TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

DỰ ĐOÁN GIÁ BẤT ĐỘNG SẢN Ở ĐÀ NẴNG

HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN	ĐIỂM BẢO VỆ
Nguyễn Văn Hiếu	19N13	
Nguyễn Thái Lộc	19N13	
Võ Văn Thành	19N13	

ĐÀ NẴNG, 06/2022

TÓM TẮT

Đà Nẵng là một trong những thị trường bất động sản trọng điểm của khu vực miền Trung nói riêng và Việt Nam nói chung. Thị trường bất động sản Đà Nẵng phát triển từ khá sớm, song hành cùng những biến động của nền kinh tế, góp phần quan trọng trong định vị và định hướng thị trường khu vực. Vấn đề đặt ra là làm thế nào để dự đoán giá của một bất động sản nào đó ở Đà Nẵng. Từ vấn đề đó, nhóm em mong muốn xây dựng một mô hình học máy để dự đoán giá bất động sản ở thành phố này. Kết quả thu được một mô hình học máy có giá trị chênh lệch trung bình của giá dự đoán từ mô hình và giá trị thực tế (RMSE) khoảng 15 tỉ VNĐ.

BẢNG PHÂN CÔNG NHIỆM VỤ

Bảng 1. Bảng phân chia công việc.

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá
Nguyễn Văn Hiếu	Crawl dữ liệu.	Đã hoàn thành
	EDA.	Đã hoàn thành
	Viết tiểu luận	Đã hoàn thành
Nguyễn Thái Lộc	Trích xuất đặc trưng.	Đã hoàn thành
	Viết tiểu luận	Đã hoàn thành
Võ Văn Thành	Crawl dữ liệu.	Đã hoàn thành
	Mô hình hóa dữ liệu.	Đã hoàn thành
	Viết tiểu luận	Đã hoàn thành

MỤC LỤC

1. Giới thiệu 7

2. Thu thập và mô tả dữ liệu 7

2.1. Thu thập dữ liệu 7

2.2. Mô tả dữ liệu 9

2.2.1. Tổng quan 9

2.2.2. Thống kê mô tả trực quan 10

3. Trích xuất đặc trưng 14

3.1. Xử lý dữ liệu trống 14

3.2. Label Encoding 15

3.3. Lựa chọn đặc trưng 16

3.4. Outliers cho tập train 17

3.5. Chuẩn hoá min-max-scaler 18

4. Mô hình hóa dữ liệu 19

4.1. Các mô hình 19

4.1.1. Linear Regression 20

4.1.2. SVR 20

4.1.3. Random Forest 21

4.1.4. Ada Boost 22

4.1.5. Gradiant Boost 23

4.2. Chia dữ liệu 24

4.3. Tham số của quá trình huấn luyện mô hình 26

4.4. Hiệu suất của các mô hình trên tập Huấn luyện, Xác thực và Kiểm thử dựa trên độ đo RMSE 26

4.5. Áp dụng mô hình vào thực tế 27

5. Kết luận 28

5.1. Kết quả đạt được 28

5.2. Hướng phát triển 28

6. Tài liệu tham khảo 28

DANH MỤC HÌNH ẢNH

Hình 1. Quy trình xử lí dữ liệu. 7
Hình 2. Các tham số đầu vào của chương trình crawl dữ liệu. 8
Hình 3. Danh sách các tin đăng bất động sản trên batdongsan.com.vn. 8
Hình 4. Trang chi tiết của một tin đăng. 9
Hình 5. Các thông tin cần crawl của một tin đăng. 9
Hình 6. Kết quả crawl dữ liệu của một tin đăng. 9
Hình 7. Biểu đồ phân tán của các đặc trưng Price,Area. 10
Hình 8. Số lượng diện tích đất trung bình đăng bán. 11
Hình 9. Số lượng trung bình số lượng các quận. 11
Hình 10. Biểu diễn trung bình các khoảng giá. 12
Hình 11. Tỉ lệ phần trăm chiếm của các quận. 12
Hình 12. Mối tương qua giữa 2 đặc trưng district và Price. 13
Hình 13. Mối tương quan giữa 2 đặc trưng Price và Area. 13
Hình 14. Mối tương quan giữa các đặc trưng Price, Area, Price. 14
Hình 15. Số dữ liệu trống trong các đặc trưng. 14
Hình 16. Dữ liệu sau khi xử lý dữ liệu trống. 15
Hình 17. Thêm mới các cột dữ liệu. 15
Hình 18.Biểu đồ cột biểu diễn giá tiền trung bình trên một mét vuông theo quận. 15
Hình 19. Biểu đồ cột biểu diễn giá tiền trung bình trên một mét vuông theo loại bất động sản. 16
Hình 20. Kết quả sau khi label encoding. 16
Hình 21. Ma trận tương quan giữa các đặc trưng. 16
Hình 22. Biểu đồ phân bố lấy mẫu của 2 đặc trưng Price và Area. 17
Hình 23. Biểu đồ boxplot của đặc trưng Price và Area. 17
Hình 1524. Hàm xử lý ngoại lệ lệch trái. 17
Hình 25. Biểu đồ phân lấy mẫu sau khi xử lý ngoại lệ của hai đặc trưng Price và Area. 18
Hình 26. Biểu đồ boxplot sau khi xử lý ngoại lệ của hai đặc trưng Price and Area. 18
Hình 27. Hàm chuẩn hoá min-max-scaler. 18

Hình 28.Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 18
Hình 29. Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 trên tập train 19
Hình 30. Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 trên tập val. 19
Hình 31. Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 trên tập test. 19
Hình 32. Dữ liệu huấn luyện. 25
Hình 33. Dữ liệu xác thực. 25
Hình 34. Dữ liệu kiểm thử. 26
Hình 35. Mô hình huấn luyện. 26
Hình 36. Bảng thống kê RMSE của các tập dữ liệu huấn luyện. 26
Hình 37. Bảng thống kê RMSE của các tập dữ liệu xác thực. 27
Hình 38. Bảng thống kê RMSE của các tập dữ liệu kiểm thử. 27
Hình 39. Biểu đồ RMSE dữ liệu kiểm thử tập dữ liệu được chuẩn hóa của mô hình Gradient Boost. 27
Hình 40. Kết quả tìm kiếm bộ siêu tham số. 27
Hình 41. Bất động sản được chọn ngẫu nhiên trong tập dữ liệu kiểm thử. 28

DANH MỤC BẢNG BIỂU

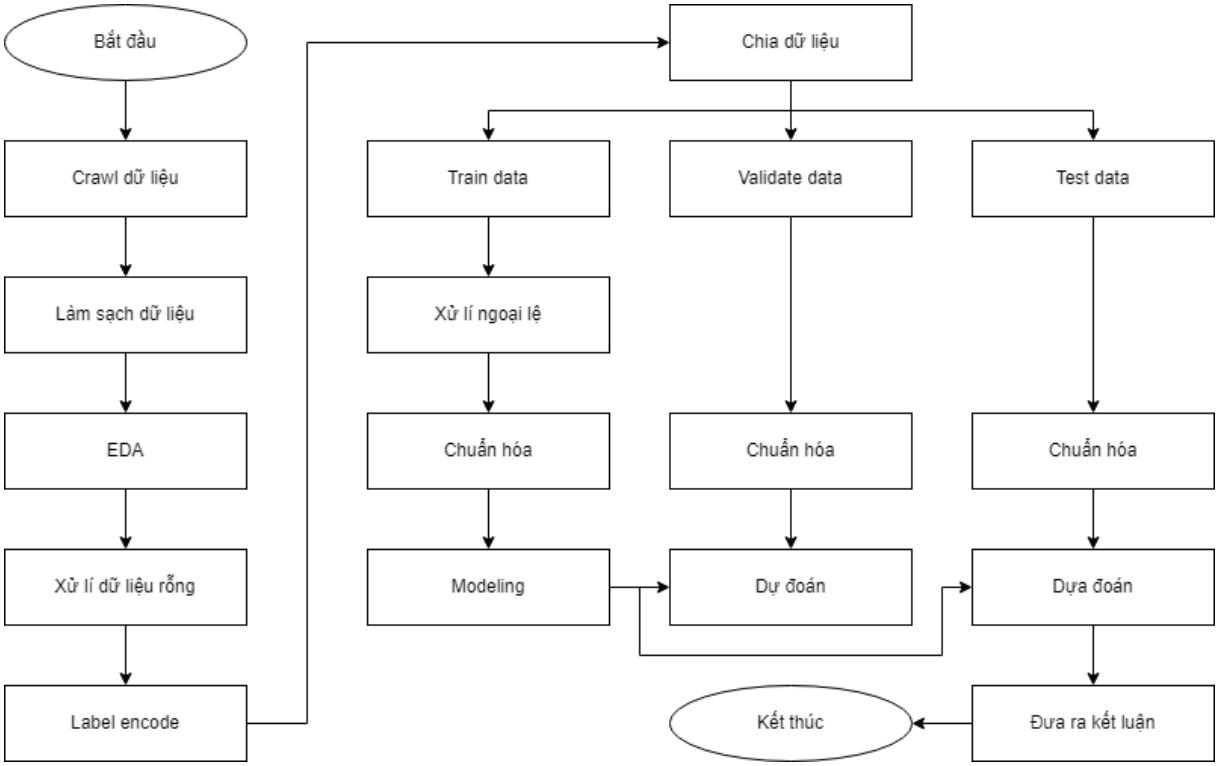
Bảng 1. Bảng phân chia công việc. 3
Bảng 2. Kiểu dữ liệu và số mẫu trống của các trường trong dữ liệu thu thập được.....10

1. Giới thiệu

Đà Nẵng là nơi có thị trường bất động sản hoạt động rất sôi nổi, song hành cùng những biến động của nền kinh tế, góp phần quan trọng trong định vị và định hướng thị trường khu vực. Việc đoán định giá bất động sản với những người có kinh nghiệm lẫn với những người mới gia nhập thị trường này là rất khó.

