TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

TÊN ĐỀ TÀI

Dự đoán giá phòng trọ/nhà trọ trên địa bàn thành phố Đà Nẵng, Việt Nam

Nhóm	1
Họ Và Tên Sinh Viên	Lớp Học Phần
Phan Minh Nhật	20.91
Nguyễn Văn Thanh Hoàng	
Trần Trọng Hiếu	

ĐÀ NẴNG, 05/2023

TÓM TẮT

Hiện nay trong khu vực địa bàn thành phố Đà Nẵng, Việt Nam, với sự gia tăng về nhu cầu đi học xa nhà của sinh viên việc tìm được một phòng trọ phù hợp với khả năng tài chính của bản thân là vấn đề cần quan tâm của sinh viên. Để có thể có sự chuẩn bị cần thiết, việc dự đoán giá phòng trọ theo nhu cầu của sinh viên là điều cần thiết.

Để giải quyết được nhu cầu này cần kết hợp các lĩnh vực khai phá dữ liệu, xử lý dữ liệu và học máy áp dụng vào giải quyết. Từ đó, có thể huấn luyện được các mô hình có độ tin cậy cao áp dụng vào hệ thống dự đoán giá trọ trong khu vực thành phố Đà Nẵng, Việt Nam.

Mục đích của đề tài là tìm hiểu tổng quan về bài toán dự đoán giá phòng trọ, tập trung và phân tích các phương pháp tiếp cận và xử lý dữ liệu chung của bài toán này. Đánh giá và so sánh hiệu suất của những mô hình áp dụng vào trong bài toán, từ đó có thể lựa chọn mô hình phù hợp với bài toán.

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)
Trần Trọng Hiếu	- Thu thập dữ liệu từ web - Làm sạch dữ liệu	- Đã hoàn thành - Đã hoàn thành
Nguyễn Văn Thanh Hoàng	- Tiền xử lí dữ liệu - Trực quan hóa dữ liệu	- Đã hoàn thành - Đã hoàn thành
Phan Minh Nhật	- Thu thập dữ liệu từ web - Mô hình hóa dữ liệu	- Đã hoàn thành - Đã hoàn thành

MỤC LỤC

Nội dung

1. Giới thiệu
1.1 Vấn đề cần giải quyết

Xây dựng hệ thống dự đoán giá thuê trọ trong khu vực thành phố Đà Nẵng, Việt Nam

- Đầu vào là thông tin về diện tích, vị trí của phòng trọ (quận, huyện)
- Đầu ra là giá tiền thuê phòng trọ đó trong 1 tháng

1.2 Giải pháp

Nghiên cứu ứng dụng các mô hình học máy vào. Từ đó có thể huấn luyện được các mô hình có độ tin cậy cao áp dụng vào hệ thống dự đoán giá trọ trong khu vực thành phố Đà Nẵng, Việt Nam

2. Thu thập và mô tả dữ liệu
2.1. Thu thập dữ liệu
2.1.1 Nguồn dữ liệu

Dữ liệu được thu thập từ 7 trang web khác nhau và tổng cộng thu được 10020 mẫu dữ liệu

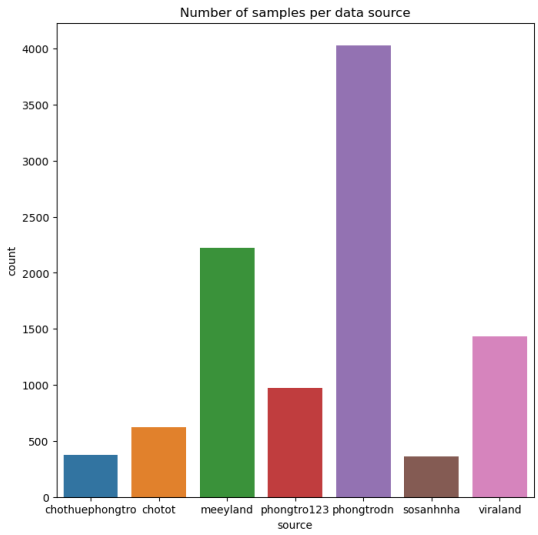


Figure 1: Số lượng mẫu dữ liệu từ 7 nguồn khác nhau

2.1.2 Công cụ thu thập
2.1.2.1 Sử dụng Selenium

Selenium là một công cụ giúp tự đống hóa quá trình sử dụng trình duyệt giống người dùng bình thường trên browser, từ việc truy cập vào trang chủ, nextpage, submit form,…

Cách thức sử dụng:

- Nạp ChromeDriver
- Đưa đường dẫn của trang web vào
- Phân tích trang web bằng cách bóc tách các thẻ HTML để lấy đường dẫn các tin cho thuê trọ
- Truy cập vào những đường dẫn này và bóc tách từng thuộc tính để lấy dữ liệu

2.1.2.2 Sử dụng Beautiful Soup

Beautiful Soup là một thư việc của Python hỗ trợ việc lấy dữ liệu từ HTML đơn giản và hiệu quả. Mặc dù tốc độ thu thập nhanh hơn khi sử dụng Selenium nhưng rất dễ dẫn tới hiện tượng crash các trang web. Về cách thức sử dụng, Beautiful Soup được sử dụng gần giống với Selenium trừ việc phải nạp ChromeDriver.

2.2. Mô tả dữ liệu

- Bộ dữ liệu gồm 2 bộ:
  - BigDataset: tổng cộng 9808 mẫu dữ liệu
  - SmallDataset: gồm 1000 mẫu dữ liệu lấy ngẫu nhiên từ BigDataset
- Đối với mỗi mẫu dữ liệu bao gồm 3 biến:
  - Price: kiểu dữ liệu float
  - Area: kiểu dữ liệu float
  - Location: kiểu dữ liệu string
- Số mẫu dữ liệu trống
  - Big Dataset

Biến	Price	Area	Location
Số lượng (mẫu)	444	1247	4313

  - Small Dataset

Biến	Price	Area	Location
Số lượng (mẫu)	50	132	428

- Phân bố của Area trên từng Location
  - Big Dataset

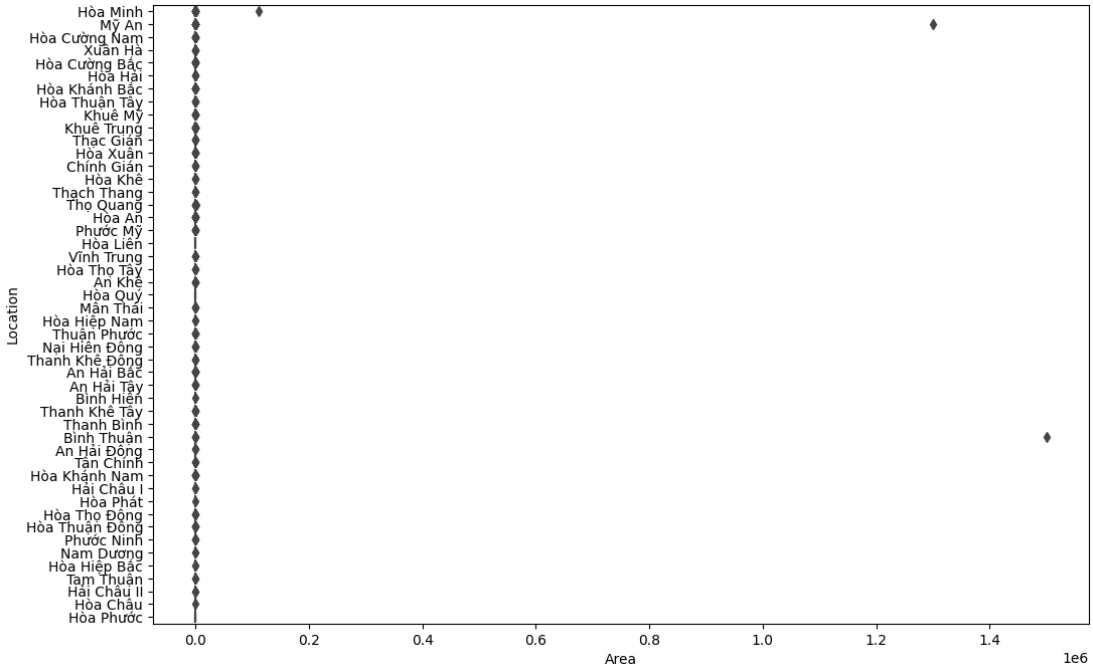


Figure 2. Phân bố của Area trên từng Location (big dataset)

