9,0", · 571 đánh giá,"
4,7 km
","
2 giường đơn

",25 m²,"
VND 1.140.000
"
```

Figure 1: Ví dụ thu nhập dữ liệu

2.2 Mô tả dữ liệu

Dữ liệu thu nhập được bao gồm 1079 mẫu, với 8 đặc trưng:

Table 1: Mô tả dữ liệu

Đặc trưng	Mô tả	Kiểu dữ liệu	Số dữ liệu trống
type_hotel	Loại khách sạn (hotel, apartment, local hotel)	String	0
location	Địa chỉ của hotel	String	0
rating	Điểm đánh giá của khách sạn (tối đa 10 điểm)	String	0

reviewer	Số lượng đánh giá của khách sạn	String	0
distance	Khoảng cách từ khách sạn đến trung tâm thành phố	String	0
type_bed	Số lượng và loại giường có trong phòng (giường đôi, giường đôi …)	String	276
area	Diện tích phòng	String	0
price	Giá phòng	String	0

Kết quả crawl:

	type_hotel	location	rating	reviewer	distance	type_bed	area	price
0	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n1 giường đôi lớn\n\n	18 m^2	\nVND 1.200.000\n
1	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n2 giường đơn\n\n	25 m^2	\nVND 1.140.000\n
2	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n2 giường đơn\n\n	28 m^2	\nVND 1.400.000\n
3	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n1 giường đôi lớn\n\n	35 m^2	\nVND 2.000.000\n
4	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n1 giường đơn\n\n	35 m^2	\nVND 2.100.000\n
...	...	...	...	...	...	...	...	...
1119	apartment	\nPhố Minh Khai Park Premium, Times City, Quận...	7,7	· 3 đánh giá	\n5,3 km\n	NaN	Căn hộ nguyên căn	\nVND 1.960.000\n
1120	apartment	\n11 Ngõ Xóm Hà Hồi, Quận Hoàn Kiếm, Hà Nội,...	0	0	0	NaN	Căn hộ nguyên căn	\nVND 366.300\n
1121	apartment	\n2 Phố Phạm Văn Bạch, Cau Giay, Hà Nội, Viê...	0	0	0	NaN	Căn hộ nguyên căn	\nVND 1.377.000\n
1122	apartment	\n2 Phố Phạm Văn Bạch, Cau Giay, Hà Nội, Viê...	0	0	0	NaN	Căn hộ nguyên căn	\nVND 1.458.000\n
1123	apartment	\n47 Đường Nguyễn Tuân 47 Đường Nguyễn Tuân, Q...	10	· 4 đánh giá	0	NaN	Căn hộ nguyên căn	\nVND 1.200.000\n

1079 rows × 8 columns

Figure 2: Kết quả crwal dữ liệu

3. Trích xuất đặc trưng

a. **Làm sạch dữ liệu:** Dữ liệu thu nhập được còn rất nhiều các khoảng trống và kí tự đặc biệt không cần thiết

	type_hotel	location	rating	reviewer	distance	type_bed	area	price
0	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n1 giường đôi lớn\n\n	18 m²	\nVND 1.200.000\n
1	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n2 giường đơn\n\n	25 m²	\nVND 1.140.000\n
2	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n2 giường đơn\n\n	28 m²	\nVND 1.400.000\n
3	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	· 571 đánh giá	\n4,7 km\n	\n1 giường đôi lớn\n\n	35 m²	\nVND 2.000.000\n
4	hotel	\n85 Ma May Street, Quận Hoàn Kiếm, Hà Nội, ...	9,0	571 đánh giá	\n4,7 km\n	\n1 giường đơn\n\n	35 m²	\nVND 2.100.000\n
...	...	...	...	...	...	...	...	...
1119	apartment	\nPhố Minh Khai Park Premium, Times City, Quận...	7,7	· 3 đánh giá	\n5,3 km\n	NaN	Căn hộ nguyên căn	\nVND 1.960.000\n
1120	apartment	\n11 Ngõ Xóm Hà Hồi, Quận Hoàn Kiếm, Hà Nội,...	0	0	0	NaN	Căn hộ nguyên căn	\nVND 366.300\n
1121	apartment	\n2 Phố Phạm Văn Bạch, Cau Giay, Hà Nội, Viê...	0	0	0	NaN	Căn hộ nguyên căn	\nVND 1.377.000\n
1122	apartment	\n2 Phố Phạm Văn Bạch, Cau Giay, Hà Nội, Viê...	0	0	0	NaN	Căn hộ nguyên căn	\nVND 1.458.000\n
1123	apartment	\n47 Đường Nguyễn Tuân 47 Đường Nguyễn Tuân, Q...	10	· 4 đánh giá	0	NaN	Căn hộ nguyên căn	\nVND 1.200.000\n

Figure 3: Các đặc trưng

- **Xóa các kí tự không cần thiết:**

- Xóa kí tự xuống dòng và khoảng trắng ở các đặc trưng: distance, location, price, type_bed,..