Dựa trên thực tế trên, nhóm em muốn thử xây dựng một mô hình học máy sử dụng các mô hình Linear Regression, SVR,… để đưa ra dự đoán về giá của một bất động sản ở thành phố này dựa trên những dữ liệu thu thập được.

Nhóm đã bàn bạc và thống nhất quy trình xử lí dữ liệu như sau:



Hình 1. Quy trình xử lí dữ liệu.

2. Thu thập và mô tả dữ liệu

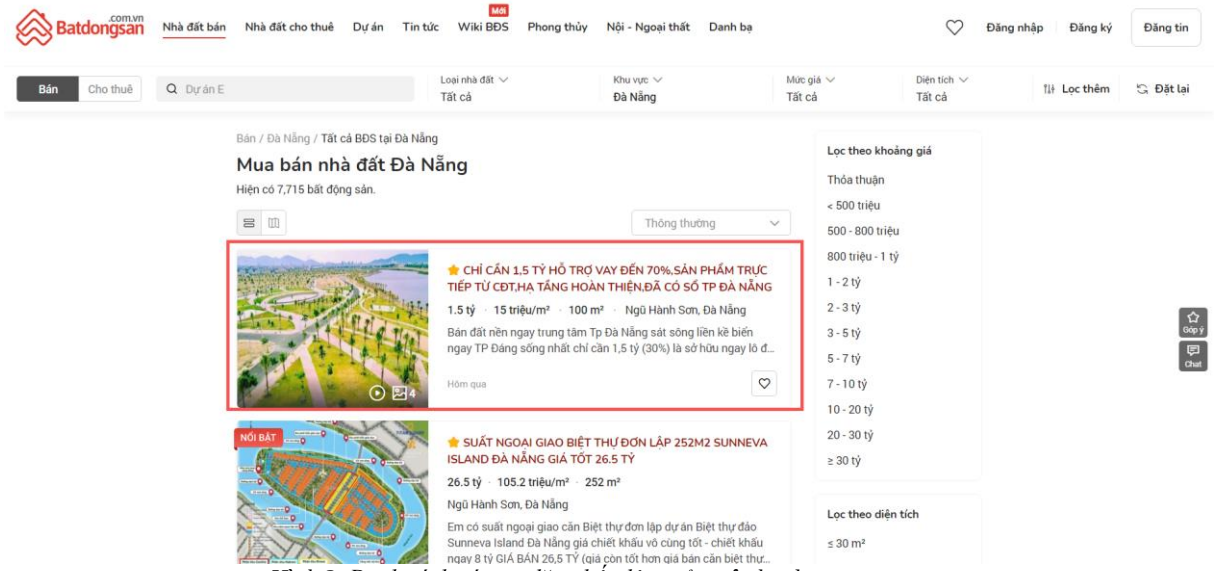
2.1. Thu thập dữ liệu

SV mô tả mô tả giải pháp thu thập dữ liệu gồm nguồn dữ liệu, công cụ thu thập, cách thức sử dụng công cụ, đầu vào và đầu ra của quá trình thu thập, và cho ví dụ minh hoạ.

- Nguồn dữ diệu: https://batdongsan.com.vn/nha-dat-ban-da-nang/
- Công cụ thu thập: Viết code crawl sử dụng thư viện Seleniumcủa Python.
- Cách thức sử dụng công cụ thu thập:
    - Thư viện Selenium [1]: cách sử dụng thư viện có thể xem trực tiếp ở mục tài liệu tham khảo [1].
    - Code crawl (file crawl_data.ipynb): điều chỉnh các đầu vào (nếu cần) rồi bấm “Run All” của IDE Visual Studio Code và chờ đợi kết quả của chương trình.

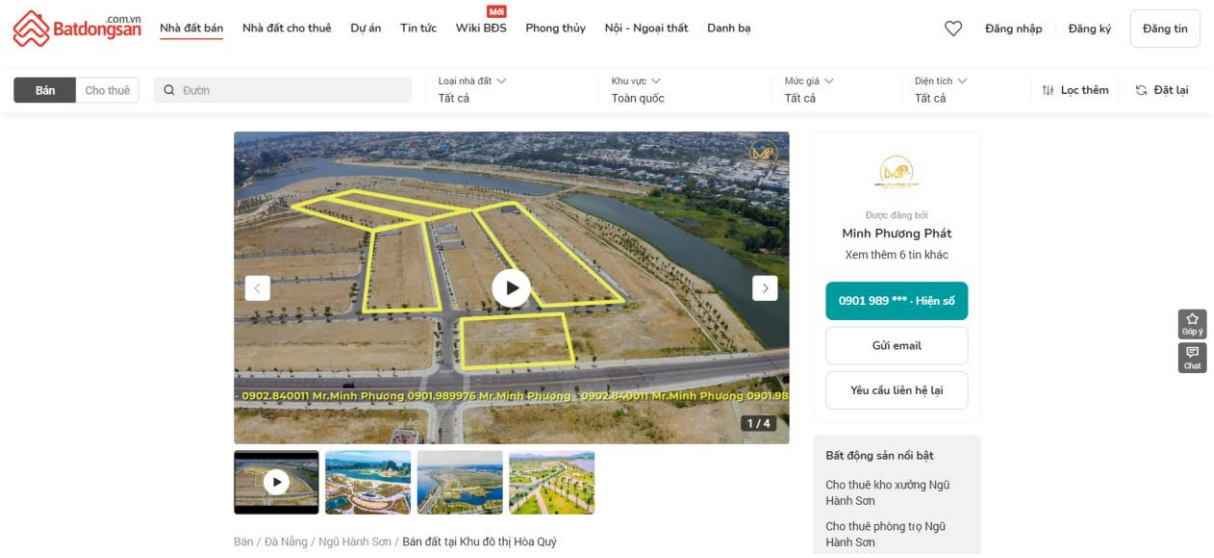
Hình 2. Các tham số đầu vào của chương trình crawl dữ liệu.

- Đầu vào của quá trình thu thập:
  - base_url: đường dẫn đến trang web muốn cào dữ liệu.
  - num_of_page: số trang dữ liệu muốn crawl từ base_url.
  - name_csv_raw: tên file csv lưu dữ liệu raw.
  - name_csv_cleaned: tên file csv lưu dữ liệu đã qua tiền xử lí ban đầu.
- Đầu ra của quá trình thu thập: 2 file csv với tên đã đặt ở 2 tham cuối của đầu vào.
- Ví dụ minh họa: Crawl data của bài viết đầu tiên của trang web https://batdongsan.com.vn/nha-dat-ban-da-nang/ (lúc viết tiểu luận này).



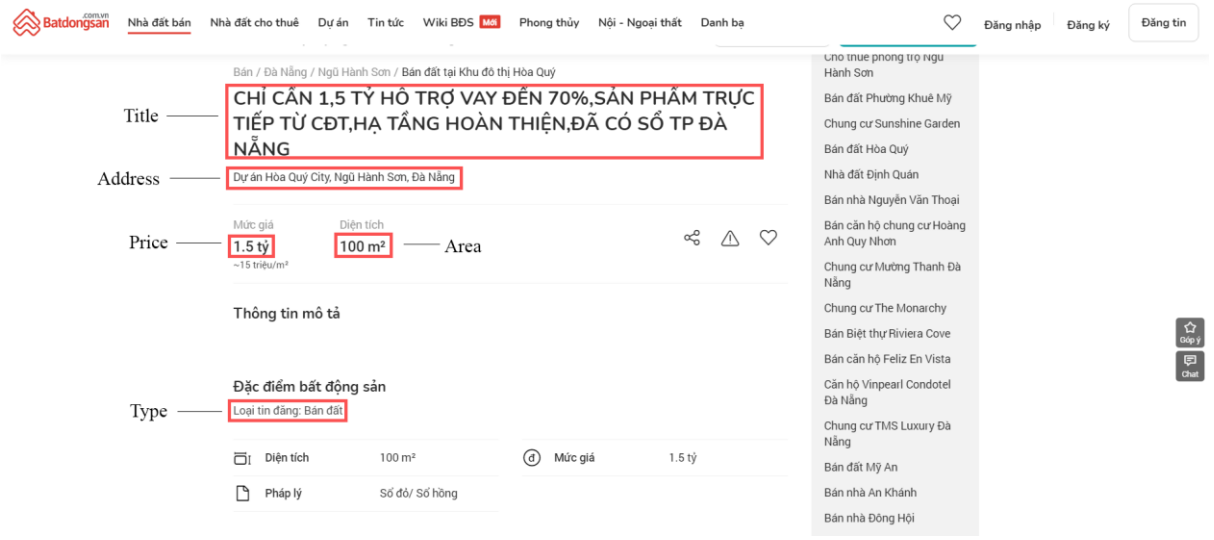
Hình 3. Danh sách các tin đăng bất động sản trên batdongsan.com.vn.

Mỗi tin đăng sẽ có một trang chi tiết:



Hình 4. Trang chi tiết của một tin đăng.

Ta cần lấy các thông tin sau của bài đăng đó:



Hình 5. Các thông tin cần crawl của một tin đăng.

Đây là kết quả thu được từ quá trình crawl:

```
Raw data: [' Chỉ cần 1,5 tỷ hỗ trợ vay đến 70%,Sản phẩm trực tiếp từ CĐT,Hạ tầng hoàn thiện,đã có sổ Tp Đà Nẵng', 'Dự án Hòa Quý City
, Ngũ Hành Sơn, Đà Nẵng', 'Loại tin đăng: Bán đất', '100 m²', '1.5 tỷ']
Cleaned data: [' Chỉ cần 1,5 tỷ hỗ trợ vay đến 70%,Sản phẩm trực tiếp từ CĐT,Hạ tầng hoàn thiện,đã có sổ Tp Đà Nẵng', ' Ngũ Hành Sơn'
, 'Bán đất', '100', '1500.0']
```

Hình 6. Kết quả crawl dữ liệu của một tin đăng.