- Small Dataset

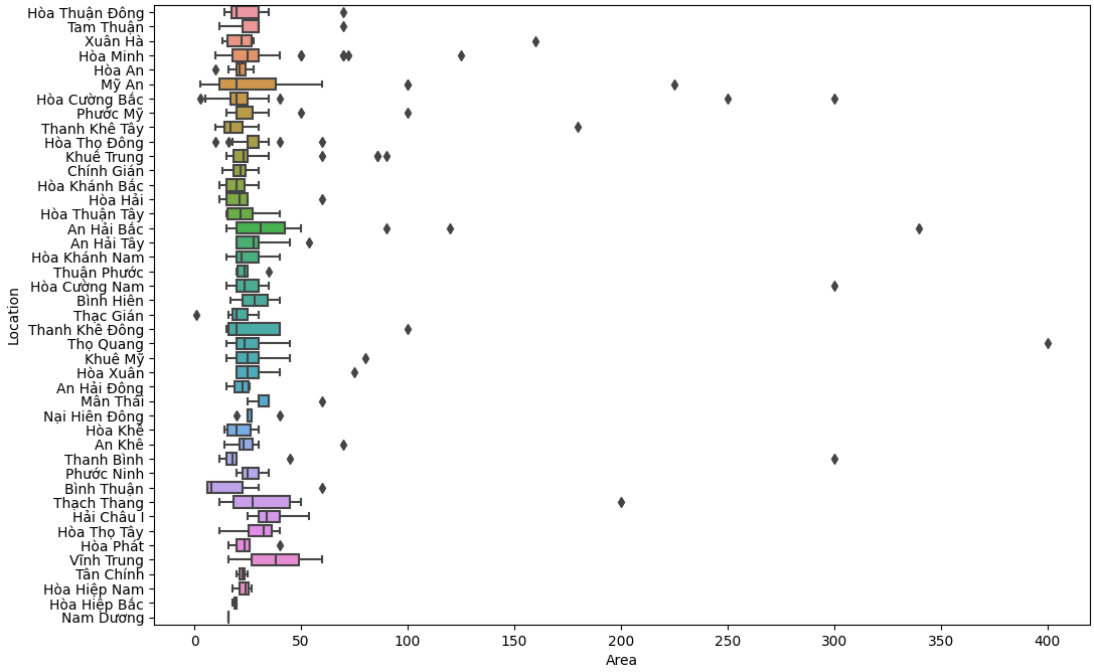


Figure 3. Phân bố của Area trên từng Location (small dataset)

- Phân bố của Price trên từng Location
  - Big Dataset

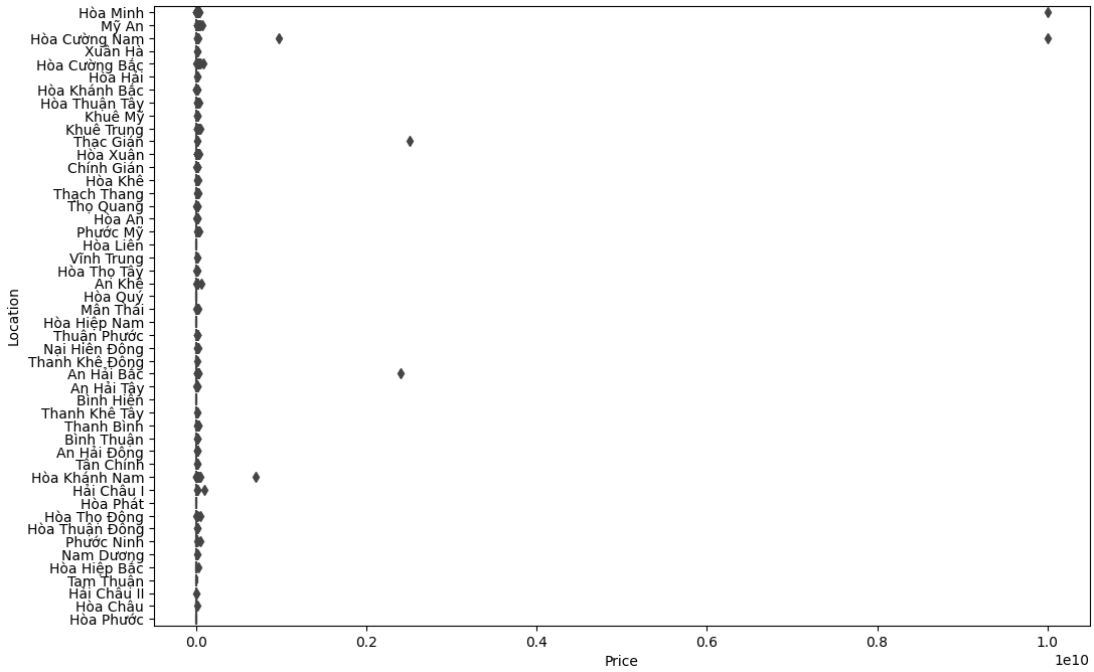


Figure 4. Phân bố của Price trên từng Location (big dataset)

- Small Dataset

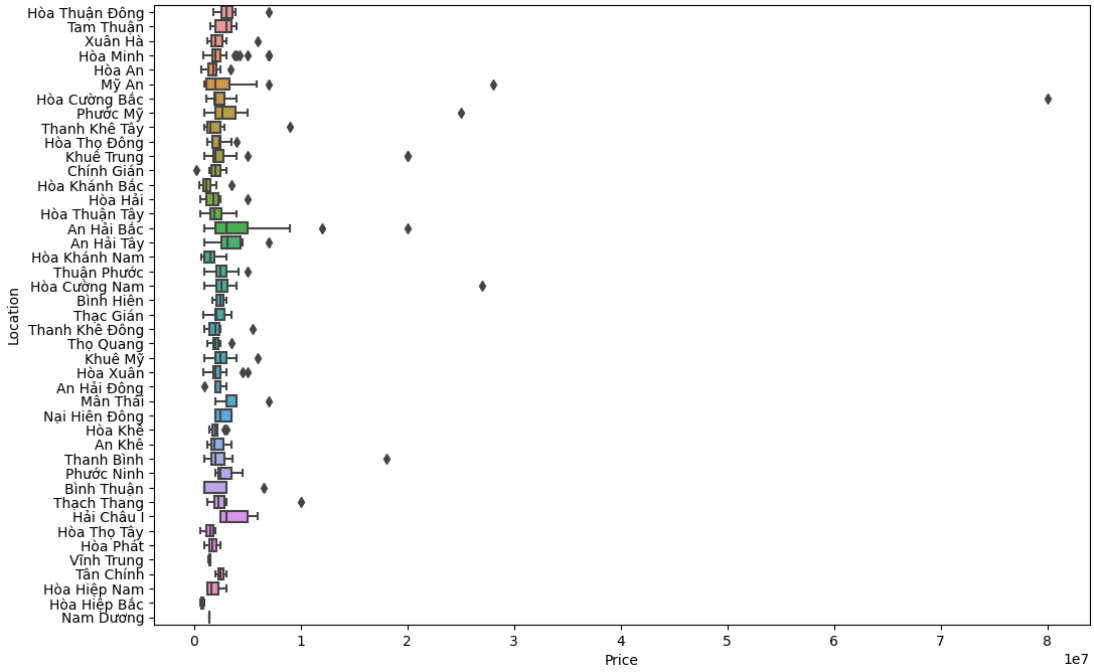


Figure 5. Phân bố của Price trên từng Location (small dataset)

- Hệ số tương quan giữa các biến

Figure 6. Biểu đồ ma trận hệ số tương quan các biến (big dataset)

Figure 7. Biểu đồ ma trận hệ số tương quan các biến (small dataset)

- Không gian đặc trưng
    - Big Dataset

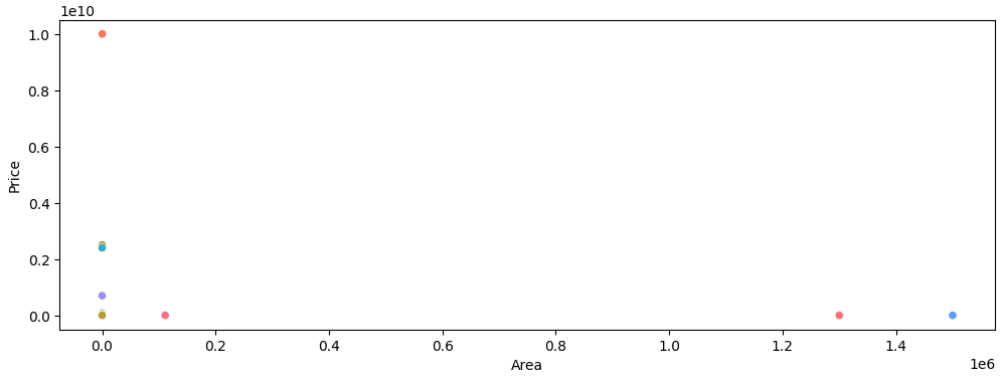


Figure 8. Không gian đặc trưng (big dataset)

