

Báo Cáo Khoa Học Dữ Liệu

Đề tài: Dự đoán giá và phân loại
đất nền ở Đà Nẵng

Nhóm 12 :
Nguyễn Quốc Cường
Võ Đức Việt

TÓM TẮT ĐỀ TÀI

- Đề tài “Dự đoán và phân loại giá đất nền ở Đà Nẵng” giải quyết vấn đề biến động giá đất và sự thiếu hụt thông tin chính xác, gây rủi ro khi quyết định đầu tư. Nghiên cứu này bắt đầu bằng việc thu thập và xử lý các dữ liệu từ nhiều nguồn khác nhau.
- Dữ liệu sau đó được phân tích bằng các công cụ thống kê và thuật toán như hồi quy tuyến tính, phân cụm K-means, phân lớp để xây dựng mô hình dự đoán và phân loại giá đất.
- Kết quả đạt được cho thấy mô hình dự đoán đạt độ chính xác cao, giúp dự đoán giá đất một cách hiệu quả. Các giá được phân loại rõ ràng, tạo điều kiện cho việc so sánh và đánh giá giá trị đất nền.

Nội Dung

- 1. Giới Thiệu**
- 2. Thu Thập Và Mô Tả Dữ Liệu**
- 3. Trích Xuất Đặc Trưng**
- 4. Mô Hình Hóa Dữ Liệu**
- 5. Kết Luận**

1

GIỚI THIỆU

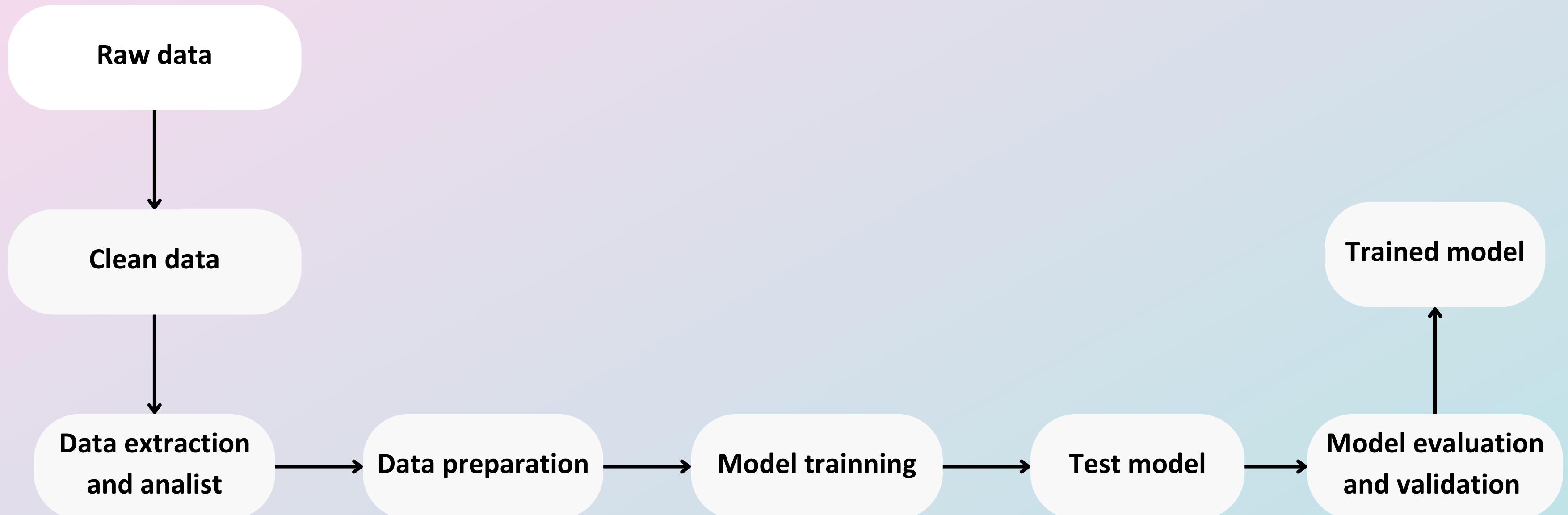
1. Giới Thiệu

Giới thiệu các bài toán:

- **Mục tiêu:** xây dựng mô hình dự đoán giá và phân loại đất nền ở Đà Nẵng dựa trên các yếu tố như giá, diện tích, loại hình đất, hướng đất, chiều ngang, chiều rộng, quận.
- **Giải pháp:**
 1. Dự đoán giá đất nền: thu thập dữ liệu, xử lý và làm sạch, xây dựng các mô hình và tối ưu hóa
 2. Phân loại đất nền: thu thập dữ liệu, xử lý và làm sạch, phân tích dữ liệu, phân cụm dữ liệu, Phân loại dữ liệu dựa trên kết quả phân cụm

1.Giới Thiệu

Sơ đồ khối:



2

THU THẬP VÀ MÔ TẢ DỮ LIỆU

2.1 Thu Thập Dữ Liệu

Nguồn dữ liệu:

- Train:
 - trang <https://www.nhatot.com>
 - trang <https://nhadat24h.net>
- Test:
 - trang <https://www.nhatot.com>

Công cụ thu thập:

- Sử dụng Selenium để thu thập liên kết các bài đăng từ danh sách trang.
- Sử dụng BeautifulSoup và Selenium để thu thập thông tin chi tiết từ từng liên kết bài đăng.

2.1 Thu Thập Dữ Liệu

Đầu vào

- URL của trang web cần thu thập dữ liệu
- Số lượng trang cần duyệt
- Các trường thông tin cần thu thập: Bao gồm giá, địa chỉ, diện tích, giá/m², hướng đất, loại hình đất, chiều ngang, chiều dài.

Đầu ra

- Danh sách các liên kết bài đăng: Được lưu trữ trong file CSV (link_nhatot_test.csv) chứa các liên kết đến từng bài đăng chi tiết.
- Thông tin chi tiết của từng bài đăng: Được lưu trữ trong file CSV (data_crawl_test.csv) chứa các thông tin như giá, địa chỉ, diện tích, giá/m², hướng đất, loại hình đất, chiều ngang, chiều dài.

2.1 Thu Thập Dữ Liệu

Ví dụ minh họa:

The screenshot shows a web browser displaying a real estate website for land sales in Da Nang. The URL in the address bar is `nhatot.com/mua-ban-dat-da-nang?page=4`. The page title is "Mua Bán Đất Tại Đà Nẵng Chính Chủ, Giá Rẻ Tháng 05/2024". The interface includes a header with navigation links like Chợ Tốt, Nhà Tốt, Chợ Tốt Xe, etc., and a search bar. Below the header are filtering options for location (Đà Nẵng), type (Mua bán), and category (Đất). The main content area shows two land listings. Listing 3 (highlighted in the screenshot) is for a plot of land in Huyện Hòa Vang, measuring 150 m² at a price of 1,95 tỷ | 13 tr/m². Listing 4 is for a plot of land in Quận Liên Chiểu, measuring 60 m² at a price of 2,03 tỷ | 33,83 tr/m². The developer tools (Elements tab) are open on the right side of the browser, showing the HTML code for the listed items. The selected item's code is visible, including its class names and attributes.

2.1 Thu Thập Dữ Liệu

Ví dụ minh họa:

The screenshot shows a real estate listing for a plot of land on the website nhatot.com. The listing details a plot of land located at Tô Hiệu, Phường Hòa Minh, Quận Liên Chiểu, Đà Nẵng. The plot is 60 m² and is listed for 2.03 tỷ. The listing includes information about orientation (Bắc), soil type (Đất thổ cư), dimensions (122.21 x 20), and location (Mặt tiền).

The developer tools' Elements tab is open, displaying the HTML structure of the page. A specific element, a media object with the class "media-body.media-middle", is selected. This element contains the width value "122.21 x 20". The DOM tree also shows other elements related to the listing, such as "AdParam_adParamTitle_bU_w" and "AdParam_adParamContainerVeh_Vz4Zt".

```
<div>...</div>
> <div class="DetailView_adviewPtyItem__V_sof">...</div>
<div class="DetailView_adviewPtyItem__V_sof">
  <div class="AdParam_adParamTitle_bU_w">Đặc điểm bất động sản</div>
  <div class="AdParam_adParamContainerVeh_Vz4Zt" flex>
    <div class="col-xs-6 no-padding AdParam_adParamItemVeh_mCpPS" da...
      <div class="AdParam_adMediaParam_3epxo" flex>
        <div class="media-left media-top" ...</div>
        <div class="media-body media-middle">
          <span>Chiều ngang: </span>
          <span itemprop="width" class="AdParam_adParamValue_IfaYa"...
        </span>
      </div>
    </div>
  </div>
</div>
> <div class="col-xs-6 no-padding AdParam_adParamItemVeh_mCpPS" da...
> <div class="col-xs-6 no-padding AdParam_adParamItemVeh_mCpPS" da...
</div>
```