Sau khi có được dữ liệu, ta chỉ cần lưu vào file csv nữa là đã hoàn thành.

2.2. Mô tả dữ liệu

2.2.1. Tổng quan

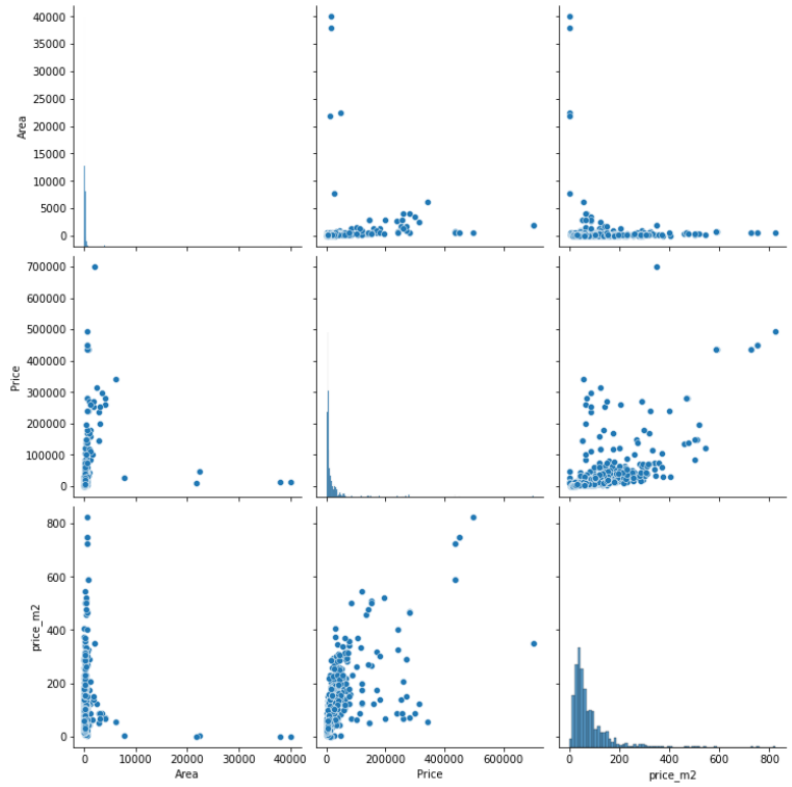
- Raw data có 2000 mẫu.
- Gồm 5 trường: tiêu đề, địa chỉ, loại bất động sản, diện tích và giá.

Bảng 2. Kiểu dữ liệu và số mẫu trống của các trường trong dữ liệu thu thập được.

Tên trường	Kiểu dữ liệu	Số mẫu dữ liệu bị trống
Tiêu đề	String	0
Địa chỉ	String	2
Loại bất động sản	String	0
Diện tích (m^2)	Float	2
Giá (triệu VNĐ)	Int	517

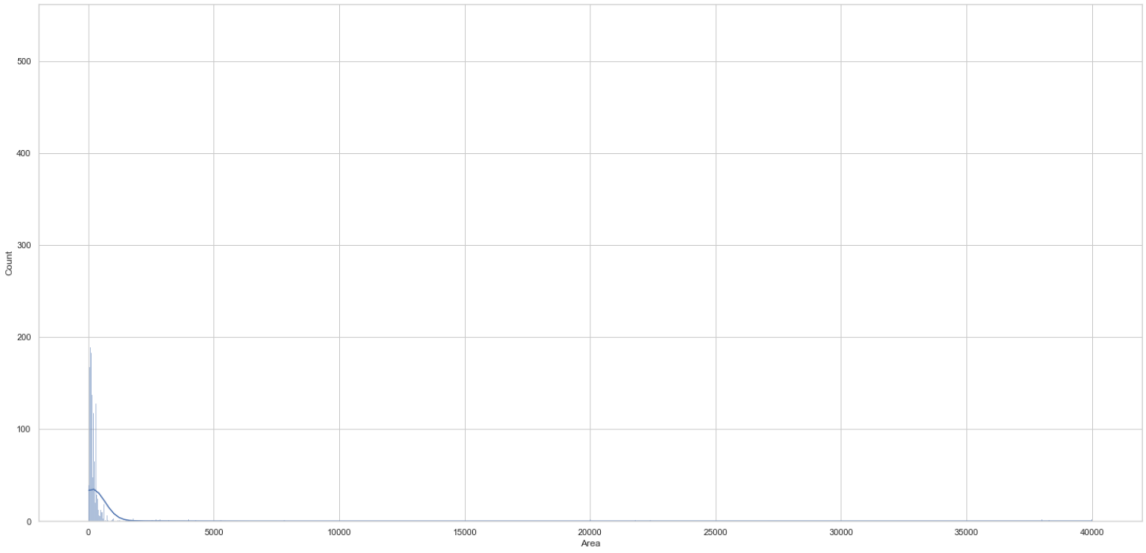
2.2.2. Thống kê mô tả trực quan

- Sử dụng pairplot để xem độ phân tán dữ liệu trong dataset.



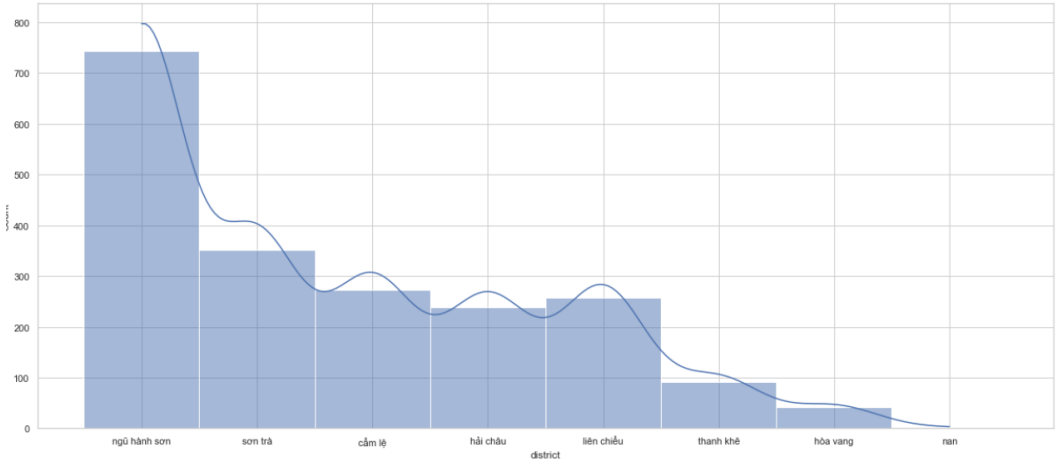
Hình 7. Biểu đồ phân tán của các đặc trưng Price,Area.

- Số lượng diện tích thông qua histplot.

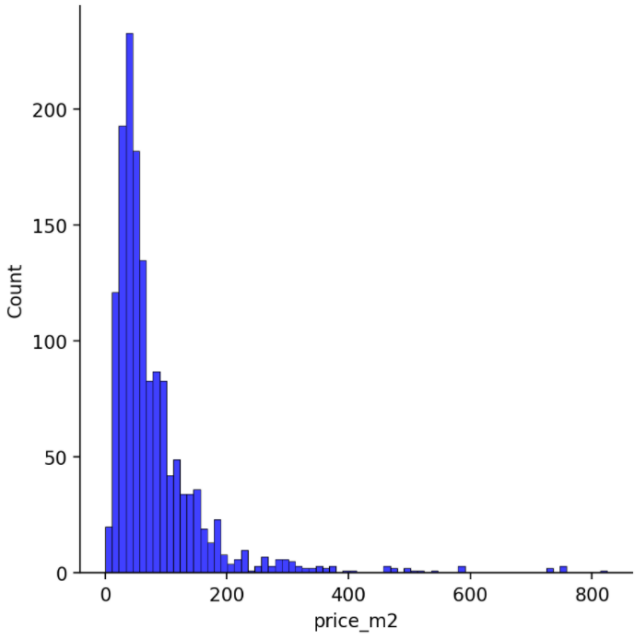


Hình 8. Số lượng diện tích đất trung bình đăng bán.

- Số lượng các quận thông qua histplot.



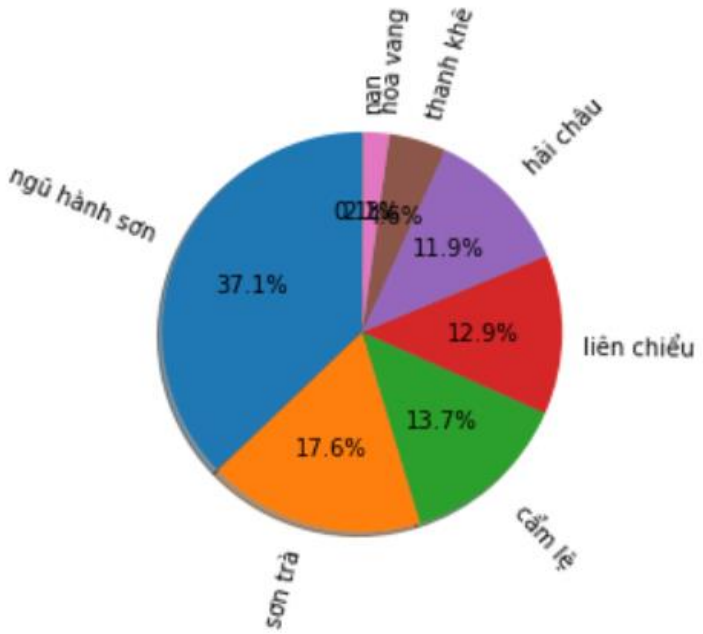
Hình 9. Số lượng trung bình số lượng các quận.



Hình 10. Biểu diễn trung bình các khoảng giá.

➢ Giá đất trung bình từ 0-200 triệu/ m2 chiếm nhiều nhất.

- Biểu diễn biểu đồ phần trăm các quận bằng plt.pie.