    - Small Dataset

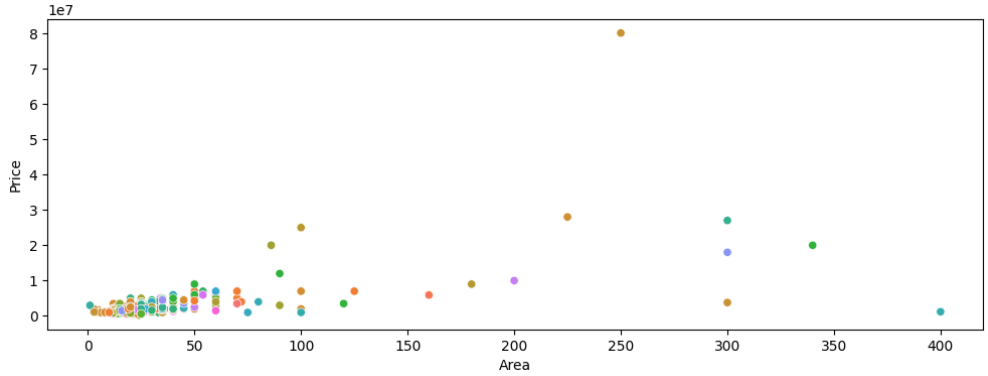


Figure 9. Không gian đặc trưng (small dataset)

- Kết luận chung
    - Ở big dataset:
        - Các giá trị ngoại lệ rất cao, dẫn tới khó quan sát phân bố của các đặc trưng
        - Độ tương quan giữa các đặc trưng riêng lẻ là rất thấp. Cho thấy rằng mức độ ảnh hưởng giữa các đặc trưng với nhau là rất thấp.
    - Ở small dataset:
        - Các giá trị ngoại lệ không quá cao, dữ liệu phân bố khá tập trung và đều nếu xét trên từng Location
        - Độ tương quan giữa các đặc trưng riêng lẻ không quá cao, độ tương quan giữa Price và Area là đáng chú ý nhất, thể hiện một mối quan hệ đồng biến nhỏ. Trong khi đó, tương quan giữa Price và Location, cũng như Area và Location rất thấp, cho thấy chúng ảnh hướng ít tới nhau.
- Đối với từng bộ dữ liệu, chia dữ liệu thành 2 phần: 80% cho tập huấn luyện và 20% cho tập kiểm thử.

3. Trích xuất đặc trưng

3.1 Lựa chọn đặc trưng

Sử dụng 2 đặc trưng cho việc huấn luyện, bao gồm: Area, Location.

Sử dụng biến Price được sử dụng làm biến mục tiêu.

3.2 Làm sạch dữ liệu

Đối với từng loại biến sẽ được làm sạch qua các công đoạn khác nhau:

- Đối với biến *Area*:
  - Loại bỏ các cụm từ dư thừa "Diện tích", sửa lại định dạng dấu thập phân (thay dấu "," bằng dấu ".")
  - Loại bỏ các mẫu dữ liệu thể hiện thông tin là "PN"
  - Đối với các mẫu dữ liệu là "KXĐ" hoặc 0 thì chuyển thành dữ liệu trống
  - Sử dụng biểu thức chính quy để trích xuất số liệu về diện tích
- Đối với biến *Location*:
  - Tìm kiếm thông tin về phường có trong từng mẫu dữ liệu
  - Đối với thông tin nào không tìm thấy thì chuyển thành dữ liệu trống
- Đối với biến *Price*:
  - Loại bỏ các mẫu dữ liệu thể hiện thông tin "Tỷ", "tỷ", "Thỏa thuận", "Thương lượng"
  - Loại bỏ các cụm từ dư thừa "/tháng"
  - Sử dụng biểu thức chính quy để trích xuất số liệu về giá tiền, sau đó biến đổi về đúng định dạng

3.3 Tiền xử lý

3.3.1 Xử lý dữ liệu trống

3.3.1.1 Đặc trưng Area

Xử lý dữ liệu trống với phương pháp thay thế bằng giá trị trung vị. Đầu tiên, xác định giá trị trung vị của tập dữ liệu huấn luyện. Sau đó, thay thế những giá trị trống trên tập huấn luyện và kiểm thử bằng giá trị trung bình này.

3.3.1.2 Đặc trưng Location

Xử lý dữ liệu trống với phương pháp thay thế bằng giá trị cố định "Unknown_W". Sau khi xử lý, phần lớn dữ liệu trong đặc trưng *Location* là giá trị Unknown_W

3.3.1.3 Biến mục tiêu Price

Xử lý dữ liệu trống với phương pháp thay thế bằng giá trị trung vị. Đầu tiên, xác định giá trị trung vị của tập dữ liệu huấn luyện. Sau đó, thay thế những giá trị trống trên tập huấn luyện và kiểm thử bằng giá trị trung bình này.

3.3.2 Xử lý ngoại lệ

3.3.2.1 Đặc trưng Area

Thay thế giá trị ngoại lệ bằng phương pháp sử dụng IQR. Đầu tiên xác định ngưỡng trên, dưới của tập dữ liệu thông qua phân vị thứ nhất, thứ ba và giá trị IQR. Sau đó thay thế dữ liệu lớn

hơn ngưỡng trên bằng giá trị ngưỡng trên, thay thế dữ liệu nhỏ hơn ngưỡng dưới bằng giá trị ngưỡng dưới. Sau khi xử lý, phạm vi phân bố của đặc trưng *Area* giảm đáng kể.

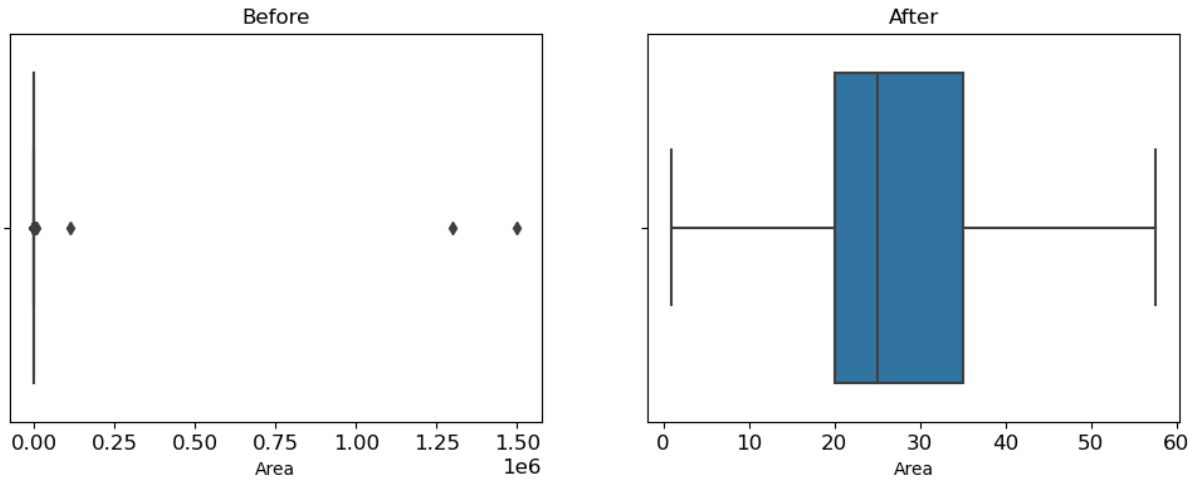


Figure 10: Sự phân bố của dữ liệu theo đặc trưng Area trước và sau khi xử lý trên BigDataset

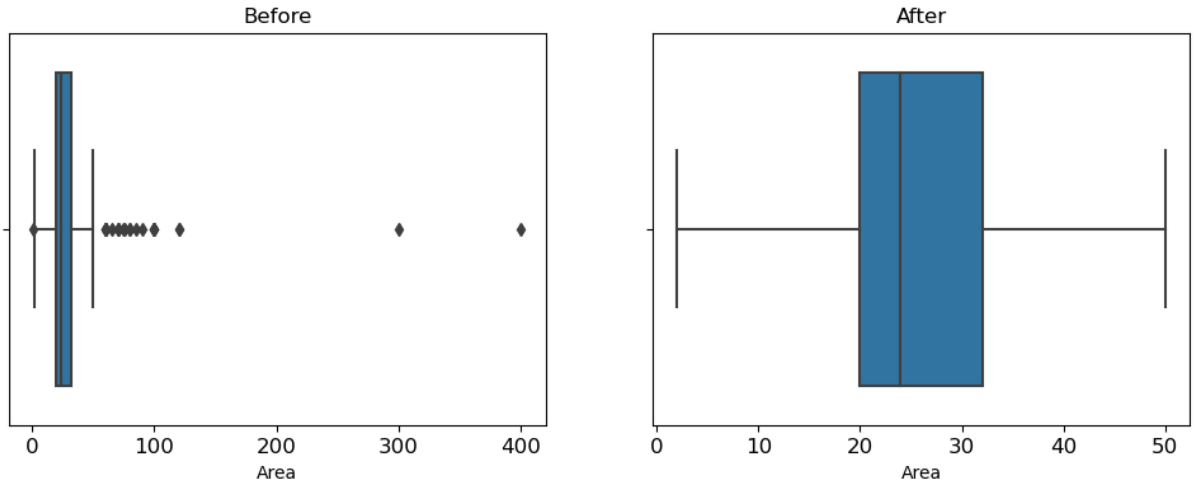


Figure 11: Sự phân bố của dữ liệu theo đặc trưng Area trước và sau khi xử lý trên SmallDataset

3.3.2.2 Biến mục tiêu Price

Thay thế giá trị ngoại lệ bằng phương pháp sử dụng IQR giống với đặc trưng Area. Sau khi xử lý, phạm vi phân bố của đặc trưng *Price* giảm đáng kể.