2.1 Thu Thập Dữ Liệu

The screenshot shows a real estate listing on the website nhatot.com. The main content includes a large image of a land title document (Sổ Đất) from the Land and Environment Department of Da Nang City, dated April 19, 2023. Below the image, the listing details a plot of land in Khanh Nam, Quy Nhon, Da Nang, measuring 98x20 meters. The price is listed as 1.67 tỷ. The listing is categorized under 'Nhà Đất' (Land). The browser's developer tools (Elements tab) are open, showing the HTML structure of the page, specifically focusing on the land title document's details.

```
1 ,links  
1588 1586,https://nhatot.com//mua-ban-dat-quan-ngu-hanh-son-da-nang/115150522.htm  
1589 1587,https://nhatot.com//mua-ban-dat-quan-ngu-hanh-son-da-nang/116397061.htm  
1590 1588,https://nhatot.com//mua-ban-dat-quan-lien-chieu-da-nang/115593243.htm  
1591 1589,https://nhatot.com//mua-ban-dat-quan-hai-chau-da-nang/116396454.htm  
1592 1590,https://nhatot.com//mua-ban-dat-quan-ngu-hanh-son-da-nang/116395994.htm
```

```
1587 5 tỷ - 126 m2,"Đường Hồ Sĩ Tân, Phường Nại Hiên Đông, Quận Sơn Trà  
1588 "4,8 tỷ - 100 m2","Đường Mỹ Đa Tây 5, Phường Khuê Mỹ, Quận Ngũ Hành  
1589 "4,4 tỷ - 62 m2","Đường Tôn Thất Thiệp, Phường Mỹ An, Quận Ngũ Hành  
1590 
```

2.2. Mô Tả Và Trực Quan Hoá Dữ Liệu

Tổng quan về tập dữ liệu:

Tập huấn luyện:

Số mẫu dữ liệu
1290
Số đặc trưng
9

Tên cột	Số mẫu trống	Kiểu dữ liệu
Gia	0	float64
Dia chi	0	object
Dien tích	0	float64
Gia/m ²	0	float64
Huong dat	334	object
Loai hinh dat	0	object
Chieu ngang	0	float64
Chieu dai	0	float64
Quan	0	object

Tập kiểm thử:

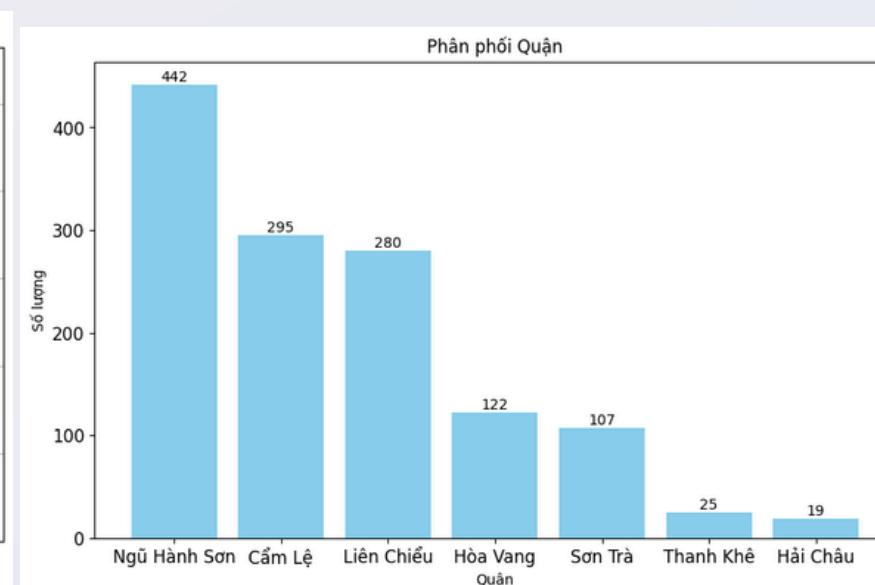
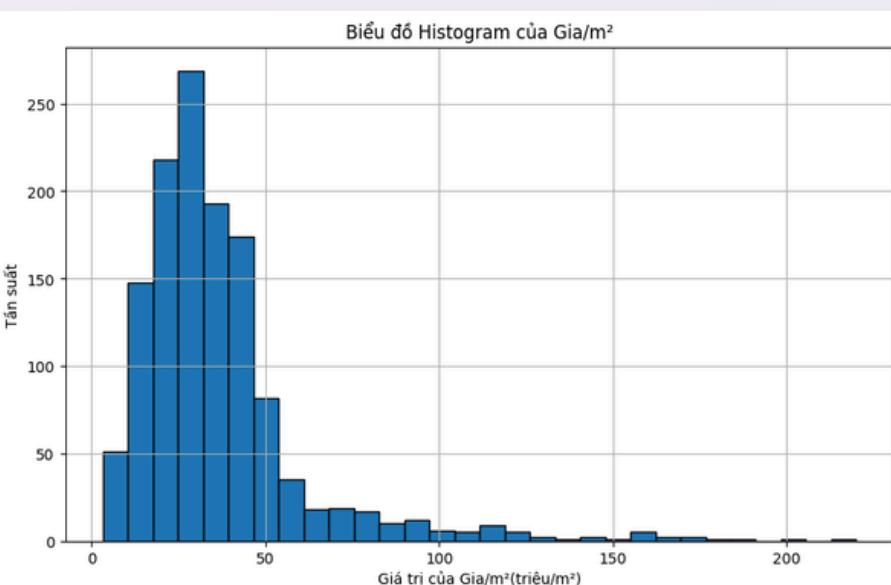
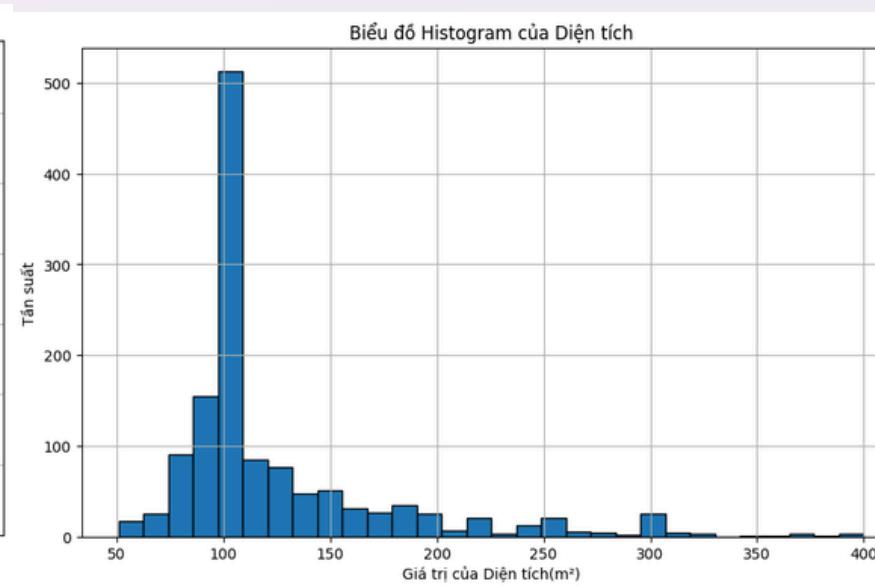
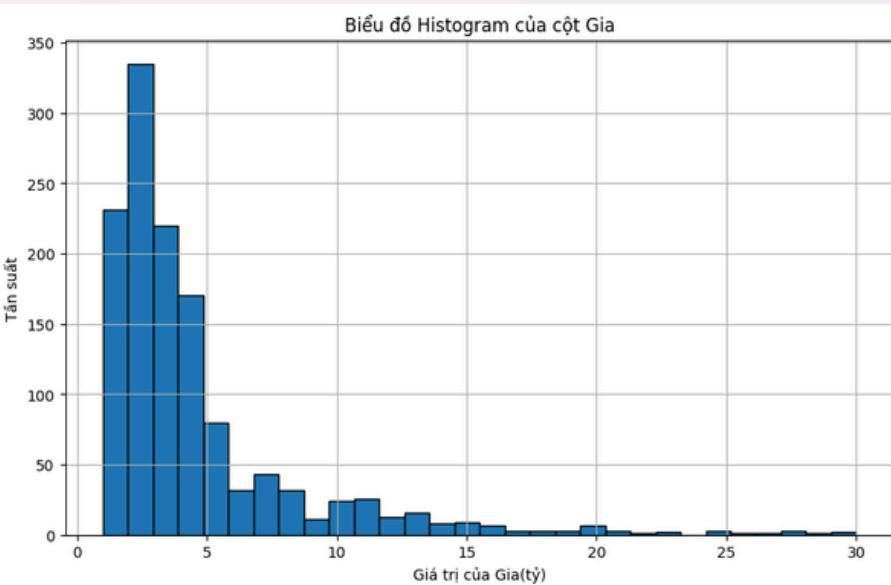
Số mẫu dữ liệu
419
Số đặc trưng
9

Tên cột	Số mẫu trống	Kiểu dữ liệu
Gia	0	float64
Dia chi	0	object
Dien tích	0	float64
Gia/m ²	0	float64
Huong dat	77	object
Loai hinh dat	0	object
Chieu ngang	0	float64
Chieu dai	0	float64
Quan	0	object