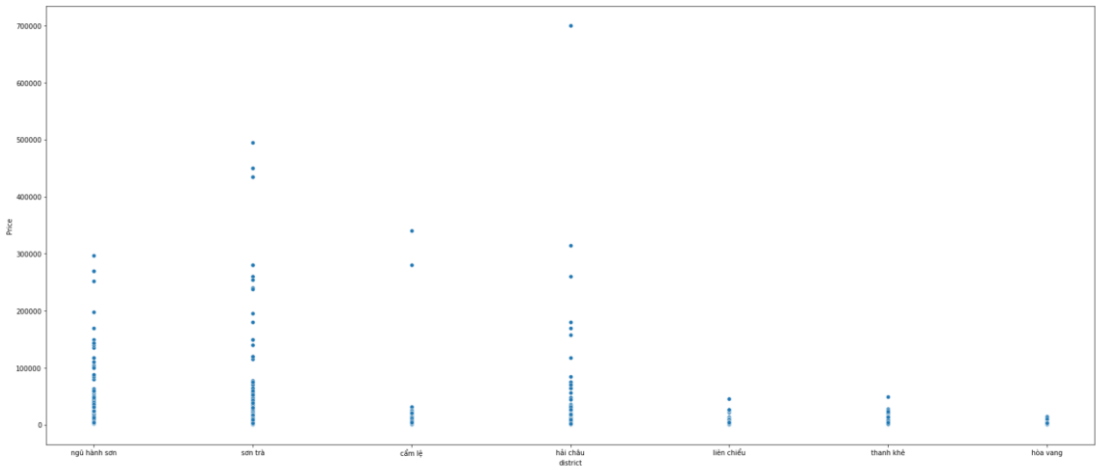


Hình 11. Tỉ lệ phần trăm chiếm của các quận.

➢ Quận ngủ hành sơn chiếm tỉ lệ cao nhất, Hòa Vang có tỉ lệ thấp nhất giữa 2 quận chênh lệnh 35.4%

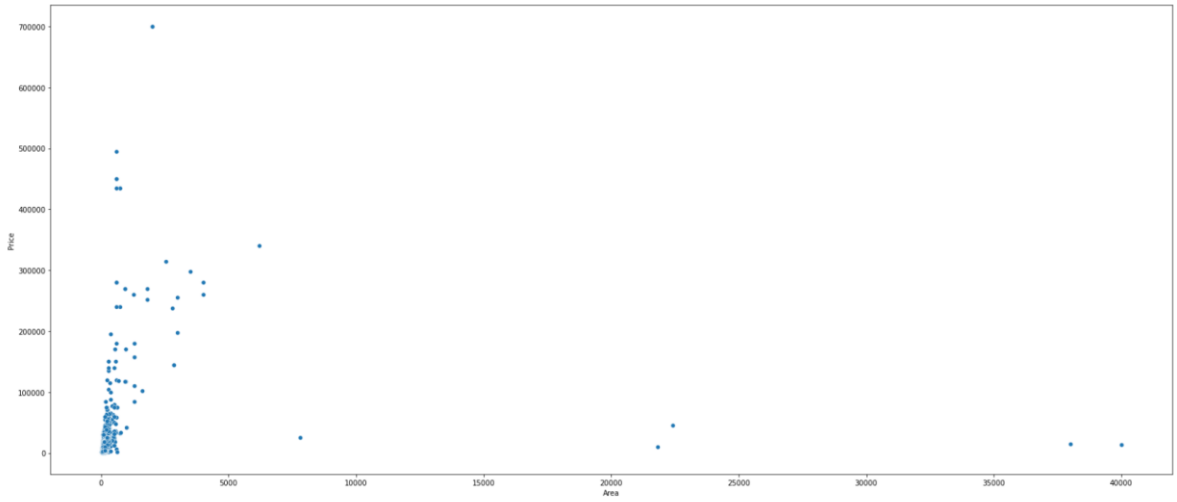
➢ Tỉ lệ giữa các quận Sơn Trà, Cẩm lệ, Liên Chiểu, Hải Châu có tỉ lệ tương đối bằng nhau chênh lệch khoảng 1-5.7%.

- Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng district,Area.



Hình 12. Mối tương qua giữa 2 đặc trưng district và Price.

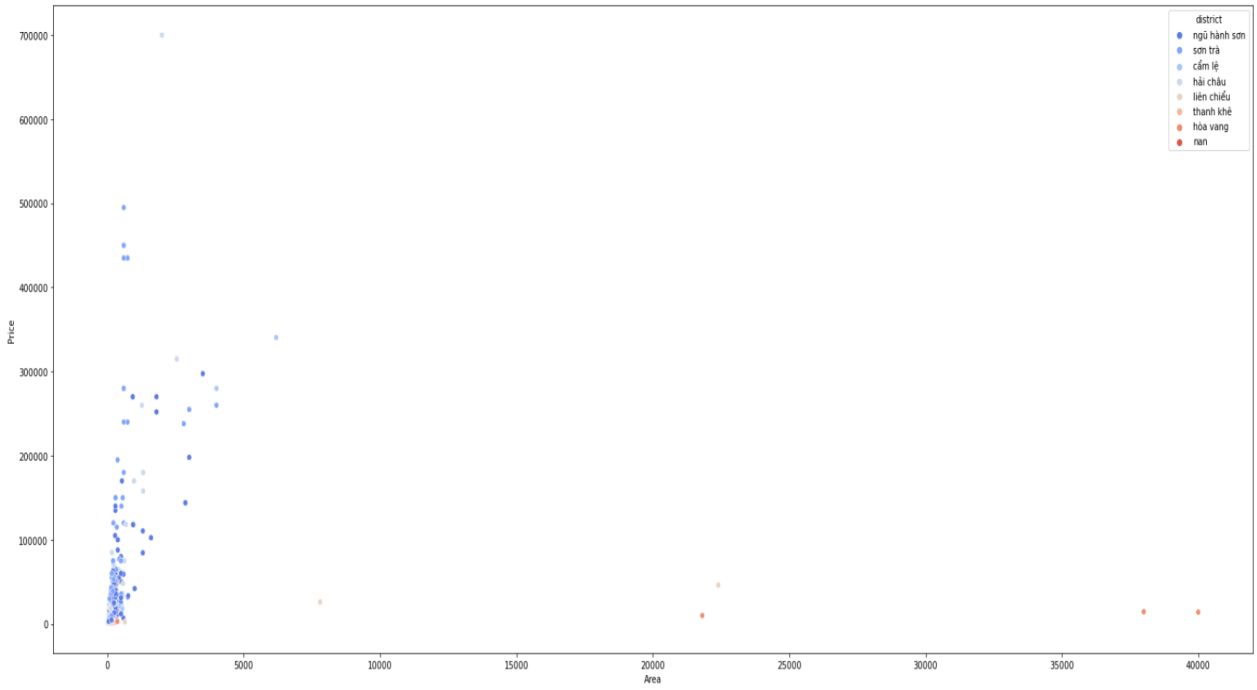
- Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng Price,Area.



Hình 13. Mối tương quan giữa 2 đặc trưng Price và Area.

➢ Hình trên cho ta thấy diện tích tỉ lệ thuận với khu vực, diện tích tăng giá cũng tăng.

- Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng Price,district,Area.



Hình 14. Mối tương quan giữa các đặc trưng Price, Area, Price.

- Trục tung biểu diễn giá, trục hành biểu diễn diện tích (m^2), các chấm nhỏ biểu diễn sự phân bố của các quận huyện .
- Từ hình 15 cho ta diện tích (Area) từ khoảng 0-5000 có giá trung bình cao nhất.
- Mức giá trung bình giao động từ 0-2 tỷ.
- Huyện Hòa Vang, Quận Thanh Khê có mức giá giao động thấp nhất .

3. Trích xuất đặc trưng

3.1. Xử lý dữ liệu trống

Dữ liệu trống xuất hiện trong dataset

```
Title        0
Address      2
Type         0
Area         2
Price      517
```

Hình 15. Số dữ liệu trống trong các đặc trưng.

Tiến hành xoá đi các hàng có dữ liệu bị trống

```
temp = data[~pd.isna(data).any(axis=1)].reset_index(drop = True)
temp
```
✓ 0.4s

	Title	Address	Type	Area	Price
0	Biệt thự Sunneva Island, vị trí ngã ba sông H...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	240.0	36000.0
1	Độc nhất vô nhị - Biệt thự đảo kim cương - Đô...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	300.0	45000.0
2	Sunneva Island - biệt thự đảo xanh - vùng đất...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	240.0	21120.0
3	Cần bán nhà phố thương mại Regal Pavillon Đà ...	Phường Hòa Cường Nam, Hải Châu	Bán shophouse, nhà phố thương mại	120.0	18700.0
4	Sở hữu vĩnh viễn biệt thự Sunneva Island đẳng...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	275.0	31000.0
...	...	...	...	...	...
1477	Chính chủ bán gấp căn Pavilon mặt tiền phố đi...	Hải Châu	Bán shophouse, nhà phố thương mại	120.0	18500.0
1478	Bán đất đường Hoàng Hiệp hướng đông nam gần P...	Phường Hòa Xuân, Cẩm Lệ	Bán đất	100.0	4000.0
1479	Mở bán Felicia Ocenview Apart căn hộ mặt biển...	Ngũ Hành Sơn	Bán condotel	50.0	2950.0
1480	Bán Gấp Lô Đất Khủng. Kiệt Otto Thái Thị Bôi	Phường Tân Chính, Thanh Khê	Bán đất	158.0	4450.0
1481	Bán lô biệt thự 150m2 ngang 7,5m MT Trương Mi...	Phường Hòa Hải, Ngũ Hành Sơn	Bán đất nền dự án	150.0	4700.0

1482 rows × 5 columns

```
temp.isnull().sum()
```
✓ 0.4s

```
Title      0
Address    0
Type       0
Area       0
Price      0
```

Hình 16. Dữ liệu sau khi xử lý dữ liệu trống.