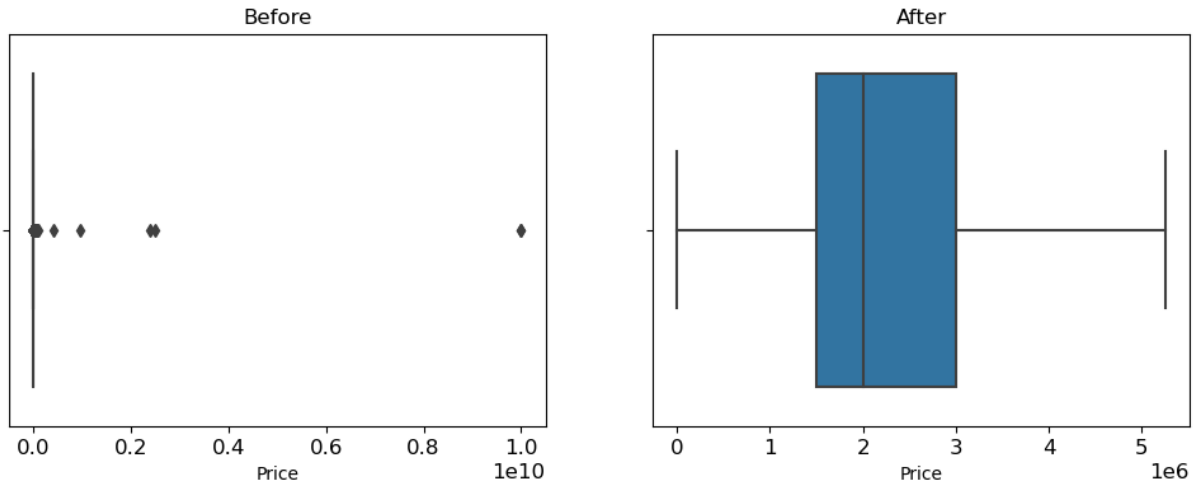


Figure 12: Sự phân bố của dữ liệu theo biến mục tiêu Price trước và sau khi xử lý trên BigDataset

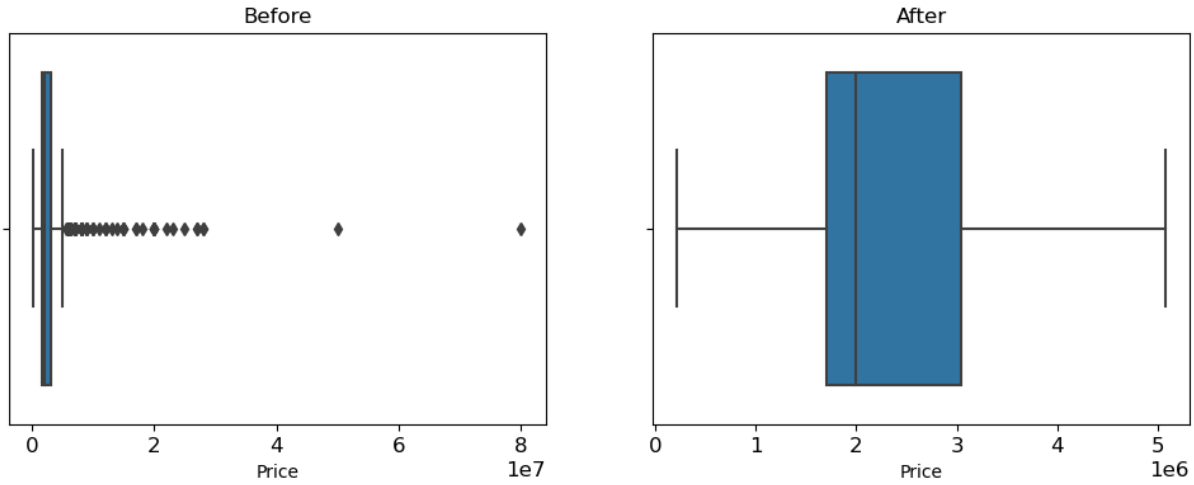


Figure 13: Sự phân bố của dữ liệu theo biến mục tiêu Price trước và sau khi xử lý trên SmallDataset

3.3.3 Chuẩn hóa dữ liệu

Sử dụng kĩ thuật chuẩn hóa MinMax đối với đặc trưng Area. Đầu tiên xác định giá trị area_min và area_max trên tập dữ liệu huấn luyện. Sau đó biến đổi dữ liệu trên tập huấn luyện/kiểm thử theo công thức $x = \frac{x - \text{area\_min}}{\text{area\_max} - \text{area\_min}}$.

Sau khi chuẩn hóa, dữ liệu được thu hẹp phạm vi phân bố nhưng vẫn giữ nguyên phân phối xác suất ban đầu.

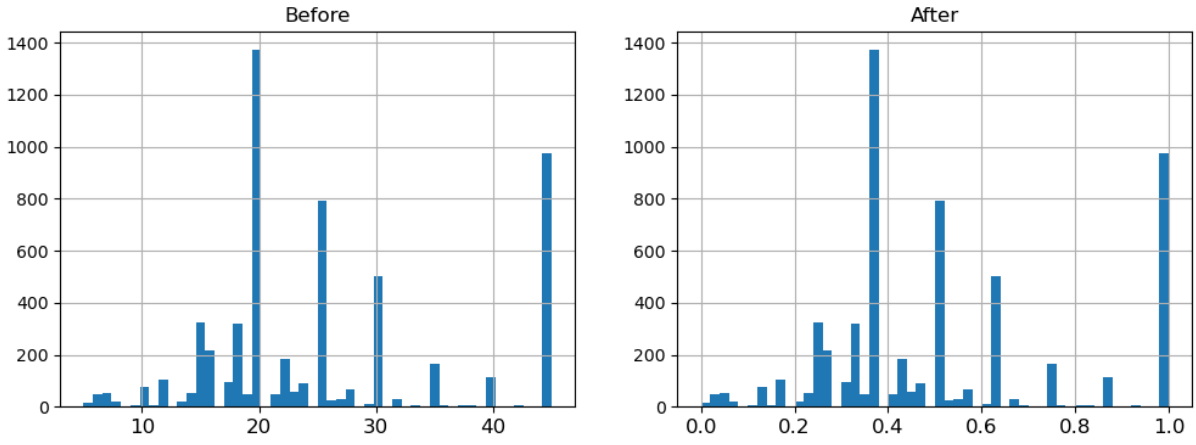


Figure 14: Sự phân bố của dữ liệu theo đặc trưng Area trước và sau khi xử lý trên BigDataset

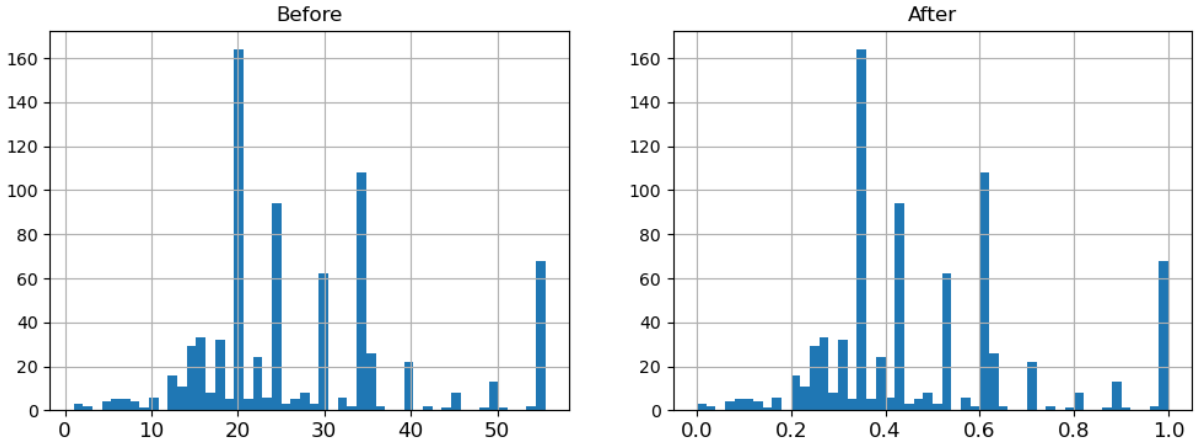


Figure 15: Sự phân bố của dữ liệu theo đặc trưng Area trước và sau khi xử lý trên SmallDataset