- Xóa kí tự "giường" ở đặc trưng giường
- Xóa kí tự "đánh giá" ở đặc trưng reviewer
- Xóa kí tự "km" ở đặc trưng distance
- Xóa kí tự "VND" ở đặc trưng price
- Xóa kí tự "m²" ở đặc trưng area

	type_hotel	location	rating	reviewer	distance	type_bed	area	price
0	hotel	Quận Hoàn Kiếm	9.0	571	4.7	1 đôi lớn	18	12000
1	hotel	Quận Hoàn Kiếm	9.0	571	4.7	2 đơn	25	11400
2	hotel	Quận Hoàn Kiếm	9.0	571	4.7	2 đơn	28	14000
3	hotel	Quận Hoàn Kiếm	9.0	571	4.7	1 đôi lớn	35	20000
4	hotel	Quận Hoàn Kiếm	9.0	571	4.7	1 đơn	35	21000
5	hotel	Quận Hoàn Kiếm	9.4	3	1.9	1 đôi	22	26832
6	hotel	Quận Hoàn Kiếm	9.4	3	1.9	1 đôi cực lớn	25	30492
7	hotel	Quận Hoàn Kiếm	9.4	3	1.9	1 đôi lớn	25	30052
8	hotel	Quận Hoàn Kiếm	9.4	3	1.9	2 đôi lớn	28	34151
9	hotel	Quận Hoàn Kiếm	9.4	3	1.9		28	33272

Figure 4: Dữ liệu sau khi xóa các kí tự không cần thiết

- **Thay thế các dữ liệu không mong muốn:**

  Ở đặc trưng area (diện tích) có các dữ liệu không mong muốn như “Căn hộ nguyên căn”, “Studio nguyên căn”, nên cần tiến hành thay các giá trị này thành giá trị số nguyên bằng cách :

- Các mẫu có area là “Căn hộ nguyên căn”: Thay bằng trung bình area của các mẫu có type_hotel là “apartment”
- Các mẫu có area là “Studio nguyên căn”: Thay bằng giác trị ngẫu nhiên trong khoảng 30 đến 70

- **Đặc trưng distance (thay thế các mẫu có distance = 0)**

- Lọc ra dữ liệu trống theo các quận khác nhau
- Lọc ra các dữ liệu có giá trị theo các quận
- Tính giá trị mean của từng quận
- Thay thế các dữ liệu bằng 0 bởi giá trị mean ứng với từng quận

- **Đặc trưng area (thay thế các mẫu có area = 0)**

  Thay thế các dữ liệu trống bằng phương pháp random từ các giá trị còn lại trong cột area

b. Xử lí dữ liệu trống: Đặc trưng type_bed có 276 mẫu trống

Thay thế các dữ liệu trống nãy bằng phương pháp random ngẫu nhiên các giá trị còn lại trong đặc trưng type_bed

c. Mã hóa dữ liệu:

Tiến hành mã hóa các đặc trưng type_hotel, location, type_bed bằng phương pháp LabelEncoder

d. Trực quan hóa dữ liệu

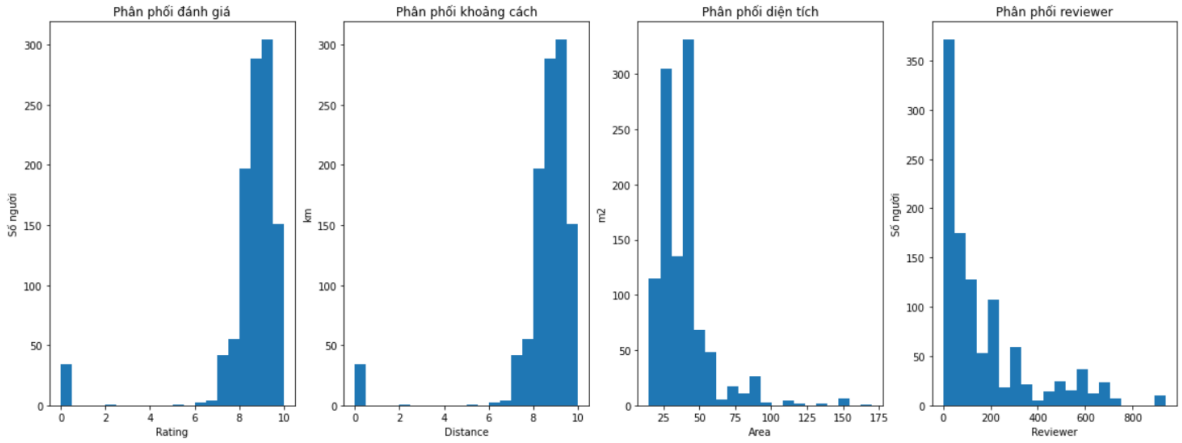

Figure 5: Biểu đồ Histogram

Dựa vào Figure 5 ta có thể thấy, các đặc trưng đều không tuân theo phân bố chuẩn.