3.2. Label Encoding

Thêm mới các cột dữ liệu:

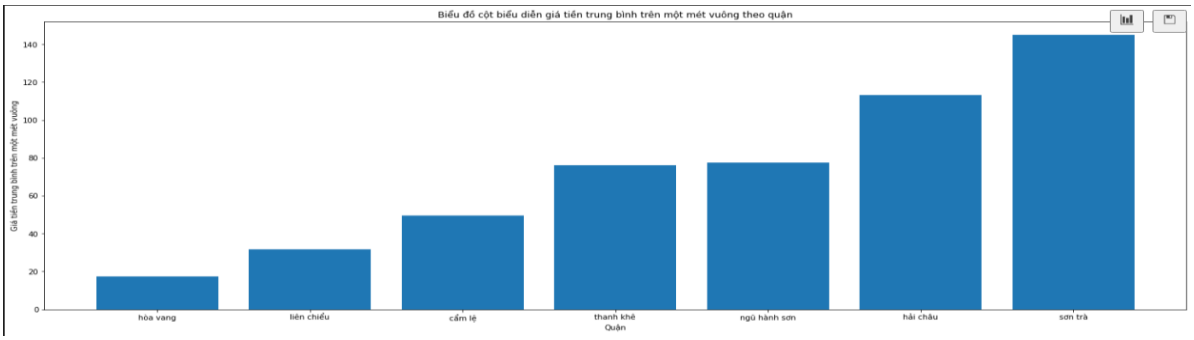
- district : quận.
- estate : loại bất động sản cần bán.
- price_m2 : giá tiền trên một mét vuông của loại bất động sản cần bán.

```
data_t['price_m2'] = data_t['Price'] / data_t['Area']
data_t.head()
```
✓ 0.1s Python

	Title	Address	Type	Area	Price	district	estate	price_m2
0	Biệt thự Sunneva Island, vị trí ngã ba sông H...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	240.0	36000.0	ngũ hành sơn	biệt thự	150.000000
1	Độc nhất vô nhị - Biệt thự đảo kim cương - Đô...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	300.0	45000.0	ngũ hành sơn	biệt thự	150.000000
2	Sunneva Island - biệt thự đảo xanh - vùng đất...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	240.0	21120.0	ngũ hành sơn	biệt thự	88.000000
3	Cần bán nhà phố thương mại Regal Pavillon Đà ...	Phường Hòa Cường Nam, Hải Châu	Bán shophouse, nhà phố thương mại	120.0	18700.0	hải châu	nhà phố thương mại	155.833333
4	Sở hữu vĩnh viễn biệt thự Sunneva Island đẳng...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	275.0	31000.0	ngũ hành sơn	biệt thự	112.727273

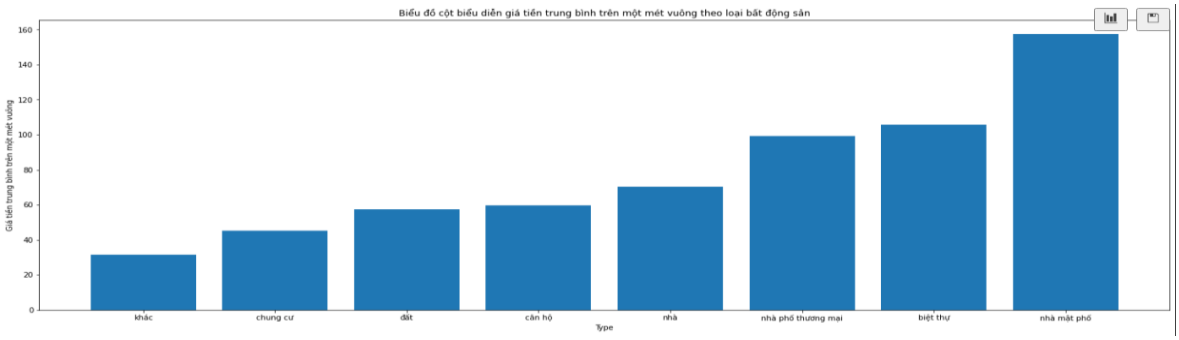
Hình 17. Thêm mới các cột dữ liệu.

Sau đó tiến hành Label Encoding cho các cột : district và estate.



Hình 18.Biểu đồ cột biểu diễn giá tiền trung bình trên một mét vuông theo quận.

Dựa vào biểu đồ cột trên ta tiến hành chuyển đổi cột district từ dữ liệu chữ sang dữ liệu số (từ 0 đến 6) theo thứ tự từ Hoà Vang cho đến Sơn Trà dựa theo giá tiền trên một mét vuông theo từng quận



Hình 19. Biểu đồ cột biểu diễn giá tiền trung bình trên một mét vuông theo loại bất động sản.

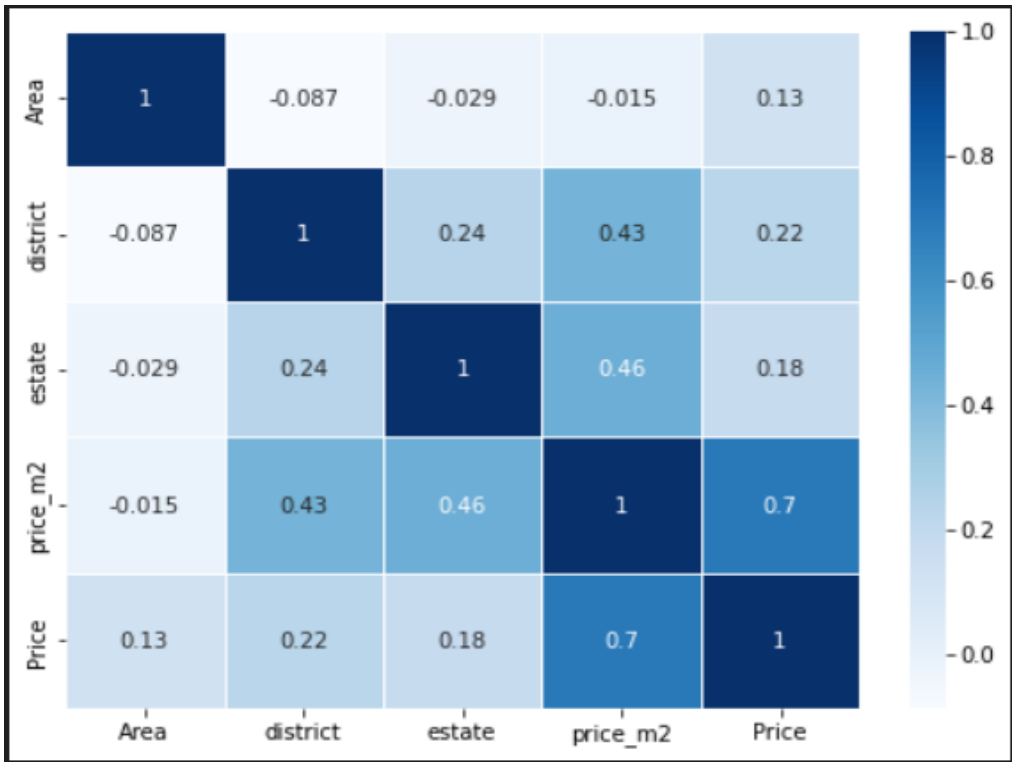
Tiếp theo cũng dựa vào biểu đồ trên ta cũng tiến hành đánh số (từ 0 đến 7) theo thứ tự từ các loại bất động sản khác cho đến nhà mặt phố dựa trên giá tiền trên một mét vuông theo loại bất động sản.

Kết quả sau khi Label Encoding:

	Title	Address	Type	Area	Price	district	estate	price_m2
0	Biệt thự Sunneva Island, vị trí ngã ba sông H...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	240.0	36000.0	4	6	150.000000
1	Độc nhất vô nhị - Biệt thự đảo kim cương - Đô...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	300.0	45000.0	4	6	150.000000
2	Sunneva Island - biệt thự đảo xanh - vùng đất...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	240.0	21120.0	4	6	88.000000
3	Cần bán nhà phố thương mại Regal Pavillon Đà ...	Phường Hòa Cường Nam, Hải Châu	Bán shophouse, nhà phố thương mại	120.0	18700.0	5	5	155.833333
4	Sở hữu vĩnh viễn biệt thự Sunneva Island đẳng...	Phường Hòa Quý, Ngũ Hành Sơn	Bán nhà biệt thự, liền kề	275.0	31000.0	4	6	112.727273

Hình 20. Kết quả sau khi label encoding.

3.3. Lựa chọn đặc trưng



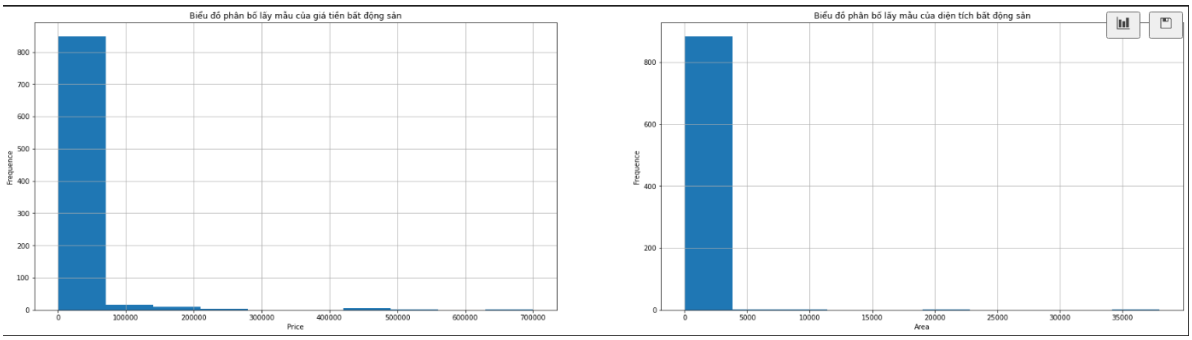
Hình 21. Ma trận tương quan giữa các đặc trưng.