3.3.4 Mã hóa one-hot

Đây là cách truyền thống để đưa dữ liệu hạng mục về dạng số. Trong cách mã hóa này, một "từ điển" cần được xây dựng chứa tất cả các giá trị khả dĩ của từng dữ liệu hạng mục. Sau đó mỗi giá trị hạng mục sẽ được mã hóa bằng một vector nhị phân với toàn bộ các phần tử bằng 0 trừ một phần tử bằng 1 tương ứng với vị trí của hạng mục đó trong từ điển.

Sử dụng kĩ thuật mã hóa one-hot đối với đặc trưng có dữ liệu hạng mục là Ward.

4. Mô hình hóa dữ liệu

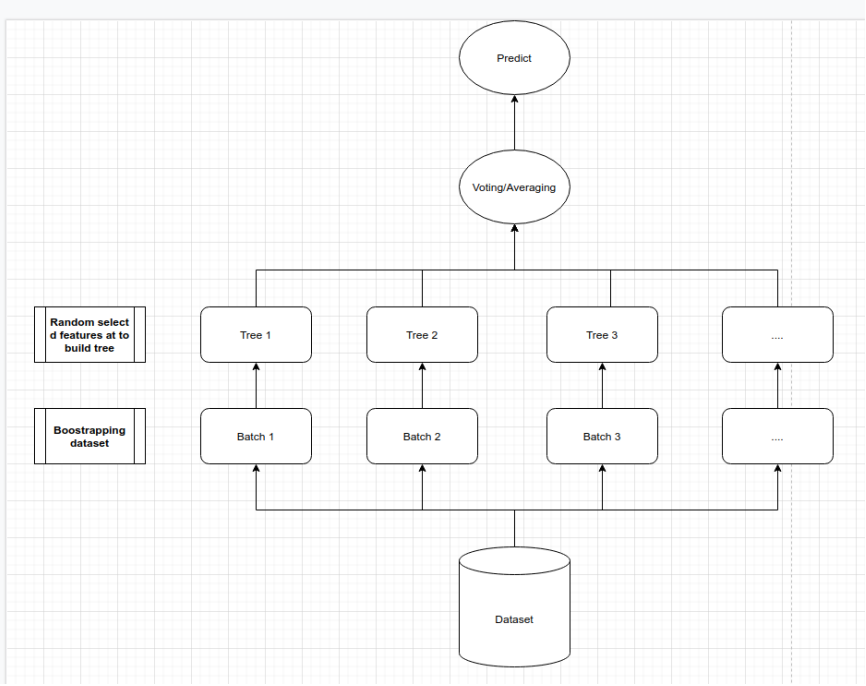
4.1 Phát biểu bài toán

- Đầu vào: Thông tin về diện tích, phường, quận của phòng trọ
- Đầu ra: Giá tiền thuê trọ trong 1 tháng

4.2 Lựa chọn mô hình

4.2.1 Mô hình hồi quy Random Forest

Mô hình Random Foresst được huấn luyện dựa trên sự phối hợp giữa luật kết hợp (ensembling) và quá trình lấy mẫu tái lập (bootstrapping). Thuật toán này tạo ra nhiều cây quyết định mà mỗi cây quyết định được huấn luyện dựa trên nhiều mẫu con khác nhau và kết quả dự báo là bầu cử (voting) từ toàn bộ những cây quyết định. Kết quả cuối cùng được tổng hợp từ nhiều mô hình.



Mô tả thuật toán:

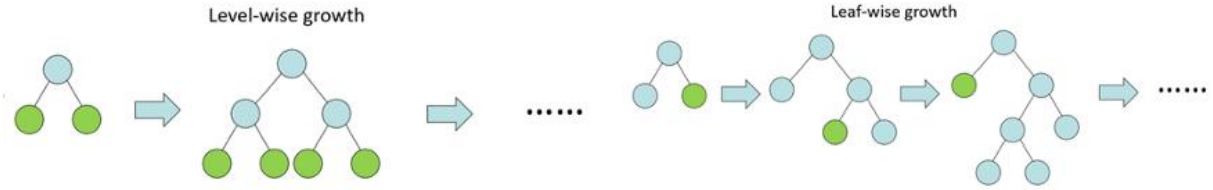
- Lấy mẫu tái lập một cách ngẫu nhiên từ tập huấn luyện để tạo thành một tập dữ liệu con
- Lựa chọn ra ngẫu nhiên **d** biến và xây dựng mô hình cây quyết định dựa trên những biến này và tập dữ liệu con ở bước 1. Vì mô hình có nhiều cây quyết định nên bước 1 và 2 sẽ được lặp lại nhiều lần
- Thực hiện bầu cử hoặc lấy trung bình giữa các cây quyết định để đưa ra dự báo

Bộ tham số chính của mô hình

- n_estimators: số lượng cây quyết định có trong mô hình
- max_depth: độ sâu tối đa của cây
- min_samples_leaf: số lượng mẫu tối thiểu cần ở một node lá của cây
- max_samples: số lượng mẫu được sử dụng để huấn luyện mô hình

4.2.2 Mô hình LightGBM

Nguyên lý căn bản của thuật toán LightGBM được phát triển từ mô hình cây quyết định tăng cường độ dốc Gradient Tree Boosting (GTB). Nguyên lý chung của GTB là thu được câu trả lời cuối cùng bằng cách kết hợp nhiều cây quyết định và bằng cách cộng kết quả của tất cả cây quyết định. LightGBM sử dụng phương pháp phát triển cây theo lá thay vì tăng trưởng cây theo cấp (được sử dụng bởi hầu hết các kĩ thuật dựa trên cây quyết định khác). LightGBM sử dụng 3 kĩ thuật cơ bản là Gradient-based One Side Sampling nhằm giảm số mẫu, Exclusive Feature Bundle nhằm giảm số đặc trưng của mẫu và Histogram algorithm nhằm giảm số điểm lựa chọn.



4.2.2.1 Kĩ thuật Gradient-based One Side Sampling (GOSS)

Kĩ thuật GOSS giữ lại các đối tượng có giá trị gradient lớn trong khi lấy mẫu ngẫu nhiên trên các mẫu dữ liệu có giá trị gradient nhỏ. Để bù lại ảnh hưởng đối với phân phối dữ liệu, khi tính toán mức tăng thông tin, GOSS giới thiệu một hệ số nhân cho các trường hợp đối tượng có độ dốc nhỏ. Cụ thể GOSS trước hết sắp xếp các thể hiện dữ liệu theo giá trị tuyệt đối của gradient và chọn các a×100% đối tượng hàng đầu. Sau đó nó lấy mẫu ngẫu nhiện b×100% đối tượng trong phần còn lại của dữ liệu. Sau đó, GOSS khuếch đại dữ liệu được lấy mẫu với độ dốc nhỏ theo hằng số (1-a)/b khi tính toán mức tăng thông tin. Bằng cách đó, các đối tượng chưa được huấn luyện sẽ được tập trung nhiều hơn mà không làm thay đổi quá nhiều đặc điểm phân phối của dữ liệu gốc.

4.2.2.2 Kĩ thuật Exclusive Feature Bundling (EFB)

EFB được xây dựng dựa trên đặc điểm là các dữ liệu nhiều chiều thường phân bố rất thưa thớt theo các chiều. Sự thưa thớt của không gian đặc trưng cung cấp cho chúng ta khả năng thiết kế một cách tiếp cận gần như không mất dữ liệu để giảm số lượng đặc trưng. Cụ thể, trong một không gian đối tượng thưa thớt, nhiều đối tượng loại trừ lẫn nhau, nghĩa là chúng không bao giờ đồng thời nhận giá trị khác 0. Có thể kết hợp các đặc trưng bị loại trừ vào một cách an toàn vào một đặc trưng duy nhất (gọi là gói đặc trưng bị loại trừ). Bằng thuật toán quét đặc trưng được thiết kế cẩn thận, chúng ta có thể tạo biểu đồ đặc trưng giống nhau từ các gói đặc trưng cũng như từ các đặc trưng riêng lẻ. Theo cách này, độ phức tạp của việc xây dựng biểu đồ thay đổi từ O(*data* × *tổng đặc trưng*) thành O(*data* × *đặc trưng kết hợp*), trong *đó đặc trưng kết hợp* << *tổng đặc trưng*. Điều này giúp tăng tốc đáng kể quá trình đào tạo cây ra quyết định mà không ảnh hưởng đến độ chính xác.