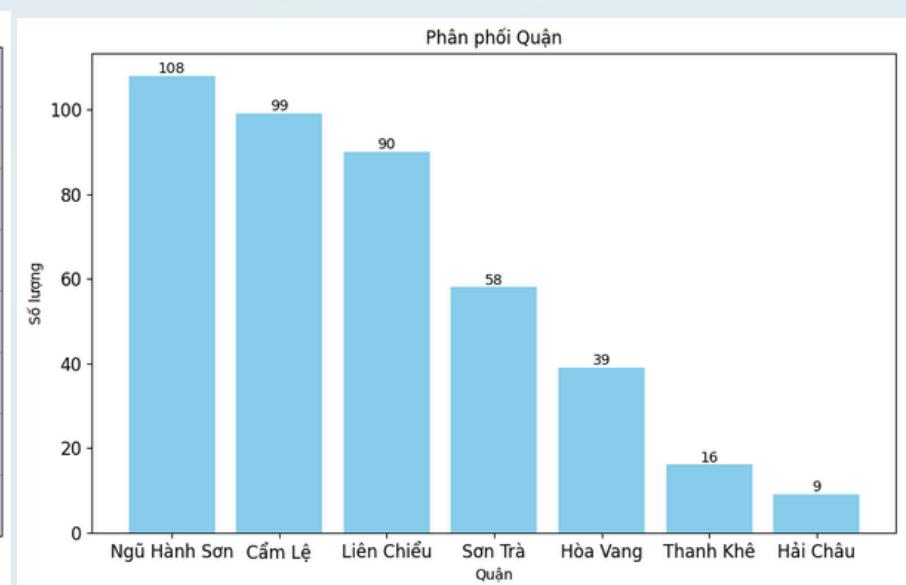
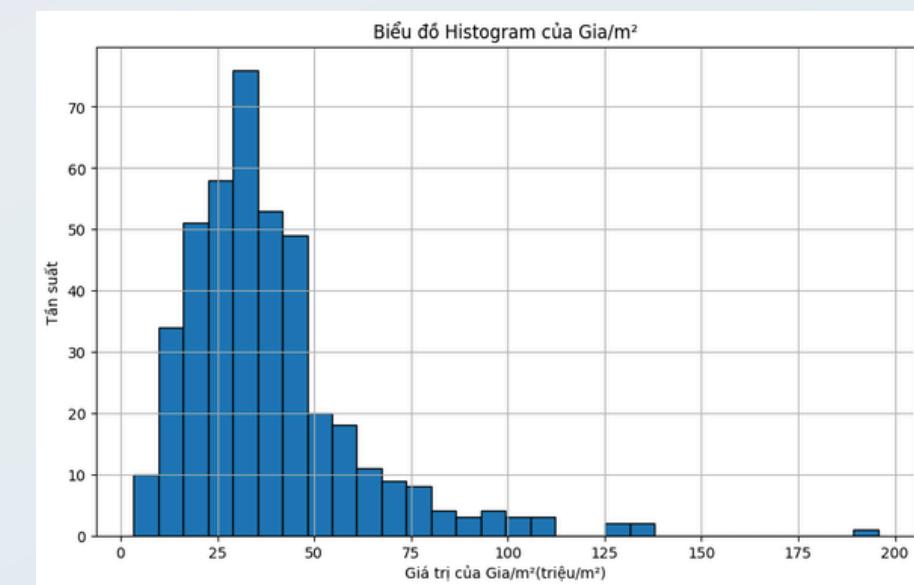
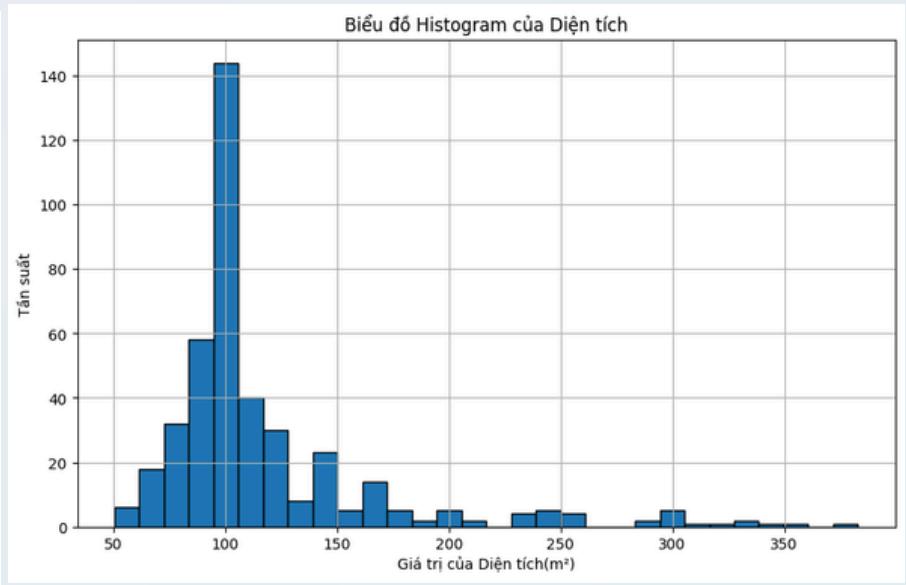
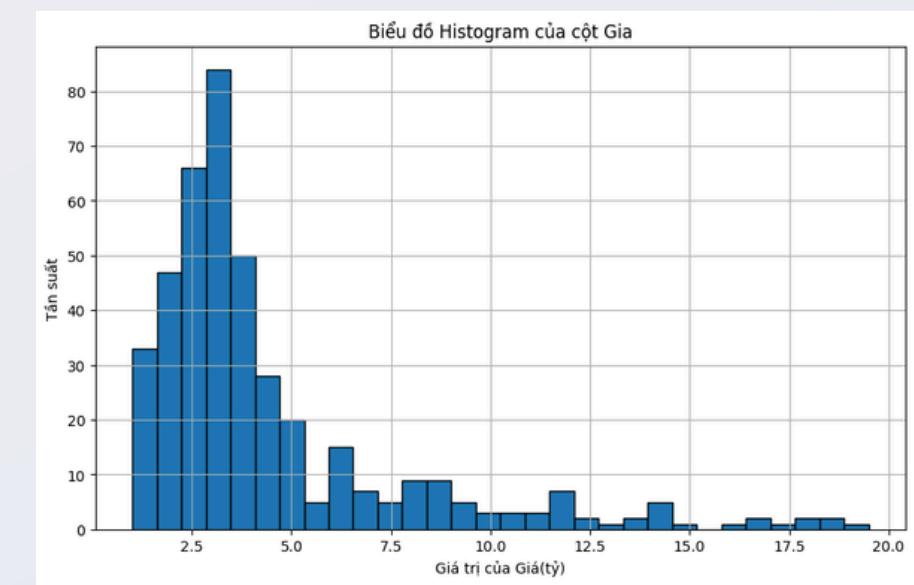
2.2. Mô Tả Và Trực Quan Hóa Dữ Liệu

Tổng quan về tập dữ liệu:

Tập huấn luyện:

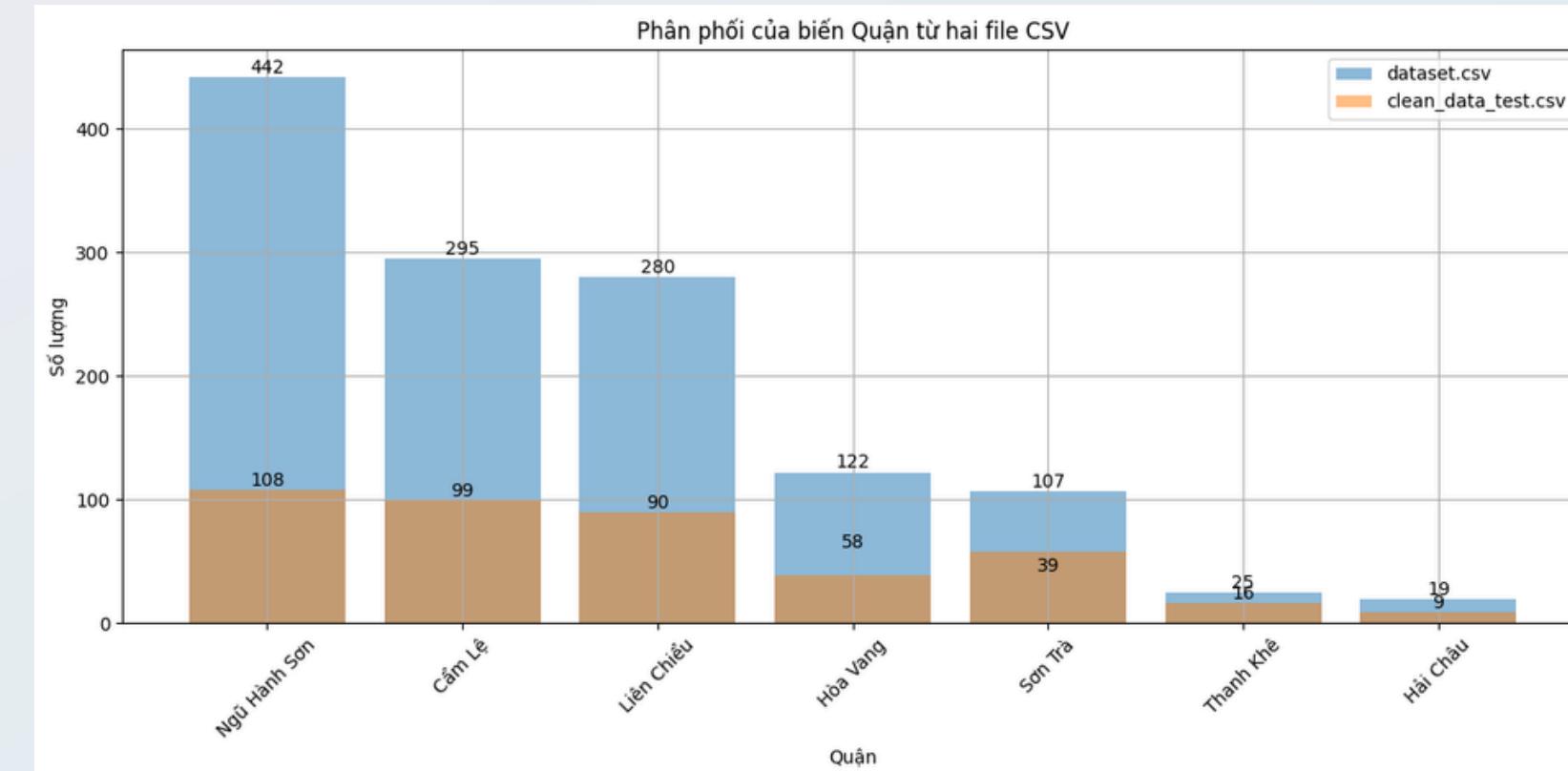
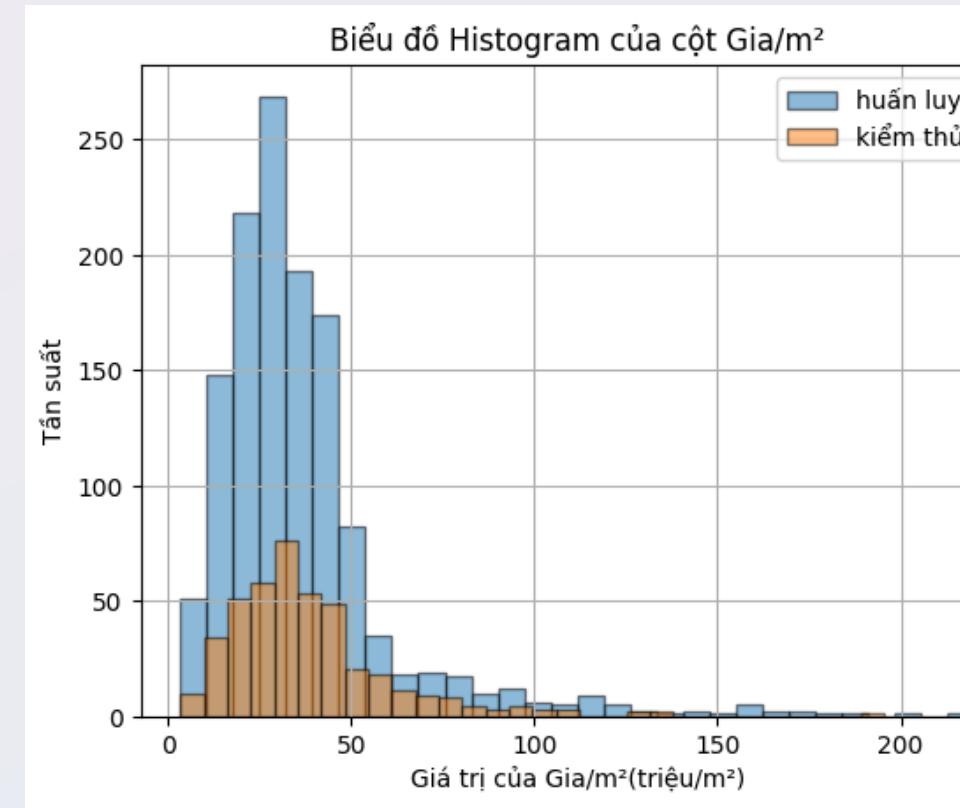
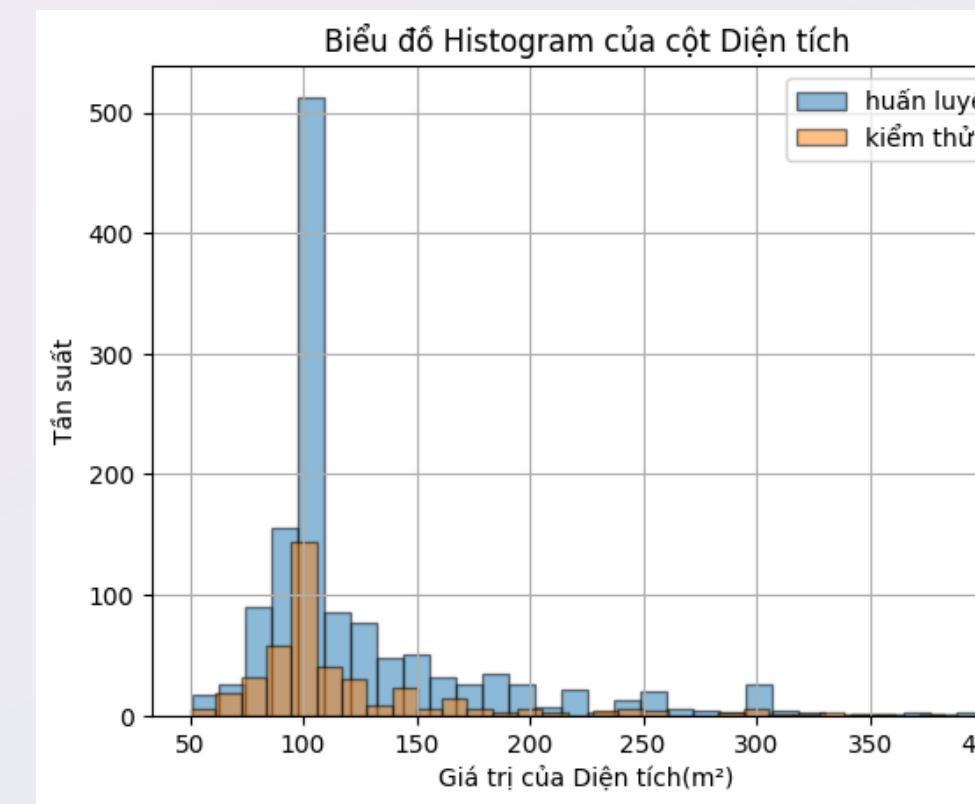
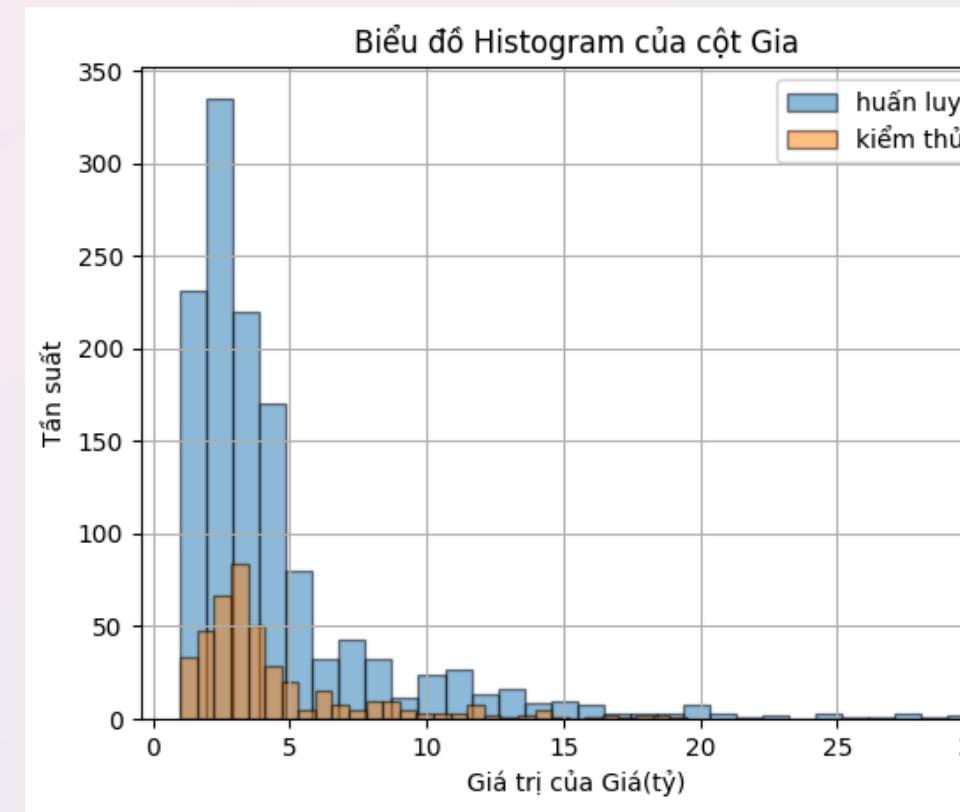


Tập kiểm thử:



2.2. Mô Tả Và Trực Quan Hóa Dữ Liệu

So sánh giữa tập huấn luyện và tập kiểm thử:



- **nhận xét:** Phân bố dữ liệu của biến mục tiêu và biến quan trọng trên cả tập huấn luyện và tập kiểm thử đều khá đồng đều.

3

Trích Xuất Đặc Trưng

3. Trích Xuất Đặc Trưng

StandardScaler

- Là một kỹ thuật được sử dụng trong việc chuẩn hóa dữ liệu trước khi đưa vào các thuật toán học máy
- Các bước thực hiện:

B1: Tính toán giá trị trung bình của từng thuộc tính (cột) trong dữ liệu đầu vào

B2: Tính toán độ lệch chuẩn của mỗi thuộc tính

B3: Chuẩn hóa mỗi thuộc tính của dữ liệu đầu vào bằng công thức

$$Z = \frac{(X - \text{mean})}{\text{std}}$$

Trong đó:

- Z là giá trị chuẩn hóa.
- X là giá trị của một thuộc tính bất kỳ.
- mean là giá trị trung bình của thuộc tính đó.
- std là độ lệch chuẩn của thuộc tính đó.