Dựa vào ma trận tương quan trên ta tiến hành lựa chọn 5 đặc trưng :

- Area: diện tích của loại bất động sản (mét vuông).
- Price: giá tiền của loại bất động sản.
- district: quận trên địa bàn thành phố Đà Nẵng.
- estate: loại bất động sản.
- price_m2: giá tiền trên một mét vuông.

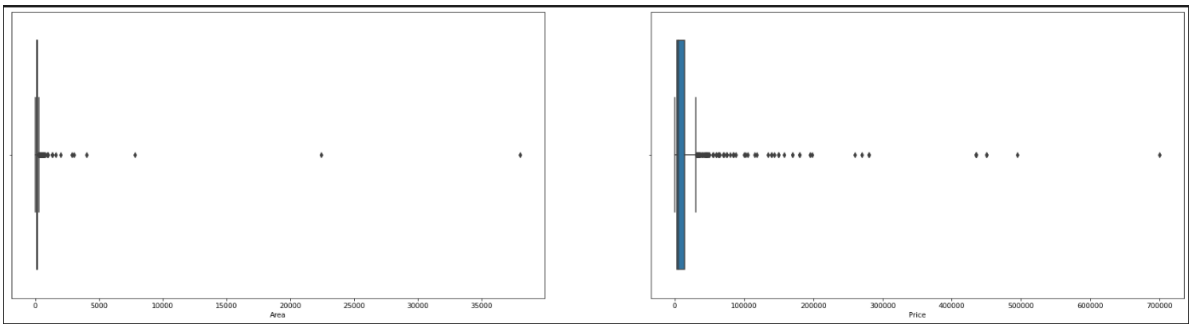
3.4. Outliers cho tập train

Biểu đồ phân bố lấy mẫu của Price và Area.



Hình 22. Biểu đồ phân bố lấy mẫu của 2 đặc trưng Price và Area.

Biểu đồ boxplot 2 đặc trưng Price và Area của tập train.



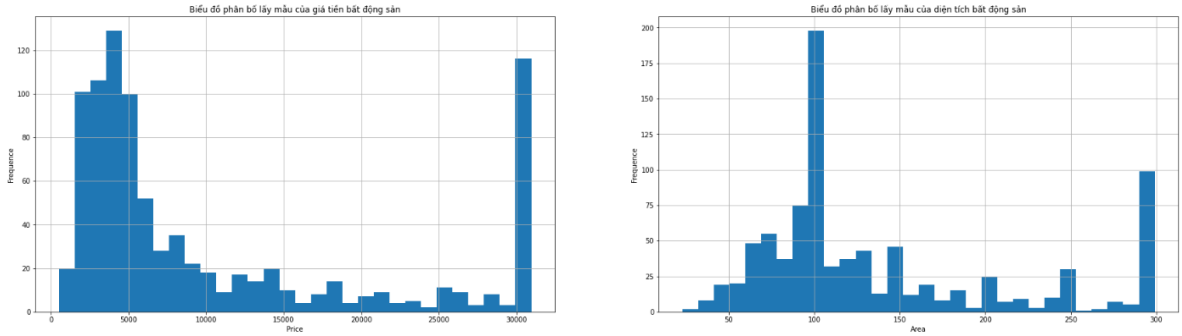
Hình 23. Biểu đồ boxplot của đặc trưng Price và Area.

- Nhìn vào các biểu đồ trên ta thấy Area và Price xuất hiện nhiều ngoại lệ và phân bố lệch trái.

```
def skewed_outliers(sr: Series):
    IQR = sr.quantile(0.75) - sr.quantile(0.25)
    upper_bridge = sr.quantile(0.75) + (IQR * 1.5)
    sr.loc[sr >= int(upper_bridge)] = int(upper_bridge)
    return sr
```

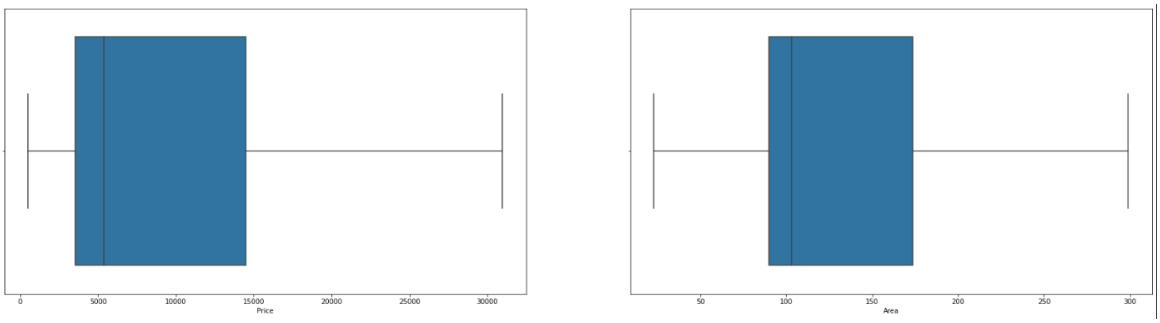
Hình 1524. Hàm xử lý ngoại lệ lệch trái.

Biểu đồ phân bố lấy mẫu sau khi xử lý ngoại lệ.



Hình 25. Biểu đồ phân lấy mẫu sau khi xử lý ngoại lệ của hai đặc trưng Price và Area.

Biểu đồ boxplot sau khi xử lý ngoại lệ.



Hình 26. Biểu đồ boxplot sau khi xử lý ngoại lệ của hai đặc trưng Price and Area.

3.5. Chuẩn hoá min-max-scaler

```
def MinMaxScaling(df: DataFrame, column: list):
    min_max = MinMaxScaler()
    df_minmax = pd.DataFrame(min_max.fit_transform(df[column]), columns=column)
    return df_minmax
```

✓ 0.7s

Hình 27. Hàm chuẩn hoá min-max-scaler.

Các tập dữ liệu sau khi chuẩn hoá 2 cột Area và price_m2:

- Dữ liệu không tiền xử lí.

	Area	price_m2	district	estate	Price
0	0.005490	0.181471	4	6	36000.0
1	0.006991	0.181471	4	6	45000.0
2	0.005490	0.106288	4	6	21120.0
3	0.002489	0.188545	5	5	18700.0
4	0.006366	0.136273	4	6	31000.0

Hình 28.Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 .

- Train.

	Area	price_m2	district	estate	Price
0	1.000000	0.181440	6	2	30961.0
1	0.391304	0.129137	3	7	14000.0
2	0.181159	0.080106	5	4	4850.0
3	0.260870	0.051070	1	2	4037.0
4	0.278986	0.039556	4	2	3300.0

Hình 29. Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 trên tập train .

- Val.

	Area	price_m2	district	estate	Price
0	0.002754	0.030038	4	2	2000.0
1	0.001699	0.047777	6	1	2350.0
2	0.000000	0.091379	4	3	2000.0
3	0.003214	0.175341	2	2	12750.0
4	0.003214	0.137248	5	7	9990.0

Hình 30. Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 trên tập val.

- Test.

	Area	price_m2	district	estate	Price
0	0.154317	0.072901	2	2	340450.0
1	0.001989	0.050890	2	7	3850.0
2	0.008742	0.050728	4	6	14200.0
3	0.002989	0.049080	4	2	5200.0
4	0.001989	0.063563	4	2	4800.0

Hình 31. Dữ liệu sau khi chuẩn hoá của hai đặc trưng Area và price_m2 trên tập test.

4. Mô hình hóa dữ liệu

4.1. Các mô hình

Các mô hình được dùng trong tiểu luận này đều là các mô hình có sẵn trong thư viện scikit-learn[2] của Python.

4.1.1. Linear Regression

- Lí thuyết

Linear Regression[3] (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. Hồi quy tuyến tính thuộc nhóm Supervised learning (học có giám sát).

Gọi $x = [x_1, x_2, \dots, x_n]$ là vector chưa thông tin input, y là một vô hướng biểu diễn output. y sẽ có dạng

$$y \approx f(x) = \hat{y}$$

$$f(x) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_0 \qquad (1)$$

Đặt $w = [w_0, w_1, \dots, w_n]^T$ là vector (cột) hệ số cần phải tối ưu và $\bar{x} = [1, x_1, x_2, \dots, x_n]$ là vector dữ liệu đầu vào mở rộng thì (1) có thể được viết lại dưới dạng:

$$y \approx \bar{x}w = \hat{y}$$

Với y là giá trị thực của outcome (dựa trên số liệu thống kê chúng ta có trong tập training data), $\hat{y}$ là giá trị mà mô hình Linear Regression dự đoán được.

- Tham số của mô hình
    - *fit_intercept*: Có tính intercept của mô hình không, nếu được set là False thì dữ liệu dự kiến sẽ được căn giữa. Mặc định là True.
    - *normalize*: Có chuẩn hóa hay không, tham số này sẽ bị bỏ qua khi *fit_intercept* bằng False. Mặc định là *deprecated*.
    - *copy_X:* Nếu bằng True thì dữ liệu sẽ được copy, ngược lại sẽ bị ghi đè. Mặc định là True.
    - *n_jobs:* Số job cần dùng để tính toán, dùng để tăng tốc độ trong trường hợp dữ liệu lớn. Mặc định là None.
    - *positive:* Khi có giá trị là True thì các hệ số bắt buộc phải là số dương, hỗ trợ cho các mảng dày đặt (dense arrays). Mặc định là False.

4.1.2. SVR

- Lí thuyết

SVR (Support Vector Regression) là một mô hình hồi quy sử dụng thuật toán Support Vector Machine (SVM, một thuật toán phân loại) để dự đoán giá trị của một biến liên tục.

- Tham số của mô hình
    - *kernel*: Chỉ định kiểu kernel sẽ được sử dụng trong thuật toán. Mặc định là *rbf.*
    - *degree*: Mức độ của hàm đa thức khi *kernel* bằng *poly*, bị bỏ qua bởi tất cả các giá trị *kernel* khác. Mặc định là 3.