- Rating phân bố từ khoảng từ 6 đến 10 điểm, nhiều nhất ở khoảng từ 8 đến 10 điểm
- Distance phân bố chủ yếu ở khoảng 7 đến 10 km, nhiều nhất ở khoảng từ 8 đến 10 km
- Area phân bố chủ yếu ở khoảng từ 18 đến 100 m^2 , nhiều nhất ở khoảng 25 đến 40 m^2
- Số lượng Reviewer phân bố chủ yếu từ 0 đến 700 lượt review,tệp trung chủ yếu ở khoảng dưới 200 review, có nhiều phòng không có lượt review nào

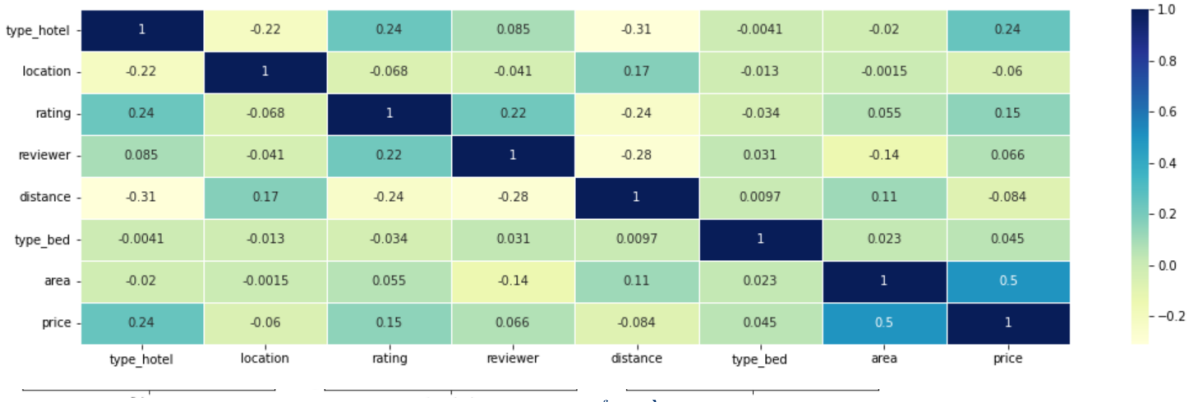

Figure 6: Biểu đồ Heatmap

Biểu đồ Heatmap cho ta thấy sự tương quan giữa các đặc trưng với nhau

- Đặc trưng price phụ thuộc nhiều nhất vào đặc trưng area (0,5), tiếp đến lần lượt là type_hotel (0.24), rating (0.15), reviewer (0.066), type_bed(0.045)
- Đặc trưng price không phục thuộc vào các đặc trưng location, distance

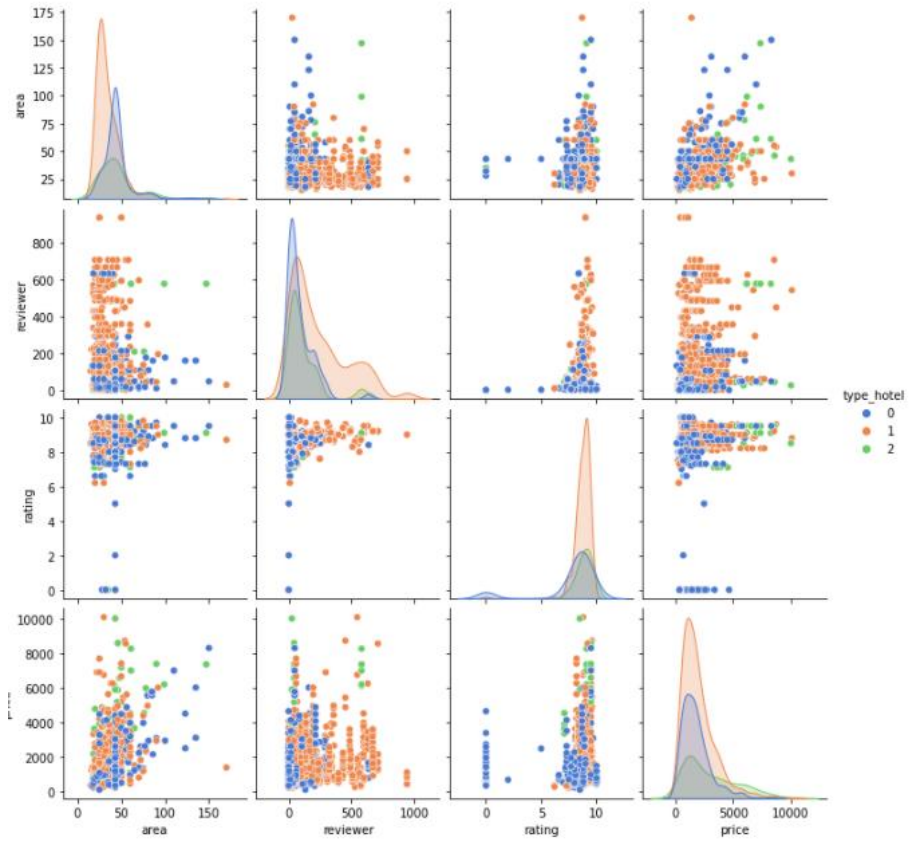

Figure 8: Biểu đồ Pairplot

Dựa vào Figure 8, ta thấy price không phụ thuộc nhiều vào 1 đặc trưng cụ thể nào.

Về tương quan giữa các cột, ta thấy price có kiểu phân tán không theo mô hình tuyến tính, vì thế khó có thể dựa vào duy nhất một đặc trưng nào để dự đoán giá phòng.