4.3 Metrics đánh giá

4.3.1 Root Mean Square Error

- Lỗi trung bình bình phương (RMSE) là độ lệch chuẩn của khoảng cách giữa giá trị dự đoán và giá trị thực sự (lỗi dự đoán). RMSE càng nhỏ tức là sai số càng bé thì mức độ tin cậy của mô hình càng cao

- Công thức tính RMSE

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Trong đó:

$\hat{y}_i$: là giá trị ước lượng

y_i: là giá trị thực sự

n: số mẫu dữ liệu

4.3.2 R-squared

- R-squared là thước đo sử dụng cho biết mức độ phù hợp của mô hình với ý nghĩa các đặc trưng.
- Công thức tính R-squared

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2}$$

Trong đó:

$\hat{y}_i$: là giá trị ước lượng

y_i: là giá trị thực sự

$\bar{y}_i$: giá trị trung bình của tập giá trị thực sự

n : số mẫu dữ liệu

4.3.3 Mean Absolute Error

- Mean Absolute Error là một độ đo sử dụng để đánh giá sự sai khác giữa mô hình dự đoán và tập dữ liệu trong bài toán hồi quy. Chỉ số này càng nhỏ thì mô hình càng chính xác.
- Công thức tính Mean Absolute Error

$$MAE = \sum_{i=1}^{n} \frac{\hat{y}_i - y_i}{n}$$

Trong đó:

$\hat{y}_i$: là giá trị ước lượng

y_i: là giá trị thực sự

n: số mẫu dữ liệu

4.3.4 Mean Absolute Percentage Error

- Mean Absolute Percentage Error là phần trăm sai số trung bình tuyệt đối.
- Công thức tính Mean Absolute Percentage Error

$$MAPE = \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

Trong đó:

$\hat{y}_i$: là giá trị ước lượng

y_i: là giá trị thực sự

n: số mẫu dữ liệu

4.4 Kết quả huấn luyện

4.4.1 Mô hình hồi quy LightGBM

- LightGBM có các tham số ảnh hưởng đến kết quả như:
  - max_depth: chiều sâu tối đa của cây quyết định
  - num_leaves: số lá tối đa của cây quyết định
  - n_estimators: số cây quyết định có trong mô hình
  - min_child_samples: số lượng mẫu dữ liệu tối thiểu cần ở một nút lá
  - min_split_gain: lượng giá trị mất mát tối thiểu phải giảm để tạo thêm một phân vùng mới trên nút là của cây
- Với việc sử dụng GridSearchCV để tìm ra bộ tham số tối ưu cho mô hình với
  - Big_dataset_best_params: {max_depth: 15, num_leaves: 21, n_estimators: 500, min_child_samples: 8, min_split_gain: 0}
  - Small_ dataset_best_params: {max_depth: 7, num_leaves: 21, n_estimators: 300, min_child_samples: 6, min_split_gain: 0}

- Kết quả thu được như sau:

			RMSE	R-squared	MAPE	MAE
Default hyperparameters	Big Dataset	tập huấn luyện	1116307.68	0.35	2.53	827697.55
		tập kiểm thử	1162229.89	0.31	1.28	867972.47
	Small Dataset	tập huấn luyện	915846.00	0.44	0.37	700777.26
		tập kiểm thử	1029754.00	0.37	0.43	793112.40
Best hyperparameters	Big Dataset	tập huấn luyện	1119162.97	0.34	2.51	834667.57
		tập kiểm thử	1158052.97	0.32	1.29	870600.93
	Small Dataset	tập huấn luyện	885289.97	0.48	0.36	676874.90
		tập kiểm thử	1016099.60	0.39	0.41	780322.54

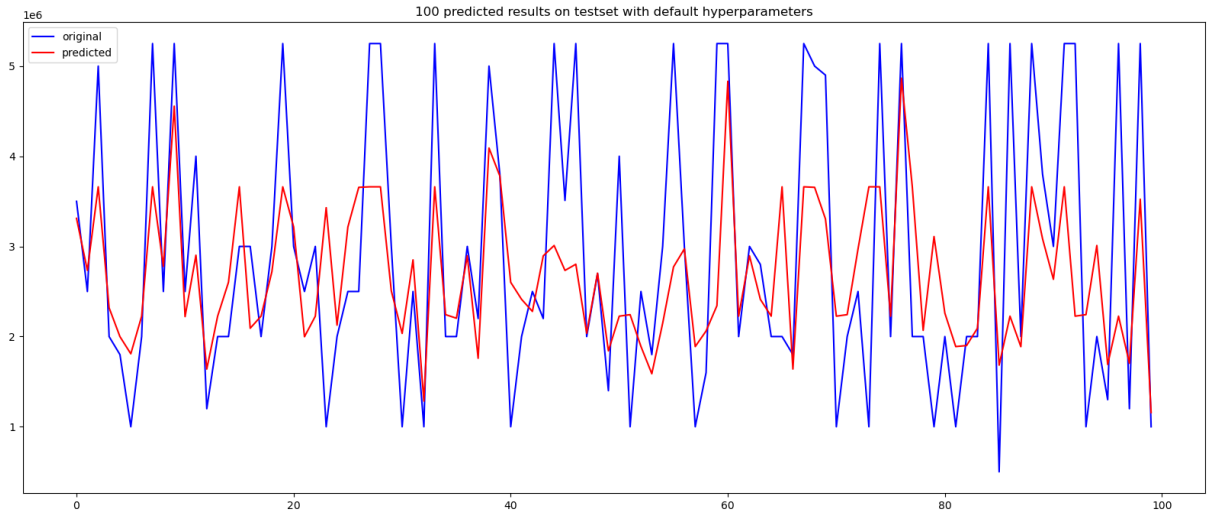


Figure 16: Kết quả dự đoán 100 mẫu dữ liệu đầu tiên trên tập kiểm thử của BigDataset với bộ siêu tham số mặc định

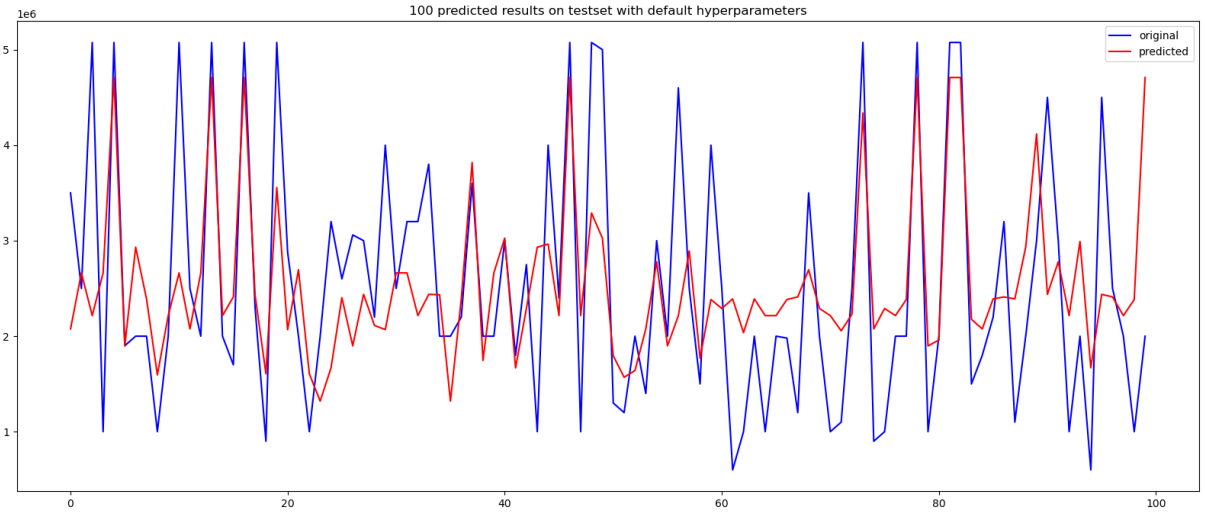


Figure 17: Kết quả dự đoán 100 mẫu dữ liệu đầu tiên trên tập kiểm thử của SmallDataset với bộ siêu tham số mặc định