3. Trích Xuất Đặc Trưng

Kết Quả

Trước khi mã hóa

Gia	Dien tich	Gia/m ²	Chieu ngang	Chieu dai	Huong dat	Loai hinh dat	Quan
1.59	70	22.71	5.1	14	Tây Bắc	thổ cư	Liên Chiểu
2.96	105	28.19	5	21	Tây Bắc	nền dự án	Ngũ Hành Sơn
2.85	111	25.68	6	19	Đông Nam	thổ cư	Ngũ Hành Sơn
2.15	168	12.8	7	24	Nam	thổ cư	Liên Chiểu
10	100	100	5	20	Đông Nam	thổ cư	Ngũ Hành Sơn

Sau khi mã hóa

Gia	Dien tich	Gia/m ²	Chieu ngang	Chieu dai	Huong dat	Loai hinh dat	Quan
1.59	-0.99	-0.59	-0.44	-1.51	-0.32	0.55	-0.68
2.96	-0.30	-0.36	-0.49	0.24	-0.32	-1.80	0.01
2.85	-0.18	-0.47	-0.001	-0.26	0.47	0.55	0.01
2.15	0.93	-1.02	0.49	0.99	-0.92	0.55	-0.68
10	-0.40	2.75	-0.49	-0.01	0.47	0.55	0.01

3. Trích Xuất Đặc Trưng

Target Encoding:

- Là một phương pháp mã hóa biến categorical dựa trên giá trị trung bình của biến mục tiêu (target variable) tương ứng với từng giá trị trong biến categorical. Phương pháp này có thể cải thiện hiệu suất mô hình trong trường hợp biến categorical có mối quan hệ mạnh với biến mục tiêu

```
categorical_cols = ['Huong dat', 'Loai hinh dat', 'Quan']
encoder = TargetEncoder(cols=categorical_cols)
encoded_data = encoder.fit_transform(frame[categorical_cols], frame[target_col])
```

3. Trích Xuất Đặc Trưng

Làm sạch dữ liệu:

Giá:

- Loại bỏ chuỗi tỷ và triệu, giá trị có thỏa thuận.
- Thay đổi dấu phẩy thành dấu chấm.
- Chuyển đổi giá trị của triệu sang giá trị tỷ.
- Lấy giá trị trước dấu gạch ngang và lấy từ 1 đến 30 tỷ.

Địa chỉ:

- Biến đổi chuỗi thành chữ cái đầu tiên của mỗi từ là in hoa, các chữ cái còn lại in thường.
- Loại bỏ các dữ liệu có kí tự đặc biệt ở đầu chuỗi, xóa khoảng trắng và xóa chuỗi “xem biểu đồ”.

3. Trích Xuất Đặc Trưng

Làm sạch dữ liệu:

Diện tích:

- Loại bỏ các chuỗi m² và m².
- Lấy giá trị bé hơn bằng 400.

Giá trên m²:

- Loại bỏ các chuỗi “triệu/m^{2”}, “triệu/m^{2”}, “đ/m^{2”}.
- Tính toán lại giá trị và làm tròn.

Hướng đất:

- Loại bỏ các chuỗi “Hướng:”.
- Biến đổi chuỗi thành chữ cái đầu tiên của mỗi từ là in hoa, các chữ cái còn lại in thường.

Loại hình đất:

- Loại bỏ dữ liệu của Đất công nghiệp.

3. Trích Xuất Đặc Trưng

Làm sạch dữ liệu:

Chiều ngang:

- Loại bỏ các chuỗi “m”, “Mặt tiền:”.
- Lấy giá trị từ 4 đến 20.
- Làm tròn giá trị.

Chiều dài:

- Loại bỏ các chuỗi “m”.
- Thay các giá trị trống bằng cách lấy diện tích chia cho chiều ngang.
- Lấy giá trị từ 10 đến 40.
- Làm tròn giá trị.

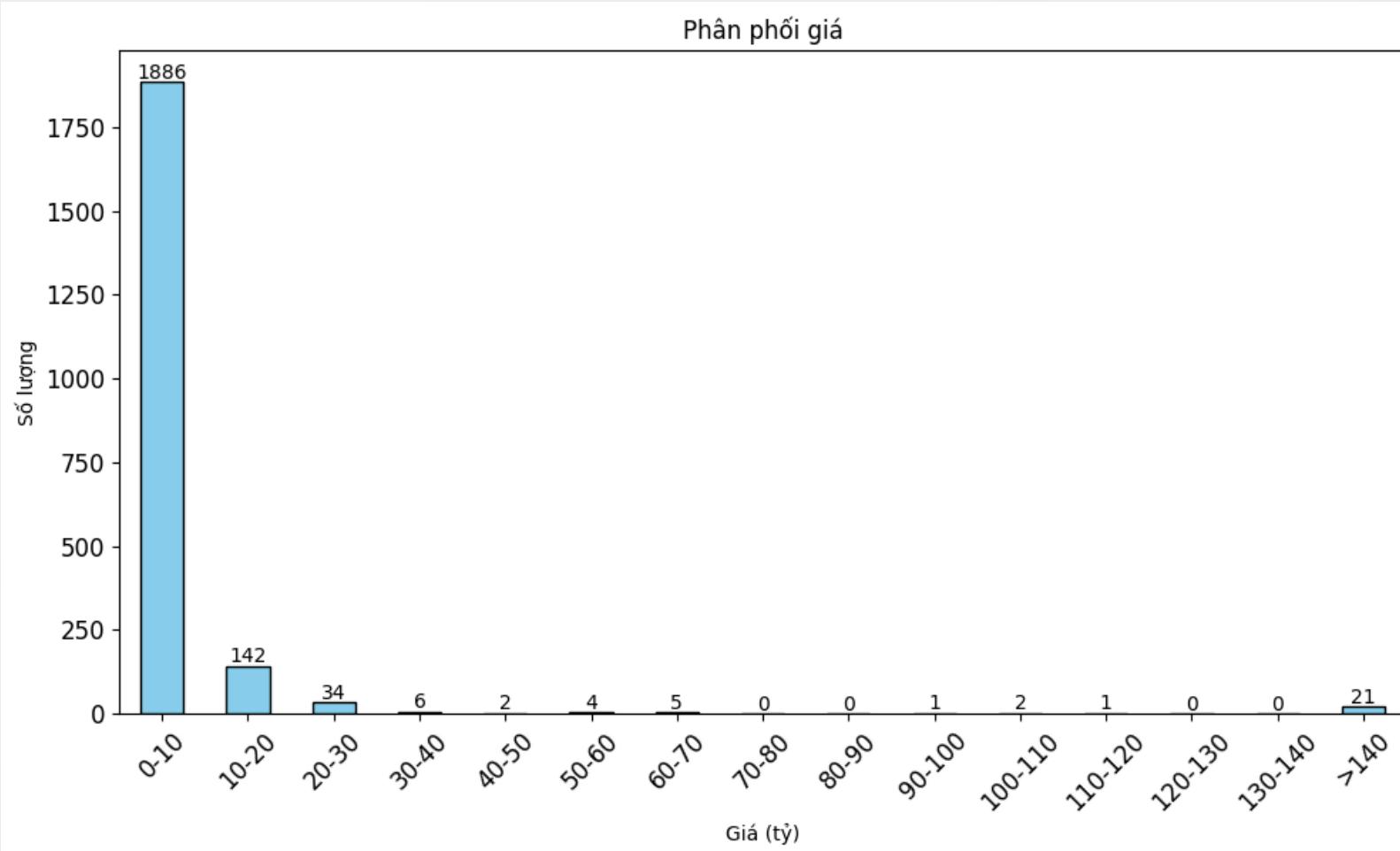
Quận:

- Thêm dữ liệu mới bằng cách lấy dữ liệu từ cột Địa chỉ, lấy giá trị của quận hoặc huyện.
- Nếu có nhiều hơn 1 dấu phẩy thì loại bỏ các giá trị sau dấu phẩy thứ nhất.