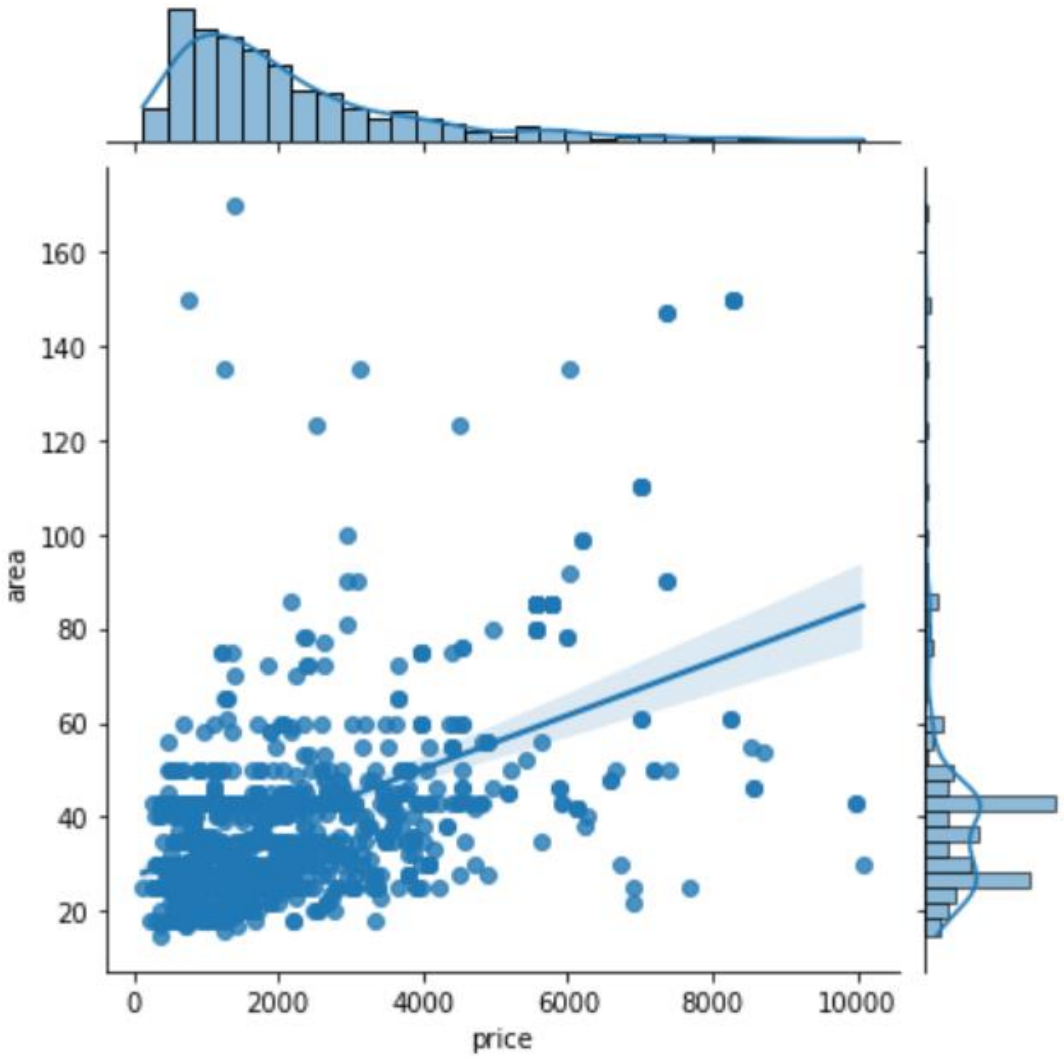

Figure 9: Biểu đồ Jointplot

Figure 9 cho thấy sự tương quan giữa price và area.

- Diện tích phòng nằm trong khoảng 20 m^2 đến 40 m^2 có giá trong tầm 50 nghìn tới 4 triệu.
- Giá tập trung mạnh ở tầm 5 trăm nghìn.
- Khu vực có sự phân tách thành 2 búp như đồ thị năm dọc bên phải.