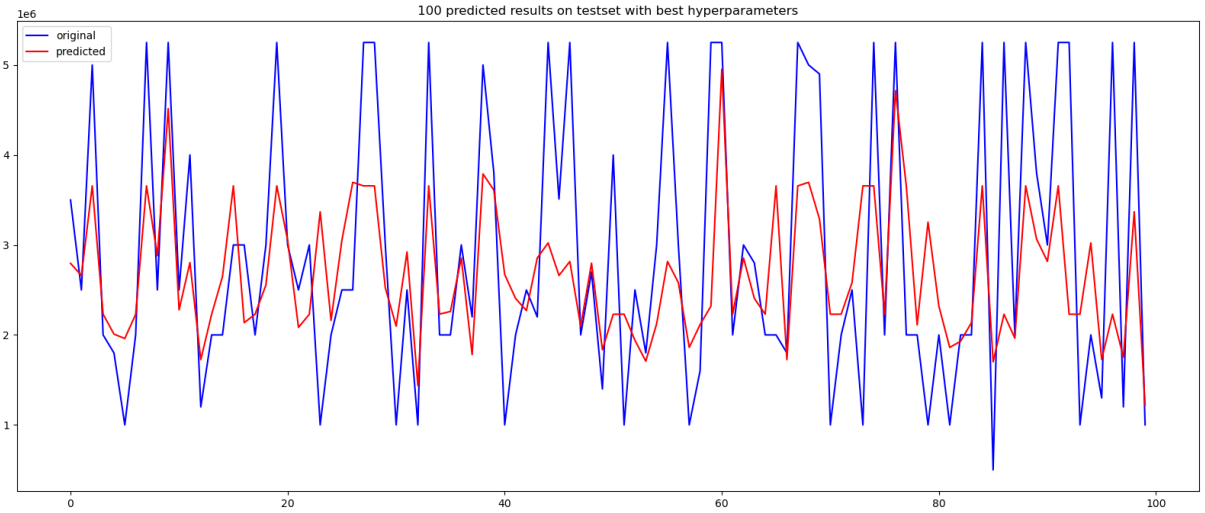


Figure 18: Kết quả dự đoán 100 mẫu dữ liệu đầu tiên trên tập kiểm thử của BigDataset với bộ siêu tham số tối ưu

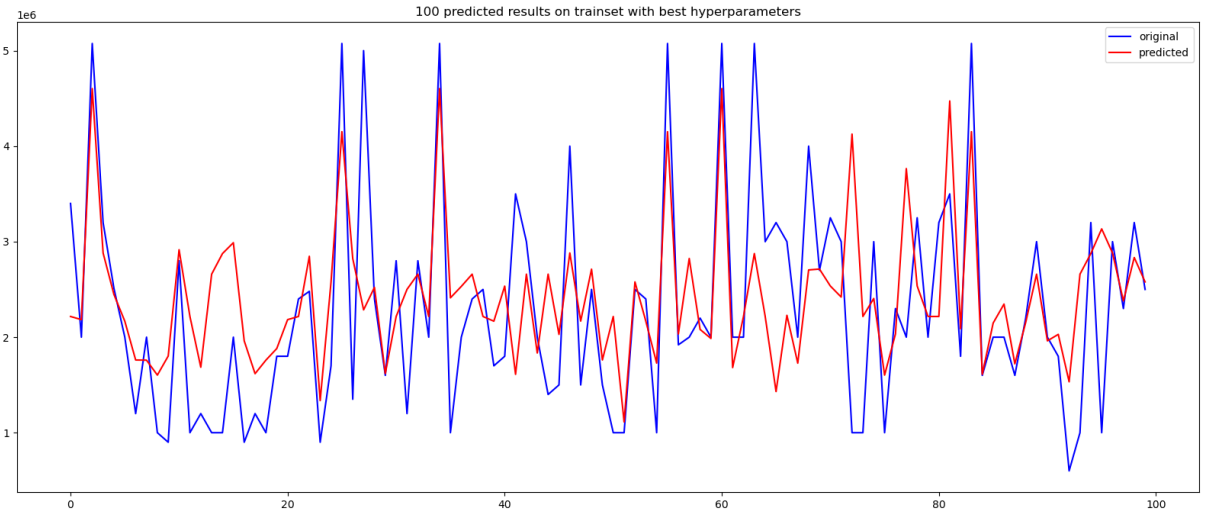


Figure 19: Kết quả dự đoán 100 mẫu dữ liệu trên tập kiểm thử của SmallDataset với bộ siêu tham số tối ưu

4.4.2 Mô hình Random Forest

- Random Forest có các tham số ảnh hưởng đến kết quả như:
  - max_depth: chiều sâu tối đa của cây quyết định
  - min_samples_leaf: số mẫu dữ liệu tối thiểu cần ở nút một lá của cây
  - min_samples_split: số mẫu dữ liệu tối thiểu cần để tách một nút nội bộ của cây
  - n_estimators: số cây quyết định có trong mô hình
- Với việc sử dụng GridSearchCV để tìm ra bộ tham số tối ưu cho mô hình với
  - Big_dataset_best_params: {max_depth: 10, min_samples_leaf: 3, min_samples_split: 4, n_estimators: 300}
  - Smalldataset_best_params: {max_depth: 7, min_samples_leaf: 2, min_samples_split: 4, n_estimators: 300}
- Kết quả thu được như sau:

			RMSE	R-squared	MAPE	MAE
Default hyperparameters	Big Dataset	tập huấn luyện	1085793.53	0.38	2.01	791101.72
		tập kiểm thử	1176814.46	0.30	1.12	874966.77
	Small Dataset	tập huấn luyện	761660.61	0.62	0.28	544189.81
		tập kiểm thử	1068065.33	0.32	0.40	801880.23
Best hyperparameters	Big Dataset	tập huấn luyện	1116243.31	0.35	2.57	829534.90
		tập kiểm thử	1163531.16	0.31	1.26	874082.89
	Small Dataset	tập huấn luyện	870579.71	0.50	0.35	662195.57
		tập kiểm thử	1013089.82	0.39	0.40	772311.76

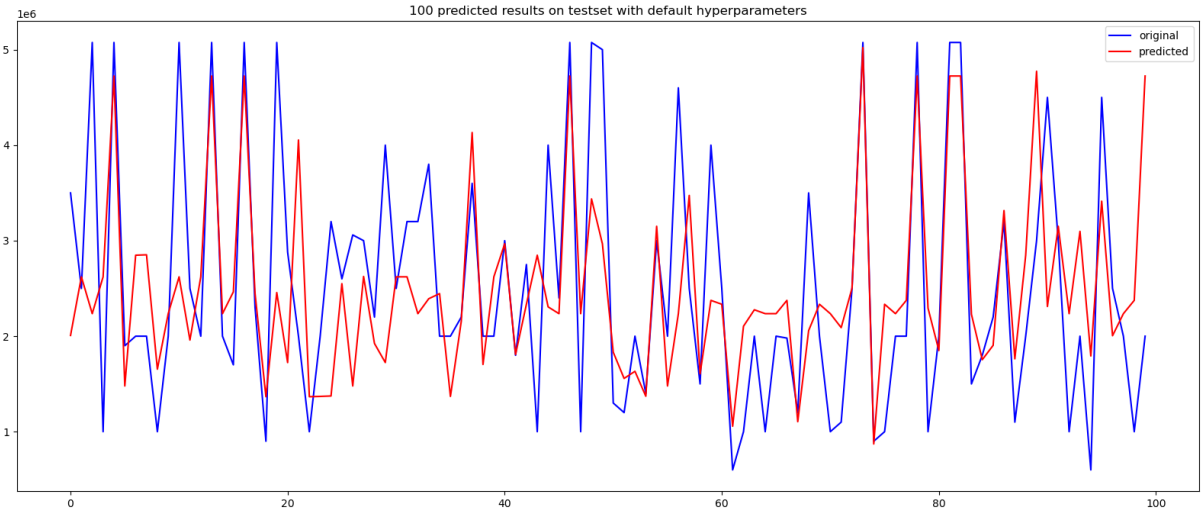


Figure 20: Kết quả dự đoán 100 mẫu dữ liệu trên tập kiểm thử của SmallDataset với bộ siêu tham số mặc định