- *gamma*: Hệ số nhân cho *kernel rbf, poly* và *sigmoid*. Mặc định là *scale*.
- *coef0*: Thuật ngữ độc lập cho hàm *kernel*, nó chỉ có tác dụng với *kernel poly* và *sigmoid*. Mặc định là 0.0.
- *tol:* Dung sai cho tiêu chí dừng. Mặc định là 10^{-3}.
- *C:* Tham số điều chỉnh, độ mạnh regularization tỉ lệ nghịch với C, C phải là số dương. Mặc định là 1.0.
- *epsilon*: Epsilon trong mô hình epsilon-SVR. Nó chỉ định epsilon-tube trong đó không có penalty nào được liên kết trong chức năng mất tập luyện với các điểm được dự đoán trong khoảng cách epsilon so với giá trị thực. Mặc định là 0.1.
- *shrinking*: Có sử dụng shrinking heuristic không. Mặc định là True.
- *cache_size*: Chỉ định kích thước của bộ nhớ đệm kernel (tính bằng MB). Mặc định là 200.
- *verbose*: Cho phép verbose output. Mặc định là False.
- *max_iter*: Giới hạn cứng về số lần lặp trong bộ giải mã. Mặc định là -1.

4.1.3. Random Forest

- Lí thuyết

Random Forest là thuật toán 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.

- Tham số của mô hình
  - *n_estimators*: Số lượng cây trong rừng. Mặc định là 100.
  - *criterion*: Chức năng đo lường chất lượng của một lần tách. Mặc định là *squared_error*
  - *max_depth*: Độ sâu tối đa của một cây. Mặc định là None.
  - *min_samples_split*: Số mẫu tối thiểu cần thiết để tách một nút bên trong. Mặc định là 2.
  - *min_samples_leaf*: Số mẫu tối thiểu cần thiết để có ở một nút lá. Mặc định là 1.
  - *min_weight_fraction_leaf*: Phần có trọng số tối thiểu của tổng trọng số (của tất cả các mẫu đầu vào) cần có ở một nút lá. Các mẫu có trọng lượng bằng nhau khi *sample_weight* không được cung cấp. Mặc định là 0.
  - *max_features*: Số lượng các tính năng cần xem xét khi tìm kiếm sự phân chia tốt nhất. Mặc định là *auto.*
  - *max_leaf_nodes*: Trồng cây với *max_leaf_nodes* theo cách tốt nhất. Các nút tốt nhất được định nghĩa là giảm tạp chất tương đối. Nếu bằng None thì không giới hạn số nút lá. Mặc định là None.
  - *min_impurity_decrease*: Một nút sẽ bị tách nếu sự phân tách này làm giảm tạp chất lớn hơn hoặc bằng giá trị này. Mặc định là 0.

- *bootstrap*: Các mẫu bootstrap có được sử dụng khi xây dựng cây hay không. Nếu bằng False, toàn bộ tập dữ liệu được sử dụng để xây dựng từng cây. Mặc định bằng True.
- *oob_score*: Có sử dụng các mẫu ngoài túi để ước tính điểm tổng quát hay không. Chỉ khả dụng nếu bootstrap = True. Mặc định là False.
- *n_jobs*: Số lượng công việc phải chạy song song. Mặc định là None.
- *random_state*: Mặc định là None.
- *verbose*: Kiểm soát độ chi tiết khi điều chỉnh và dự đoán. Mặc định là 0.
- *warm_start*: Khi bằng True, sử dụng lại giải pháp của lệnh gọi trước đó để phù hợp và thêm nhiều công cụ ước tính hơn vào nhóm, nếu không, chỉ vừa với một khu rừng hoàn toàn mới. Mặc định bằng False.
- *ccp_alpha*: Tham số độ phức tạp được sử dụng để cắt tỉa Minimal Cost-Complexity. Mặc định là 0.
- *max_samples*: Nếu bootstrap là True, thì số lượng mẫu sẽ lấy từ X để đào tạo mỗi công cụ ước lượng cơ sở. Mặc định là None.

4.1.4. Ada Boost

- Lí thuyết

AdaBoost, hoặc Adaptive Boost, cũng là một thuật toán tổng hợp sử dụng các phương pháp bagging và boosting để phát triển một công cụ dự đoán nâng cao.

AdaBoost tương tự như Random Forest theo nghĩa là các dự đoán được lấy từ nhiều cây quyết định. Tuy nhiên, có ba điểm khác biệt chính làm cho AdaBoost trở nên độc đáo:

1. Đầu tiên, AdaBoost tạo ra một khu rừng gốc cây thay vì cây cối. Gốc là cây chỉ được tạo thành từ một nút và hai lá.
2. Thứ hai, các gốc cây được tạo ra không có trọng số như nhau trong quyết định cuối cùng (dự đoán cuối cùng). Những gốc cây tạo ra nhiều lỗi hơn sẽ có ít ý nghĩa hơn trong quyết định cuối cùng.
3. Cuối cùng, thứ tự thực hiện các gốc cây là rất quan trọng, bởi vì mỗi gốc cây nhằm mục đích giảm thiểu các lỗi mà (các) gốc cây trước đó đã mắc phải.

- Tham số của mô hình
    - *base_estimator*: Công cụ ước tính cơ sở mà từ đó nhóm tăng cường được xây dựng. Mặc định là None.
    - *n_estimators*: Số lượng công cụ ước tính tối đa mà tại đó quá trình thúc đẩy bị chấm dứt. Trong trường hợp hoàn toàn phù hợp, thủ tục học máy được dừng lại sớm. Mặc định là 50.
    - *learning_rate*: Trọng số được áp dụng cho mỗi bộ hồi quy tại mỗi lần lặp tăng cường. Tỷ lệ học tập cao hơn làm tăng sự đóng góp của mỗi bộ hồi quy. Mặc định là 1.

- *loss*: Sử dụng khi cập nhật weights sau mỗi lần lặp tăng cường. Mặc định là *linear*.
- *random_state*: Mặc định là None.

4.1.5. Gradiant Boost

- Lí thuyết

Gradient Boost cũng là một thuật toán tổng hợp sử dụng các phương pháp thúc đẩy (boosting) để phát triển một công cụ dự đoán nâng cao. Theo nhiều cách, Gradient Boost tương tự như AdaBoost, nhưng có một số điểm khác biệt chính:

1. Không giống như AdaBoost xây dựng các gốc cây, Gradient Boost xây dựng các cây thường có 8–32 lá.
2. Gradient Boost xem vấn đề tăng cường (boosting problem) là một vấn đề tối ưu hóa, trong đó nó sử dụng một hàm mất mát (loss function) và cố gắng giảm thiểu lỗi. Đây là lý do tại sao nó được gọi là Gradient boost, vì nó được lấy cảm hứng từ sự giảm dần độ dốc (gradient descent).
3. Cuối cùng, cây được sử dụng để dự đoán lượng dư của các mẫu (dự đoán trừ thực tế).

- Tham số của mô hình
  - loss: Hàm mất mát được tối ưu hóa. Mặc định là squared_error.
  - learning_rate: learning_rate thu hẹp mức đóng góp của mỗi cây bằng learning_rate. Có một sự cân bằng giữa learning_rate và n_estimators. Mặc định là 0.1.
  - n_estimators: Số lượng các giai đoạn thúc đẩy để thực hiện. Mặc định là 100.
  - subsample: Phần mẫu được sử dụng để phù hợp với từng base learner. Mặc định là 1.
  - criterion: Đo lường chất lượng của một lần tách. Mặc định là friedman_mse.
  - min_samples_split: Số mẫu tối thiểu cần thiết để tách một nút bên trong. Mặc định là 2.
  - min_samples_leaf: Số mẫu tối thiểu cần thiết để có ở một nút lá. Mặc định là 1.
  - min_weight_fraction_leaf: Phần có trọng số tối thiểu của tổng trọng số (của tất cả các mẫu đầu vào) cần có ở một nút lá. Các mẫu có trọng lượng bằng nhau khi *sample_weight* không được cung cấp. Mặc định là 0.
  - max_depth: Độ sâu tối đa của các công cụ ước tính hồi quy riêng lẻ. Mặc định là 3.
  - min_impurity_decrease: Một nút sẽ bị tách nếu sự phân tách này làm giảm tạp chất lớn hơn hoặc bằng giá trị này. Mặc định là 0.

- init: Một đối tượng ước tính được sử dụng để tính toán các dự đoán ban đầu. Mặc định là None.
- random_state: Mặc định là None.
- max_features: Số lượng các tính năng cần xem xét khi tìm kiếm sự phân chia tốt nhất. Mặc định là None.
- alpha: alpha-quantile của hàm mất mát huber và hàm mất mát quantile. Mặc định là 0.9.
- verbose: Cho phép verbose output. Mặc định là 0.
- max_leaf_nodes: Trồng cây với max_leaf_nodes theo cách tốt nhất. Các nút tốt nhất được định nghĩa là giảm tạp chất tương đối. Nếu Không thì không giới hạn số nút lá. Mặc định là None.
- warm_start: Khi bằng True, sử dụng lại giải pháp của lệnh gọi trước đó để phù hợp và thêm nhiều công cụ ước tính hơn vào nhóm, nếu không, chỉ vừa với một khu rừng hoàn toàn mới. Mặc định bằng False.
- validation_fraction: Tỷ lệ dữ liệu đào tạo để dành làm xác nhận được thiết lập để dừng sớm. Mặc định là 0.1.
- n_iter_no_change: sử dụng để quyết định xem liệu việc dừng sớm có được sử dụng để chấm dứt đào tạo khi điểm xác thực không được cải thiện hay không. Mặc định là None.
- tol: Dung sai cho tiêu chí dừng. Mặc định là 10^{-4}.
- ccp_alpha: Tham số độ phức tạp được sử dụng để cắt tỉa Minimal Cost-Complexity. Mặc định là 0.

4.2. Chia dữ liệu

Dữ liệu sau khi xử lí dữ liệu trống gồm 1482 mẫu được chia thành 3 tập: huấn luyện (train), xác thực (validate) và kiểm thử (test) theo tỉ lệ 6:2:2.