e. Chuẩn hóa dữ liệu: Bằng phương pháp Min-Max Scaling

	type_hotel	location	rating	reviewer	distance	type_bed	area	price
0	1	6	9.0	483	4.700000	4	4.700000	1200.0
1	1	6	9.0	483	4.700000	10	4.700000	1140.0
2	1	6	9.0	483	4.700000	10	4.700000	1400.0
3	1	6	9.0	483	4.700000	4	4.700000	2000.0
4	1	6	9.0	483	4.700000	5	4.700000	2100.0

Figure 10: Dữ liệu sau khi chuẩn hóa

f. Xử lí ngoại lệ

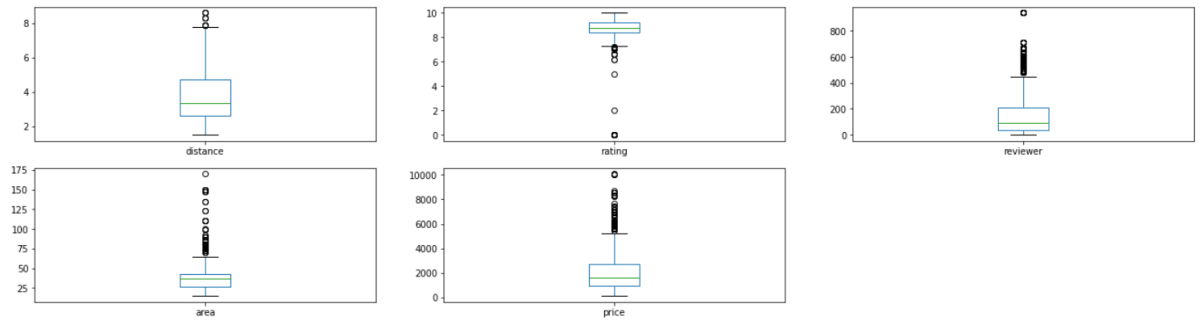

Figure 11: Số lượng ngoại lệ trước khi xử lí

Có nhiều ngoại lệ ở đặc trưng distance, rating, reviewer, area , distance, price

Kết quả sau khi xử lí ngoại lệ

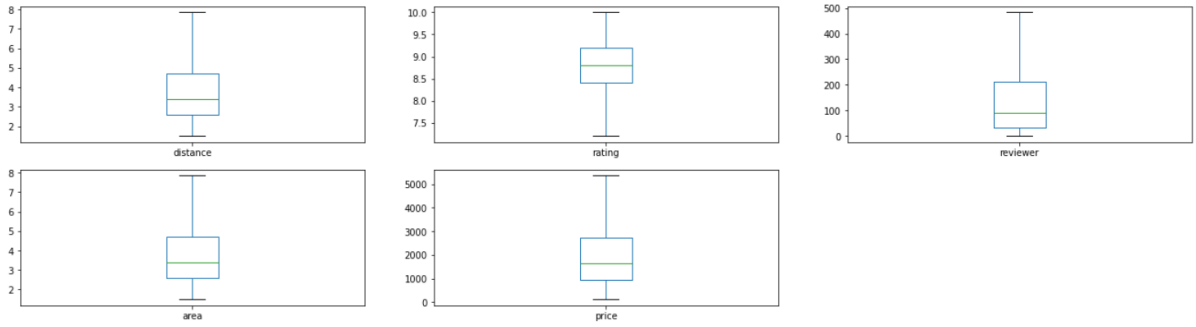

Figure 12: Kết quả sau khi xử lí ngoại lệ

g. Chuẩn hóa dữ liệu

Chuẩn hóa dữ liệu bằng kĩ thuật chuẩn hóa Min-max Scaling

4. Mô hình hóa dữ liệu

4.1 Các mô hình/ thuật toán

a. K-nearest neighbor [5]

K-nearest neighbors là thuật toán học máy có giám sát, đơn giản và dễ triển khai. Thường được dùng trong các bài toán phân loại và hồi quy

Thuật toán K-nearest neighbor cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm củng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp. Đây là 3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

b. Linear Regression [6]

Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X).

Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục

Thuật toán hồi quy tuyến tính (linear regression) thuộc vào nhóm học có giám sát (supervised learning) là được mô hình hoá bằng:

$$y(\mathbf{x}, \theta) = \theta^\top \phi(\mathbf{x})$$

Khi khảo sát tìm tham số của mô hình ta có thể giải quyết thông qua việc tối thiểu hoá hàm lỗi (loss function):

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta^\top \phi(\mathbf{x}_i) - y_i \right)^2$$

Hàm lỗi này thể hiện trung bình độ lệch giữa kết quả ước lượng và kết quả thực tế. Việc lấy bình phương giúp ta có thể dễ dàng tối ưu được bằng cách lấy đạo hàm vì nó có đạo hàm tại mọi điểm! Qua phép đạo hàm ta có được công thức chuẩn (normal equation) cho tham số:

$$\hat{\theta} = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

c. RandomForest [7]

Thuật toán Random Forest mình sẽ xây dựng nhiều cây quyết định bằng thuật toán Decision Tree, tuy nhiên mỗi cây quyết định sẽ khác nhau (có yếu tố random). Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định.

- Xây dựng thuật toán Random Forest

- Lấy ngẫu nhiên n dữ liệu từ bộ dữ liệu với kĩ thuật Bootstrapping, hay còn gọi là random sampling with replacement. Tức khi mình sample được 1 dữ liệu thì mình

không bỏ dữ liệu đấy ra mà vẫn giữ lại trong tập dữ liệu ban đầu, rồi tiếp tục sample cho tới khi sample đủ n dữ liệu. Khi dùng kĩ thuật này thì tập n dữ liệu mới của mình có thể có những dữ liệu bị trùng nhau.

- Sau khi sample được n dữ liệu từ bước 1 thì mình chọn ngẫu nhiên ở k thuộc tính (k < n). Giờ mình được bộ dữ liệu mới gồm n dữ liệu và mỗi dữ liệu có k thuộc tính.