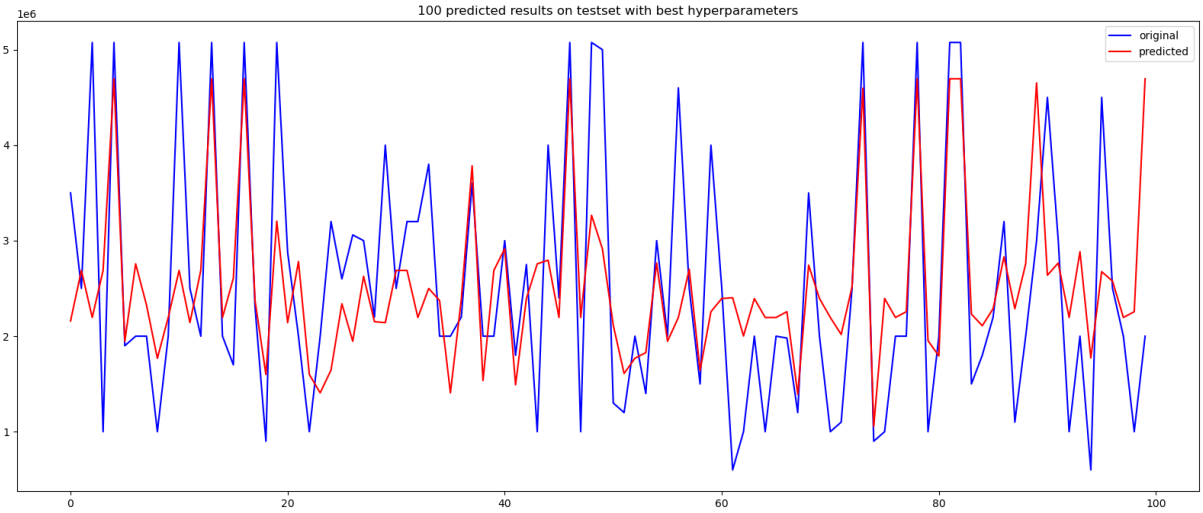


Figure 21: Kết quả dự đoán 100 mẫu dữ liệu đầu tiên trên tập kiểm thử của SmallDataset với bộ siêu tham số tối ưu

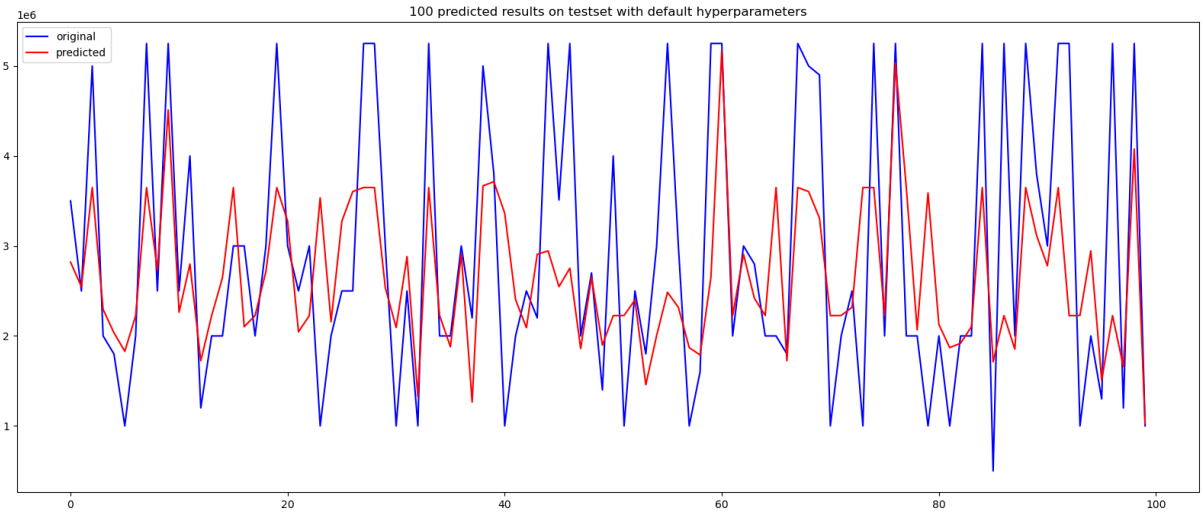


Figure 22: Kết quả dự đoán 100 mẫu dữ liệu đầu tiên trên tập kiểm thử của BigDataset với bộ siêu tham số mặc định

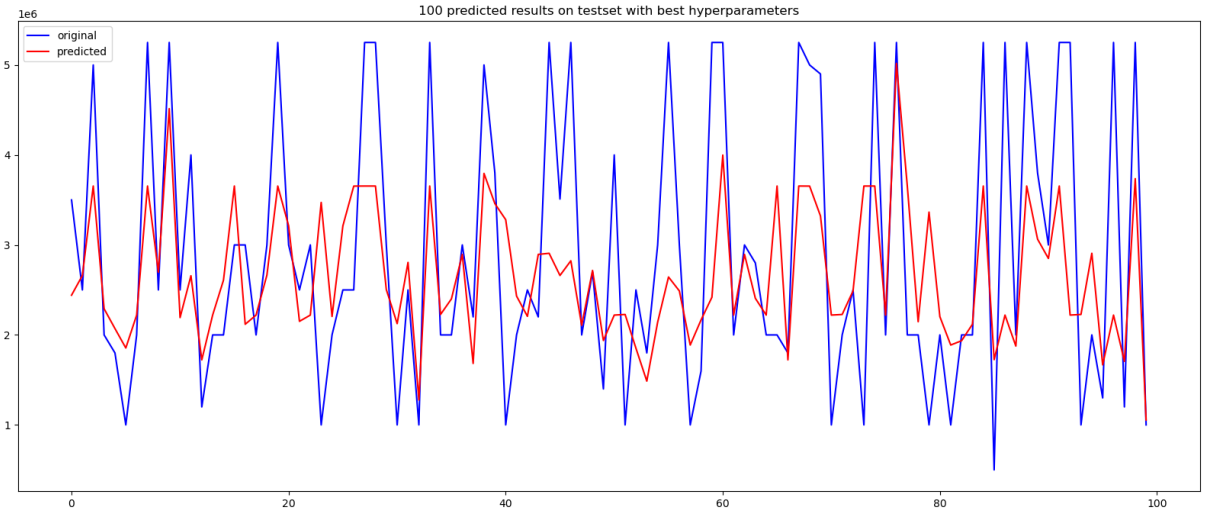


Figure 23: Kết quả dự đoán 100 mẫu dữ liệu đầu tiên trên tập kiểm thử của BigDataset với bộ siêu tham số tối ưu

4.4.3 Nhận xét kết quả

- Nhìn chung, kết quả dự đoán chưa đạt được độ tin cậy cao
- Kết quả dự đoán của mô hình Random Forest nhỉnh hơn kết quả của mô hình LightGBM
- Tốc độ huấn luyện trên SmallDataset nhanh hơn BigDataset gần 4 lần

5. Kết luận và Hướng phát triển

5.1 Kết luận

5.1.1 Kết quả đạt được

Đề tài này đã trình bày các kết quả nghiên cứu trong vấn đề dự đoán giá phòng trọ trong địa bàn thành phố Đà Nẵng, Việt Nam dựa trên các mô hình học máy.

Kết quả đạt được là:

- Đã xây dựng hệ thống dự đoán giá trọ trong địa bàn thành phố Đà Nẵng, Việt Nam
- Thử nghiệm 2 mô hình học máy để ứng dụng trong vấn đề dự đoán

Dựa trên kết quả thử nghiệm đã so sánh các mô hình học máy trong việc dự đoán giá phòng trọ.

5.1.2 Giới hạn của đề tài

Về cơ bản, đề tài đã hoàn thành yêu cầu đặt ra là dự đoán giá trọ. Tuy nhiên kết quả được đánh giá chưa cao, bởi vì tập dữ liệu còn chứa phần lớn mẫu dữ liệu rác hoặc không đủ thông tin.

Một điểm hạn chế khác của đề tài là còn nhiều đặc trưng chưa được tổng hợp từ trang web như đặc trưng về nội thất, điều kiện cơ sở vật chất, …

5.2 Hướng phát triển

Trong giới hạn nghiên cứu của đồ án nhóm xin đề xuất hướng nghiên cứu trong tương lai của đề tài này là.

- Cải thiện độ chính xác để tăng độ tin cậy của dự án
- Thử nghiệm với các phương pháp học máy khác
- Mở rộng và tinh lọc bộ dữ liệu
- Thử nghiệm với những đặc trưng khác

6. Tài liệu tham khảo

Welcome to LightGBM’s documentation! — LightGBM 3.3.2 documentation

Random Forest Regression in Python - GeeksforGeeks

https://www.mathworks.com/campaigns/offers/next/machine-learning-vs-deep-learning.html

Selenium with Python — Selenium Python Bindings 2 documentation (selenium-python.readthedocs.io)

Beautiful Soup Documentation — Beautiful Soup 4.12.0 documentation (crummy.com)

11. Giới thiệu về feature engineering — Deep AI KhanhBlog (phamdinhkhanh.github.io)

Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation (utep.edu)

Data Cleaning: Definition, Benefits, And How-To | Tableau