	Area	price_m2	district	estate	Price
0	1.000000	0.181440	6	2	30961.0
1	0.391304	0.129137	3	7	14000.0
2	0.181159	0.080106	5	4	4850.0
3	0.260870	0.051070	1	2	4037.0
4	0.278986	0.039556	4	2	3300.0
...	...	...	...	...	...
883	0.134058	0.191545	4	4	9500.0
884	0.931159	0.114742	1	6	26600.0
885	0.152174	0.139462	6	7	7500.0
886	0.550725	0.172778	6	2	25000.0
887	0.655797	0.072298	2	2	12240.0

888 rows × 5 columns

Hình 32. Dữ liệu huấn luyện.

	Area	price_m2	district	estate	Price
0	0.002754	0.030038	4	2	2000.0
1	0.001699	0.047777	6	1	2350.0
2	0.000000	0.091379	4	3	2000.0
3	0.003214	0.175341	2	2	12750.0
4	0.003214	0.137248	5	7	9990.0
...	...	...	...	...	...
292	0.005509	0.114383	5	5	12500.0
293	0.002112	0.078909	1	7	4380.0
294	0.008263	0.143958	4	2	22000.0
295	0.005233	0.090421	6	7	9500.0
296	0.004912	0.028079	1	2	2850.0

297 rows × 5 columns

Hình 33. Dữ liệu xác thực.

	Area	price_m2	district	estate	Price
0	0.154317	0.072901	2	2	340450.0
1	0.001989	0.050890	2	7	3850.0
2	0.008742	0.050728	4	6	14200.0
3	0.002989	0.049080	4	2	5200.0
4	0.001989	0.063563	4	2	4800.0
...	...	...	...	...	...
292	0.000888	0.132929	6	7	5600.0
293	0.003419	0.042556	4	2	5070.0
294	0.003739	0.127436	6	7	16300.0
295	0.000863	0.043190	1	5	1800.0
296	0.003654	0.057583	2	2	7250.0

297 rows × 5 columns

Hình 34. Dữ liệu kiểm thử.

4.3. Tham số của quá trình huấn luyện mô hình

Quá trình huấn luyện diễn ra với 5 mô hình với tất cả các tham số mặc định đã được nêu chi tiết ở phần 4.1.

```
models = {
    "LinearRegression": LinearRegression(),
    "SVR": SVR(),
    "RandomForest": RandomForestRegressor(),
    "GradiantBoost": GradientBoostingRegressor(),
    "AdaBoost": AdaBoostRegressor(),
}
```

Hình 35. Mô hình huấn luyện.

4.4. Hiệu suất của các mô hình trên tập Huấn luyện, Xác thực và Kiểm thử dựa trên độ đo RMSE

Quá trình mô hình hóa dữ liệu dùng ba loại dữ liệu để kiểm chứng hiệu quả của bước feature engineer và mô hình hóa dữ liệu là: dữ liệu không được tiền xử lí, dữ liệu được chuẩn hóa và dữ liệu kết hợp xử lí ngoại lệ ở tập huấn luyện với chuẩn hóa.

	LinearRegression	SVR	RandomForest	GradientBoost	AdaBoost
Train non preprocess	32909.672101	55147.703808	6201.833728	1604.357525	10580.017121
Train scale	32909.672101	55138.983749	6056.095128	1604.357525	20852.161581
Train scale and outliers	4312.144499	11125.257440	625.066596	871.913925	3021.833199

Hình 36. Bảng thống kê RMSE của các tập dữ liệu huấn luyện.

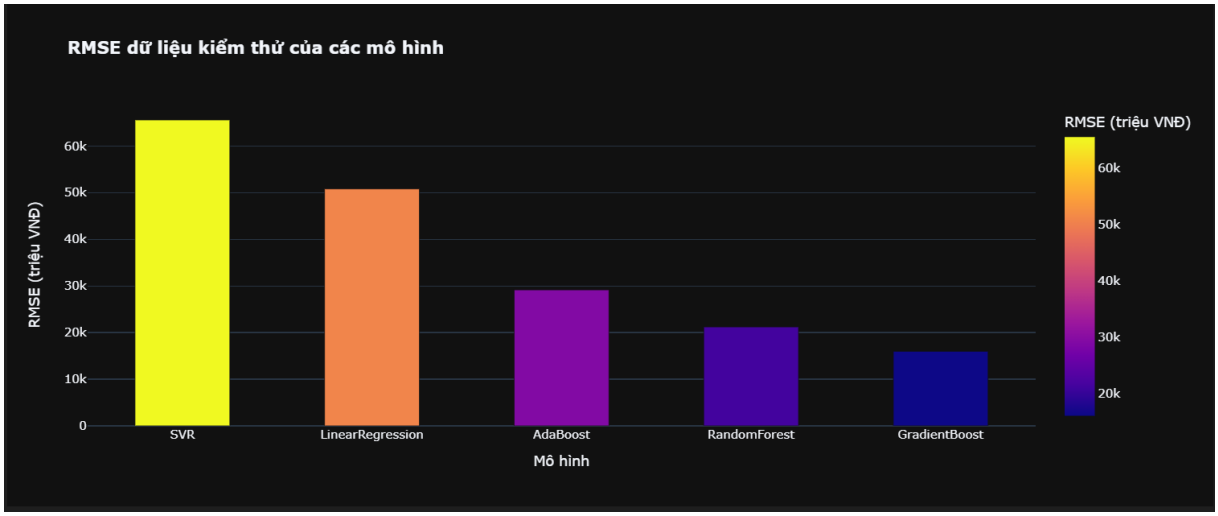
	LinearRegression	SVR	RandomForest	GradientBoost	AdaBoost
Validate non preprocess	34203.665193	50394.203858	11122.032051	11030.247390	15607.697771
Validate scale	34203.665193	50392.137792	11202.999284	9157.896153	24542.905785
Validate scale and outliers	48945.127981	50392.416367	48071.300404	48706.625031	47222.587709

Hình 37. Bảng thống kê RMSE của các tập dữ liệu xác thực.

- Qua hai hình trên, ta thấy dữ liệu kết hợp xử lí ngoại lệ ở tập huấn luyện với chuẩn hóa có giá trị chênh lệch trung bình của giá dự đoán từ mô hình và giá trị thực tế (RMSE) cao hơn nhiều lần ở phần lớn các mô hình (4/5 mô hình) ở tập dữ liệu xác thực. Vì vậy không dùng dữ liệu này để đưa vào mô hình thực tế.
- Gradient Boost là mô hình có RMSE thấp nhất với dữ liệu được chuẩn hóa ở tập dữ liệu xác thực nên sẽ chọn mô hình và loại dữ liệu này để kiểm thử.

	LinearRegression	SVR	RandomForest	GradientBoost	AdaBoost
Test non preprocess	50788.141738	65563.910377	21407.159435	15484.698552	23324.652437
Test scale	50788.141738	65567.962479	21212.397859	15980.257173	29148.960745

Hình 38. Bảng thống kê RMSE của các tập dữ liệu kiểm thử.



Hình 39. Biểu đồ RMSE dữ liệu kiểm thử tập dữ liệu được chuẩn hóa của mô hình Gradient Boost.

- Mô hình Gradient Boost vẫn cho kết quả RMSE tốt nhất (~16 tỉ VNĐ).

Sau khi tìm được mô hình và tập dữ liệu tốt nhất để sử dụng, tiếp tục dùng Randomized Search CV của thư viện scikit learn để tìm ra bộ siêu tham số tốt nhất cho mô hình vừa tìm được.

```
RMSE: 13409.276341552113
Bộ siêu tham số tốt nhất: {'n_estimators': 1500, 'min_samples_split': 2, 'max_depth': 3}
RMSE của dữ liệu kiểm thử: 15110.11973714083
RMSE cải thiện được: 870.1374359641959
```

Hình 40. Kết quả tìm kiếm bộ siêu tham số.

4.5. Áp dụng mô hình vào thực tế

Sau khi tìm được bộ siêu tham số, áp dụng vào mô hình Gradient Boost rồi bắt đầu huấn luyện tương tự với những bước đã làm trước đây.

Để áp dụng vào thực tế, đầu tiên chọn ngẫu nhiên một bất động sản trong tập dữ liệu kiểm thử và đưa vào mô hình. Bất động sản ngẫu nhiên được chọn là bất động sản thứ 1082 trong tập dữ liệu kiểm thử.

Bất động sản được chọn là:

	Area	price_m2	district	estate
1082	0.002301	0.03299	4	2

Hình 41. Bất động sản được chọn ngẫu nhiên trong tập dữ liệu kiểm thử.

Giá của bất động sản này là 3 tỉ 100 triệu VNĐ, giá mô hình dự đoán của bất động sản này là: 3 tỉ 182 triệu VNĐ, sai số là 82 triệu VNĐ.

5. Kết luận

5.1. Kết quả đạt được

- Biết được đặc trưng giá bất động sản theo từng khu vực, từng loại bất động sản ở Đà Nẵng.
- Biết được các yếu tố ảnh hưởng đến giá bất động sản ở Đà Nẵng.
- Xây dựng được mô hình có thể dùng để tham khảo trong thực tế với sai số không quá lớn.

5.2. Hướng phát triển

- Thu thập thêm dữ liệu đặc trưng của từng loại bất động sản.
- Tìm hiểu, áp dụng, so sánh thêm nhiều mô hình để tìm ra được mô hình cho ra sai số dự đoán tốt nhất để có thể triển khai trong thực tế

6. Tài liệu tham khảo

[1] Hướng dẫn lấy dữ liệu web, Web Crawling với Selenium - Python - WebDrive, https://sudo.vn/blog/huong-dan-lay-du-lieu-web-web-crawling-voi-selenium-python-webdrive.html

[2] Thư viện học máy Scikit learn, https://scikit-learn.org/0.21/documentation.html

[3] Linear Regression, https://machinelearningcoban.com/2016/12/28/linearregression/