- Dùng thuật toán Decision Tree để xây dựng cây quyết định với bộ dữ liệu ở bước

d. So sánh giữa giảm chiều dữ liệu PCA và không giảm chiều dữ liệu PCA

```
PCA
                Variance          MAE         RMSE
Algorithms
Linear          0.079514  1136.589779  1415.022060
KNeighbors      0.625172   649.483565   909.634724
RandomForest    0.715755   535.422571   786.884358
-----------------------------------------------
No PCA
                Variance          MAE         RMSE
Algorithms
Linear          0.093591  1145.263322  1403.900058
KNeighbors      0.582283   708.150231   966.588500
RandomForest    0.831293   418.323128   606.138127
```

Figure 13: So sánh Score khi sử dụng PCA và NoPCA

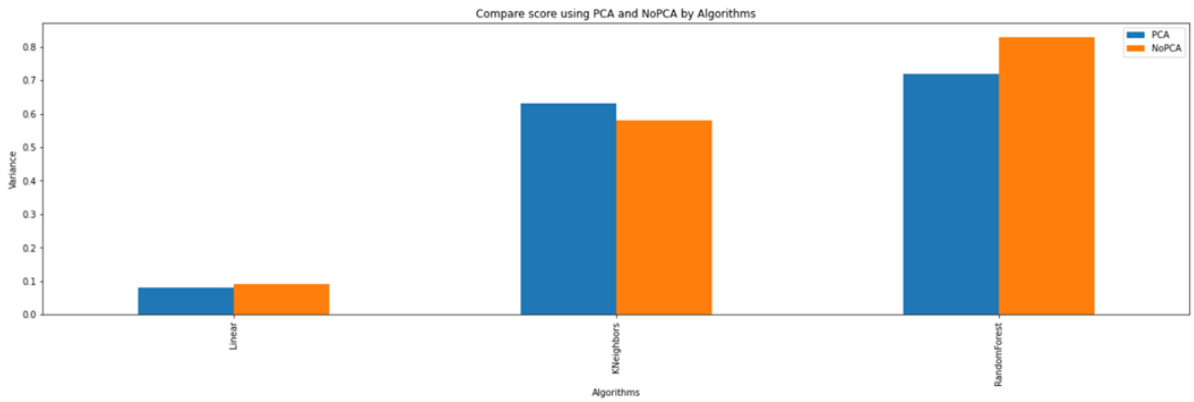

Figure 14: Biểu đồ so sánh Score khi sử dụng PCA và NoPCA

Dựa vào hai Figure 13, 14, mô hình thuật toán Liner Regression và RandomForest khi sử dụng giảm chiều dữ liệu thì Score sẽ thấp hơn khi không xử dụng PCA, và ngược lại với mô hình thuậ toán K-nearest neighbor

⇨ **Chọn không giảm chiều dữ liệu khi tiến hành dự đoán giá phòng khách sạn**

e. Chia dữ liệu thành các tập Huấn luyện/Xác thực/Kiểm thử

```
### Chia tap du lieu train,validation, test
X_train, X_val_test, y_train, y_val_test = train_test_split(X_data,y_target, test_size=0.3, random_state=0)
X_val, X_test, y_val, y_test = train_test_split(X_val_test,y_val_test, test_size=0.5, random_state=0)
```
✓ 0.5s

Figure 15: Chia tập dữ liệu

Tiến hành chia tập dữ liệu thành các tập Huấn luyện/Xác thực/Kiểm thử như Figure 15

f. Các metrics đánh giá mô hình

- **RMSE:** Căn bậc 2 của trung bình bình phương sai số
- **MAE:** là 1 metric đánh giá mô hình bằng cách tính trung bình giá trị tuyệt đối sai số giữa giá trị thực tế và giá trị dự đoán
- **Variance:** Phương sai là là phép đo mức chênh lệch giữa các số liệu trong một tập dữ liệu trong thống kê. Nó đo khoảng cách giữa mỗi số liệu với nhau và đến giá trị trung bình của tập dữ liệu

g. So sánh các mô hình

Algorithms	Variance	MAE	RMSE
Linear	0.076140	1050.173418	1303.150822
KNeighbors	0.397380	774.165432	1062.381014
RandomForest	0.749822	477.939452	682.325913

Figure 16: So sánh kết quả đánh giá mô hình dựa vào maxtics

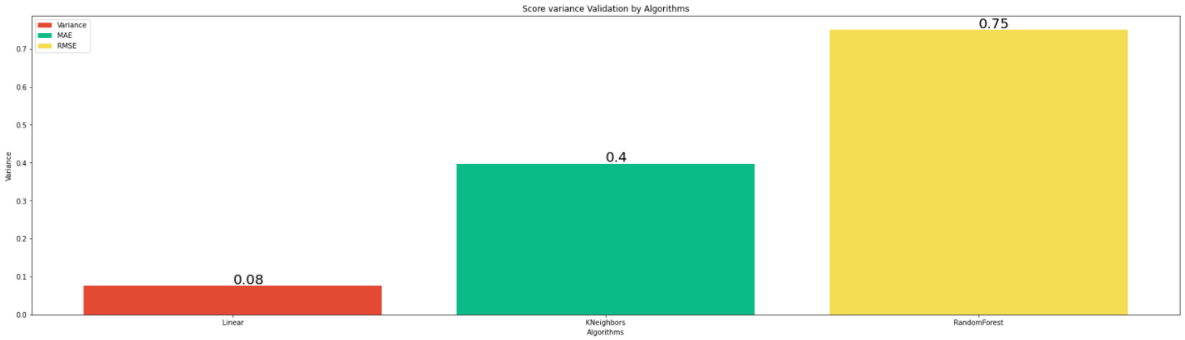

Figure 17: So sánh score variance giữa các thuật toán

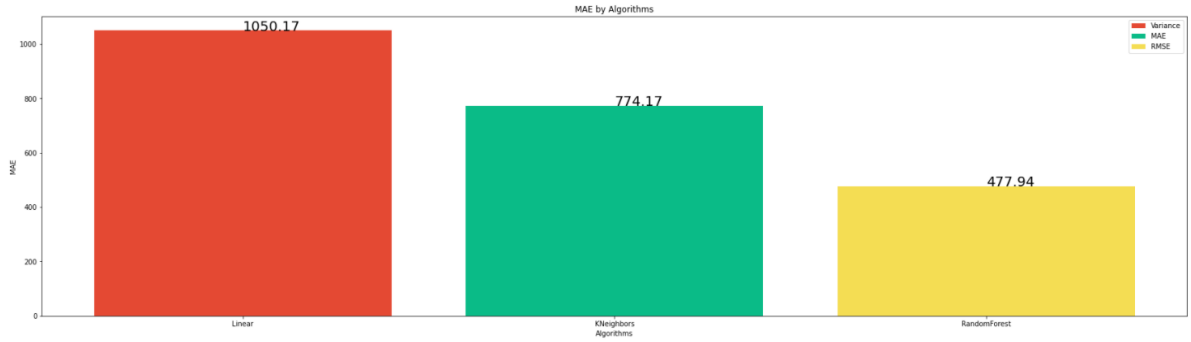

Figure 18: So sánh score MAE giữa các thuật toán

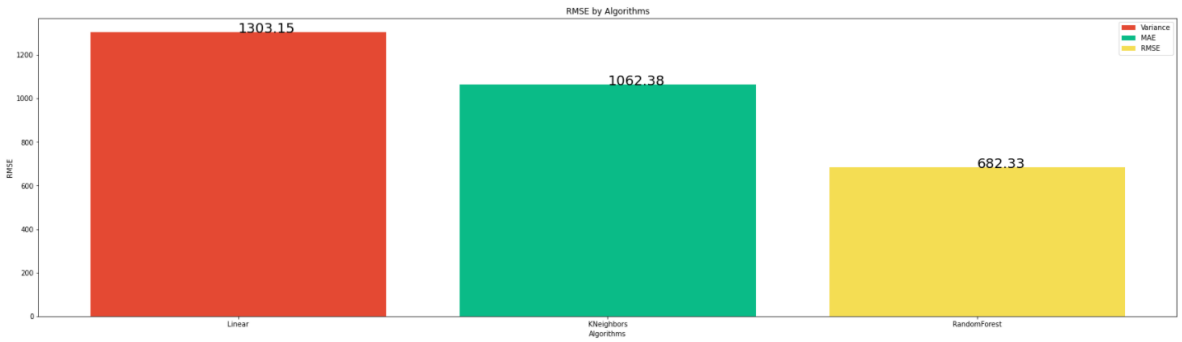

Figure 19: So sánh score RMSE giữa các thuật toán

Cả 3 metrics đánh giá mô hình (Variance, MAE, RMSE) đều cho thấy hiệu suất đánh giá mô hình của thuật toán RandomForest là cao nhất, tiếp theo là K-Nearest Neighbor, thấp nhất là mô hình thuật toán Liner Regression

Kết luận: Chọn mô hình RandomForestRegressor để đánh giá và dự đoán kết quả

h. Kết quả của mô hình RandomForestRegressor

```
Evaluate prediction based on variance :  81.04863375003359
```

Figure 21: Kết quả dự đóa dựa trên Variance

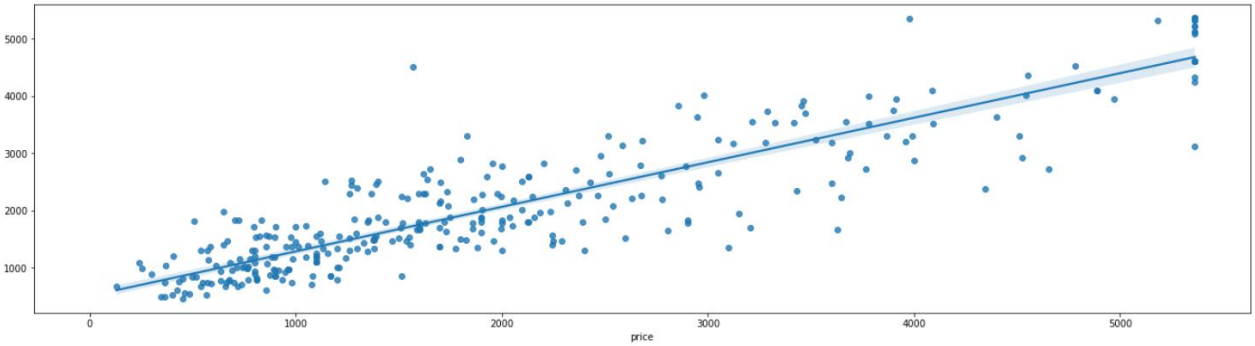

Figure 20: Biểu đồ hiển thị giá trị dự đoán và giá trị thực tế

Nhận xét: Kết quả dự đoán giá phòng khách sạn khi sử dụng mô hình RandomForestRegressor khá cao, xấp xỉ 81%

- Biểu đồ ở Figure 20 thể hiện độ chính xác giữa giá trị dự đoán và giá trị thực tế
- Mỗi điểm thể hiện sự tương quan giữa giá trị dự đoán và giá trị thực tế : Tập trung chủ yếu từ 500 nghìn đồng đến 1 triệu 500 nghìn đồng.
- Sự tuyến tính của giữa giá trị dự đoán và giá trị thực tế khá caom thể hiện độ chính xác của thuật toán cao

i. Cải tiến thuật toán RandomForestRegressor

- Tiến hành cải tiến thuật toán bằng siêu tham số sử dụng GridSearch CV

```
param_grid = {
            "n_estimators"      : [10,20,30],
            "max_features"      : ["auto", "sqrt", "log2"],
            "min_samples_split" : [2,4,8],
            "bootstrap": [True, False],
            }
```

Figure 22: Các siêu tham số được sử dụng

```
Best score: 0.7104506783102585
Score by before gridsearch:81.04863375003359
Score by after gridsearch:81.45680290710244
Cải thiện được: 0.4081691570688548
```

Figure 23: Kết quả cải tiến

Nhận xét: Khi sử dụng siêu tham số thì kết quả được cải thiện khoảng 4%

- Cải tiến bằng cách lựa chọn các đặc trưng có ảnh hưởng cao đối với giá tiền: type_hotel, area, rating, reviewer, price, type_bed

```
Score by before Choice:81.68498325343394
Score by after Choice:81.04863375003359
Cải thiện được: 0.6363495034003535
```

Figure 24: Kết quả cả tiến khi lựa chọn đặc trưng

Nhận xét: Khi lựa chọn ra các đặc trưng có ảnh hưởng đến giá phòng thì kết quả được cải thiện khoảng 6%

j. So sánh dữ liệu có min-max scaling và không min-max scaling

```
Minmax Scaling
              Variance       MAE      RMSE
Algorithms
Linear        0.093591  0.218844  0.268265
KNeighbors    0.582283  0.135317  0.184701
RandomForest  0.828552  0.081215  0.116746
-----------------------------------------
No Minmax Scaling
              Variance          MAE         RMSE
Algorithms
Linear        0.093591  1145.263322  1403.900058
KNeighbors    0.708969   587.808565   797.094498
RandomForest  0.828856   419.849728   610.483014
```

Figure 25: Kết quả so sánh có Min-max scaling và không Min-max scaling

5. Kết luận

- **Công việc đã làm:**
  Bước 1. Crawl dữ liệu
  Bước 2. Xử lý các giá trị không hợp lệ (area: theo dữ liệu string, distance==0, type_bed==",…)
  Bước 3. Mô tả dữ liệu
  Bước 4. Trích xuất đặc trưng: làm sạch, xử lý ngoại lệ, chuẩn hóa dữ liệu, giảm chiều dữ liệu...
  Bước 5. Áp dụng các model dự đoán tỉ lệ
  Bước 6. So sánh các tỉ lệ accuracy và các metric
  Bước 7. Chọn ra thuật toán phù hợp nhất
- **Kết quả đạt được:**

  1. Quá trình crawl data: Crawl được (1079,8)

  2. Quá trình so sánh model:

  Thông qua việc so sánh Variance, MAE, RMSE

  => Chọn được model RandomForestRegressor:

  - Có sử dụng MinMaxScaler
  - Không dùng PCA
  - Tỉ lệ score ~81%

  3. Quá trình cải thiện dữ liệu:

  Sử dụng thư viện để tìm siêu tham số

  => Có cải thiện được tuy nhiên không đáng kể

  - Tăng tỷ lệ lên tầm -0.5 -> 0.5

  Sử dụng thủ công tìm tham số

  => Có cải thiện được tuy nhiên không quá nhiều

  - Tăng tỷ lệ lên tầm -0.6 -> 0.6
- **Hướng phát triển:**
  - Crawl dữ liệu nhiều đặc trưng hơn.
  - Xử lý đặc trưng tại bước crawl dữ liệu .
  - Xử lý các giá trị không hợp lệ tại bước làm sạch dữ liệu theo computed sao cho đúng với các giá trị thực tế hơn.
  - Thử nghiệp với các kĩ thuật Feature engineering khác: OneHotEncode,…. và các mô hình thuật toán khác.

6. Tài liệu tham khảo

[1] **Visualizing statistical relationships**
https://seaborn.pydata.org/tutorial/relational.html

[2] **Car Price Prediction (Linear Regression - RFE)**
https://www.kaggle.com/code/goyalshalini93/car-price-prediction-linear-regression-rfe?fbclid=IwAR1MTNRxYTC2mhW5Fif68ZMuS85zSPvc-aNvxOHwFV_nzqTiBEkSjT9Cju0

[3] **Đánh giá mô hình hồi quy, machine learning (performance and predict)**
https://rpubs.com/nguyenngocbinhneu/performance_predict

[4] **Cross-Validation with Linear Regression**
https://www.kaggle.com/code/jnikhilsai/cross-validation-with-linear-regression/notebook

[5] **Thuật Toán K-Nearest Neighbors (KNN) Siêu Cơ Bản**
https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn

[6] **Hồi quy tuyến tính (Linear Regression)**
https://dominhhai.github.io/vi/2017/12/ml-linear-regression/

[7] Random Forest algorithm
https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html