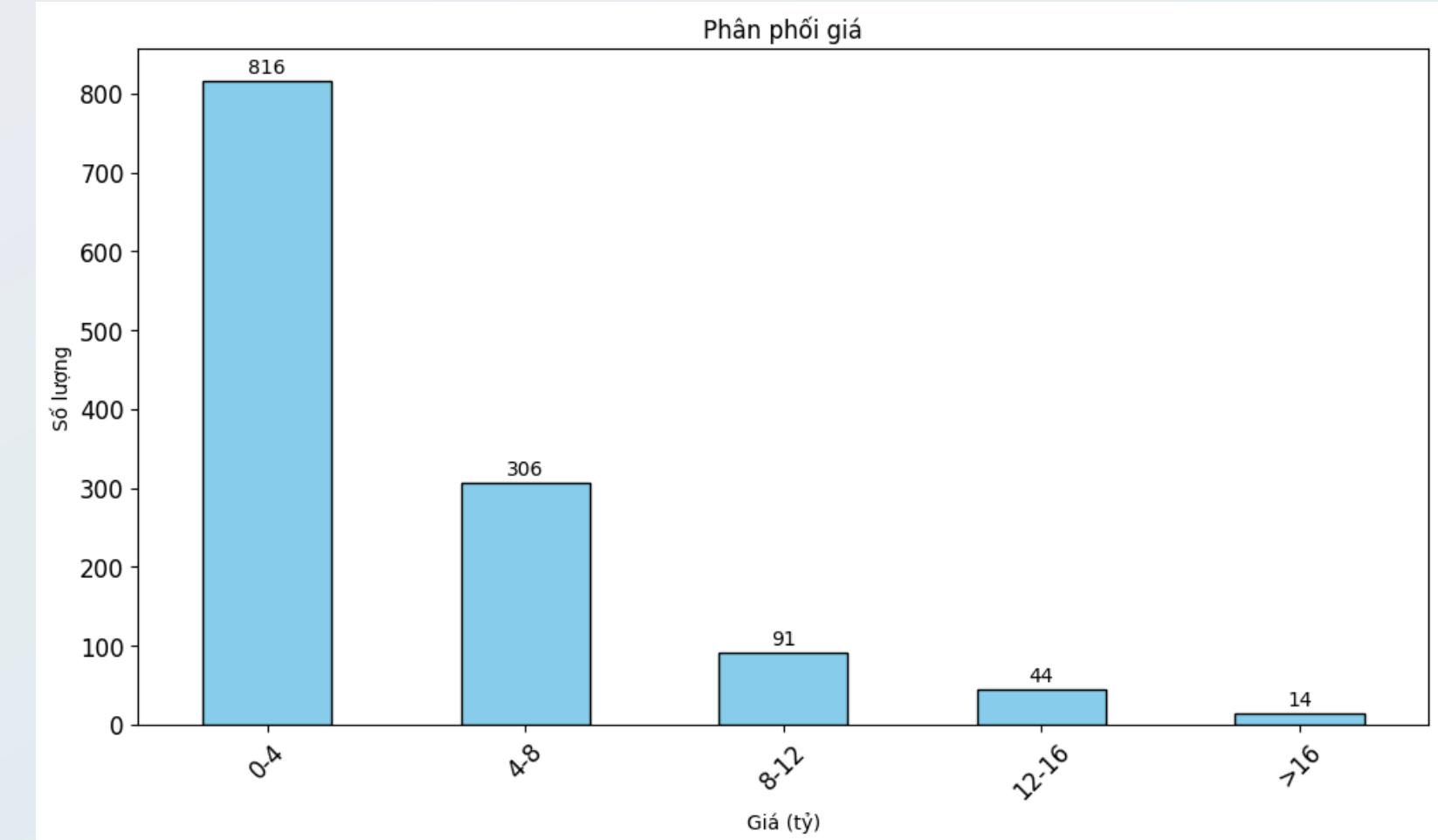
3. Trích Xuất Đặc Trưng

Trực quan dữ liệu trước và sau khi làm sạch:

Trước khi làm sạch



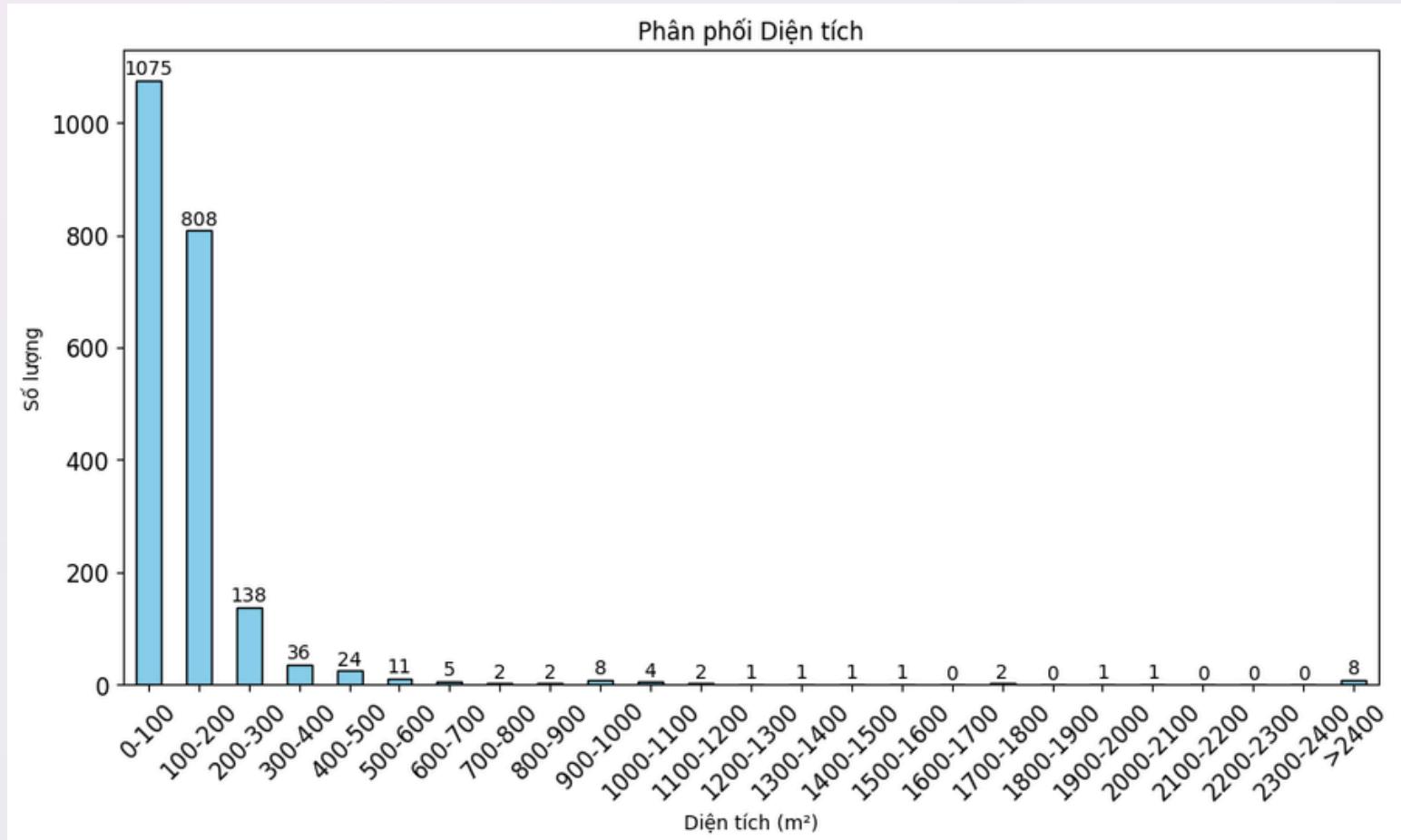
Sau khi làm sạch



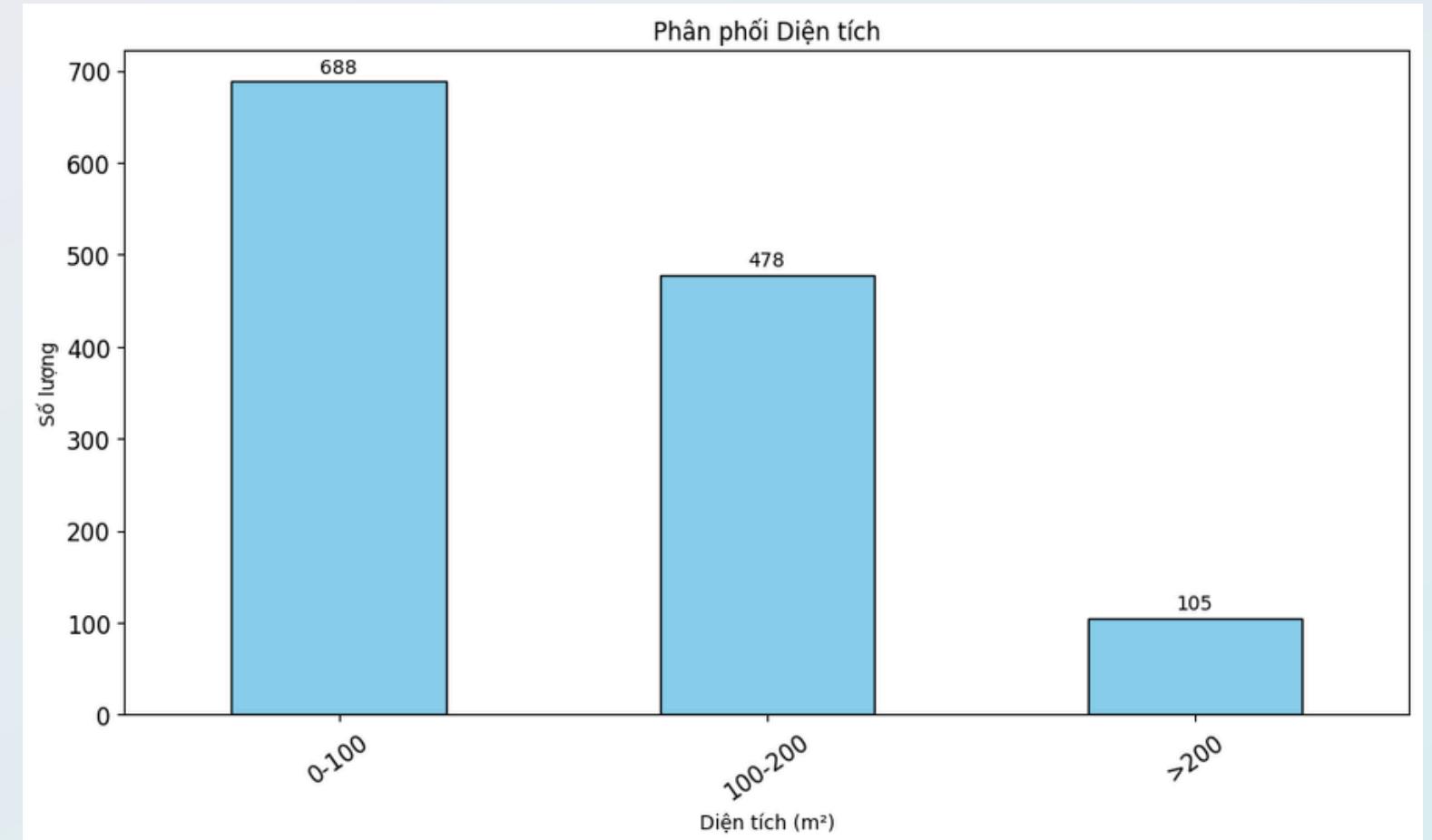
3. Trích Xuất Đặc Trưng

Trực quan dữ liệu trước và sau khi làm sạch:

Trước khi làm sạch



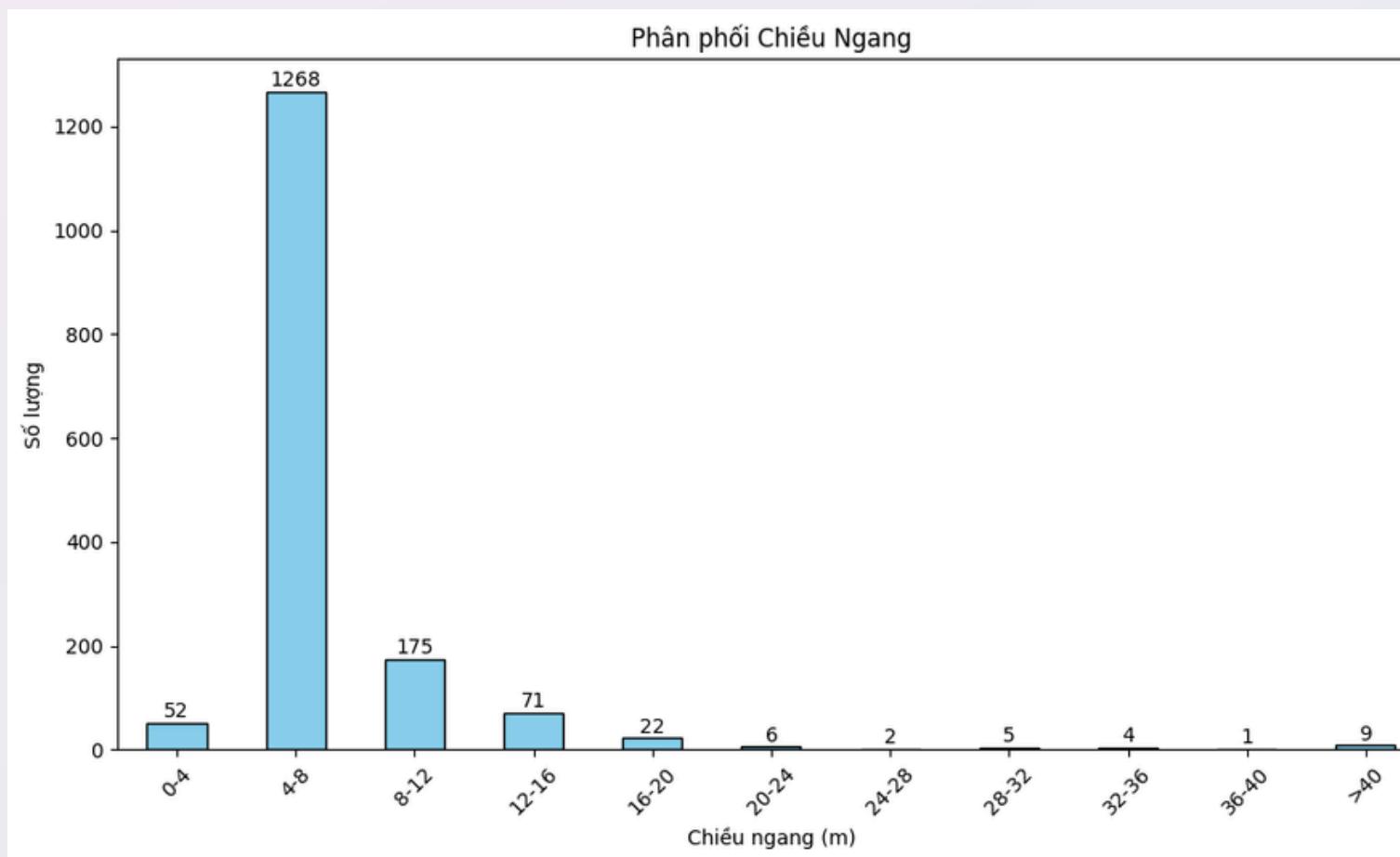
Sau khi làm sạch



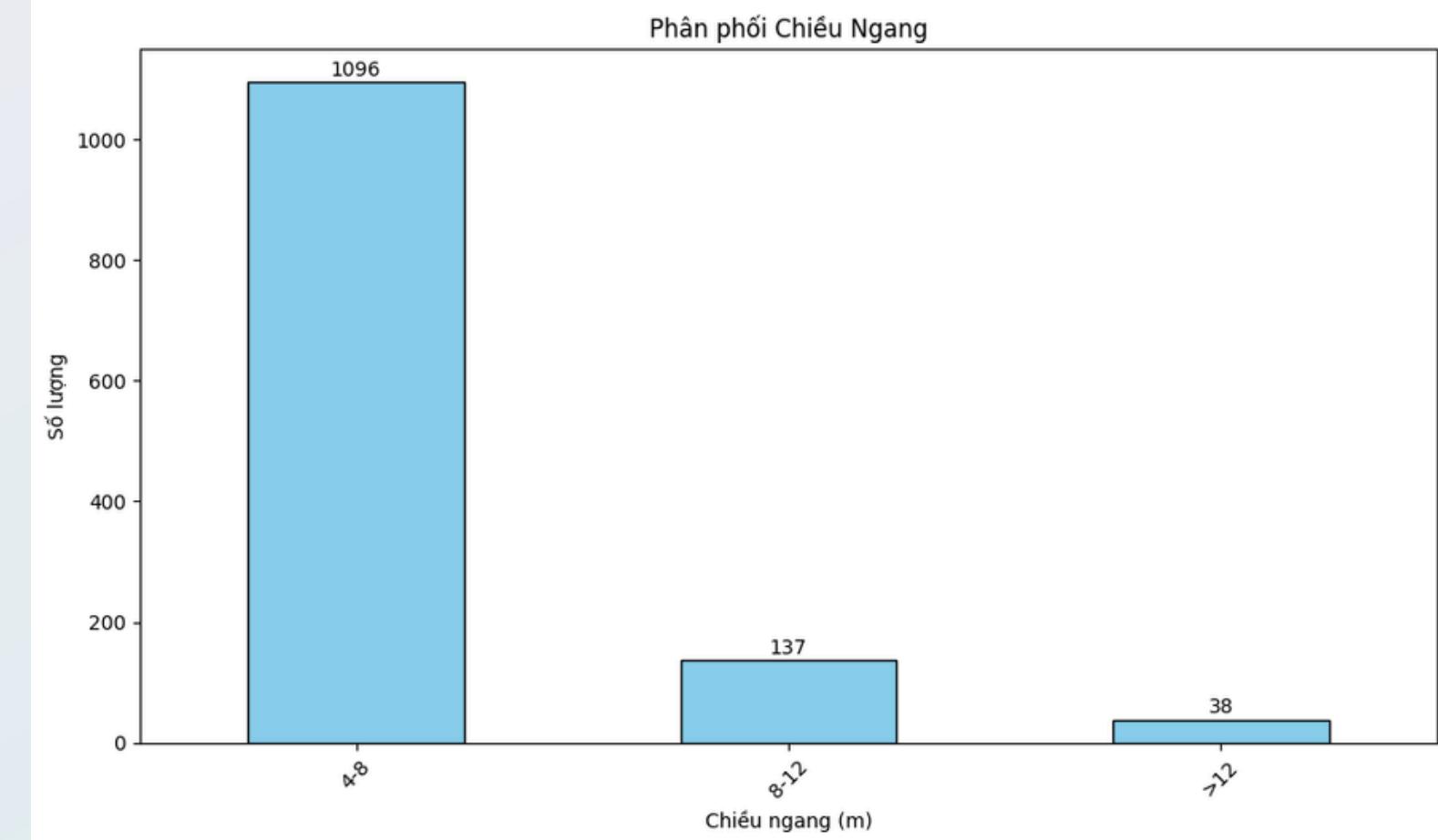
3. Trích Xuất Đặc Trưng

Trực quan dữ liệu trước và sau khi làm sạch:

Trước khi làm sạch



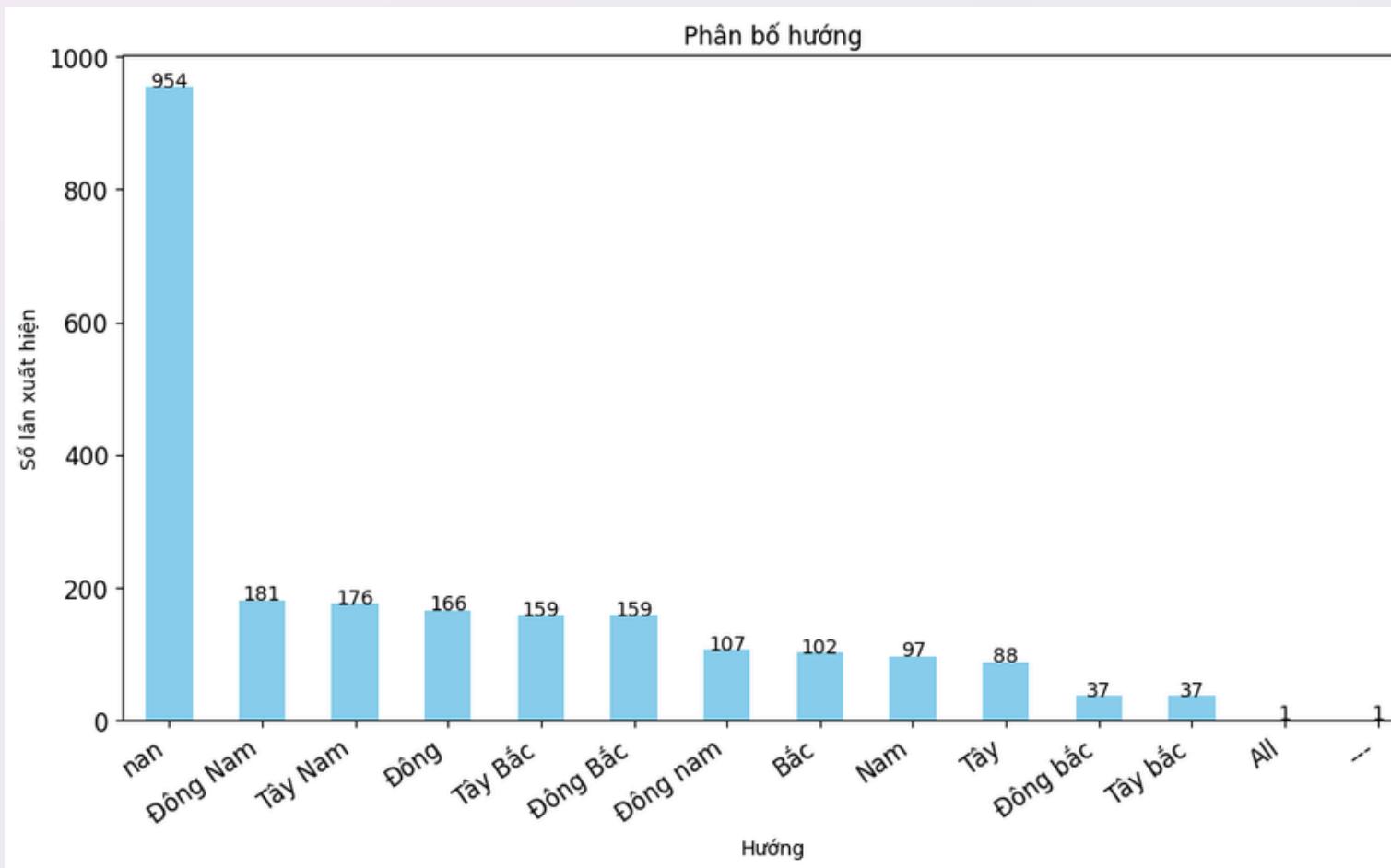
Sau khi làm sạch



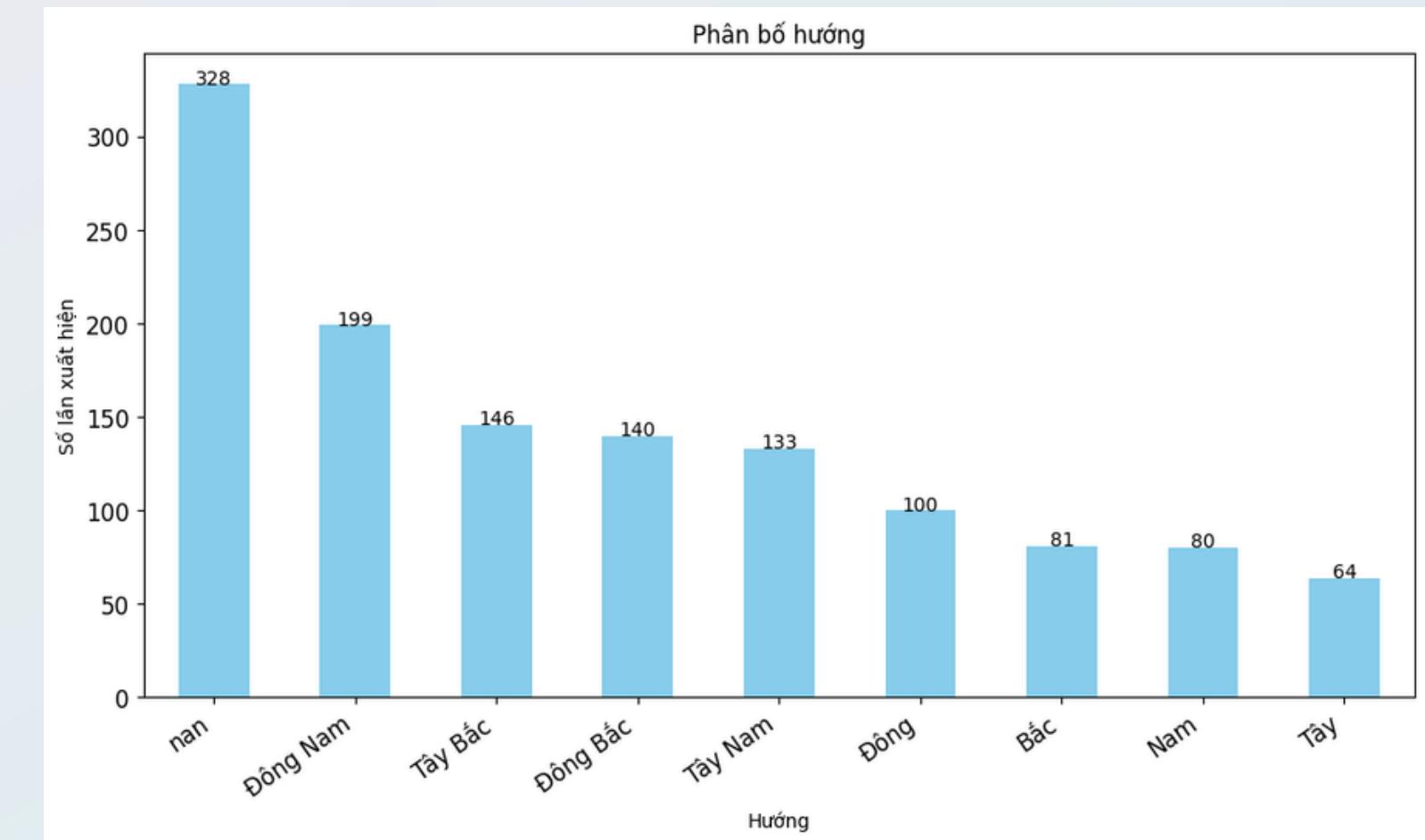
3. Trích Xuất Đặc Trưng

Trực quan dữ liệu trước và sau khi làm sạch:

Trước khi làm sạch



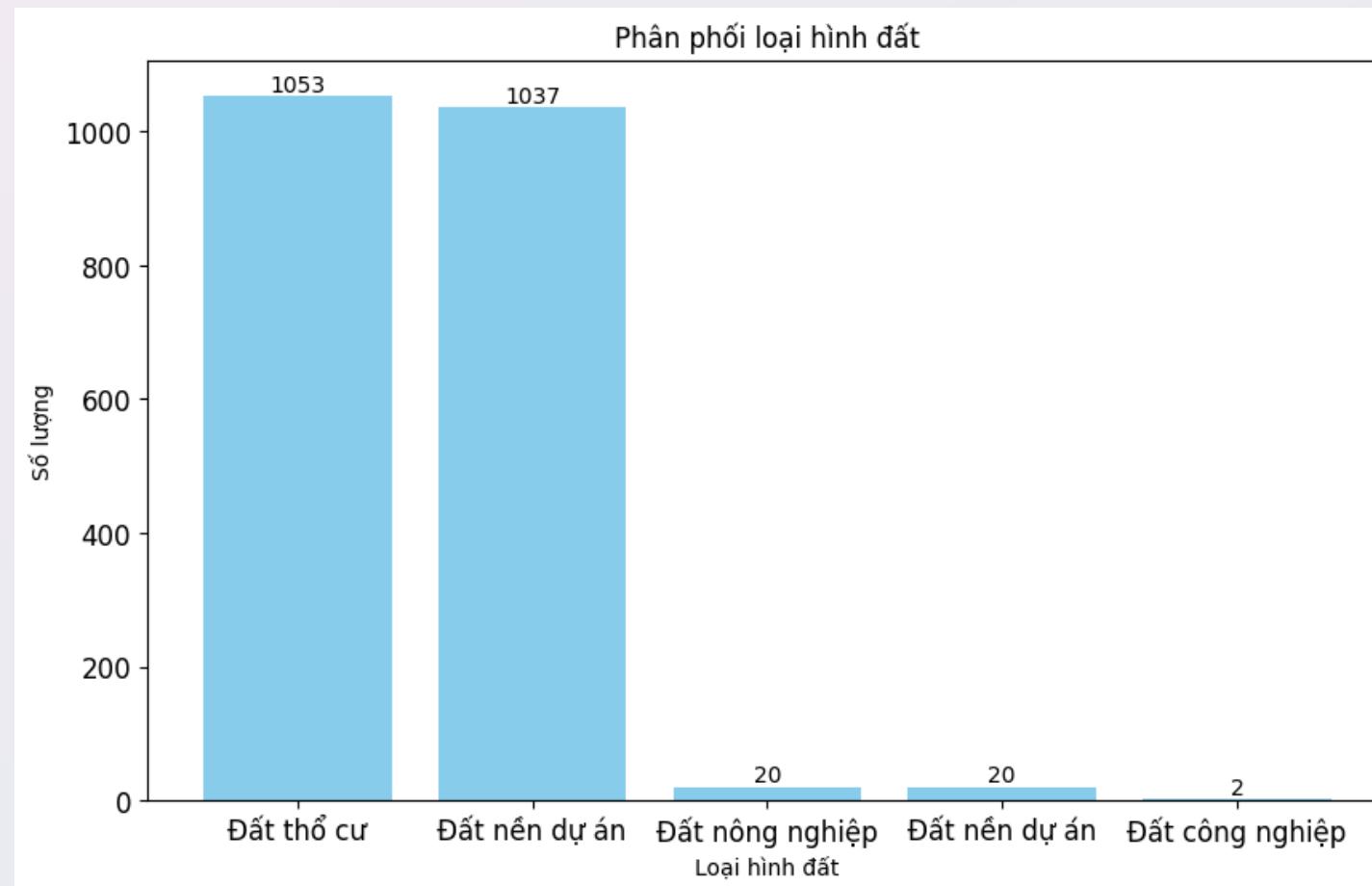
Sau khi làm sạch



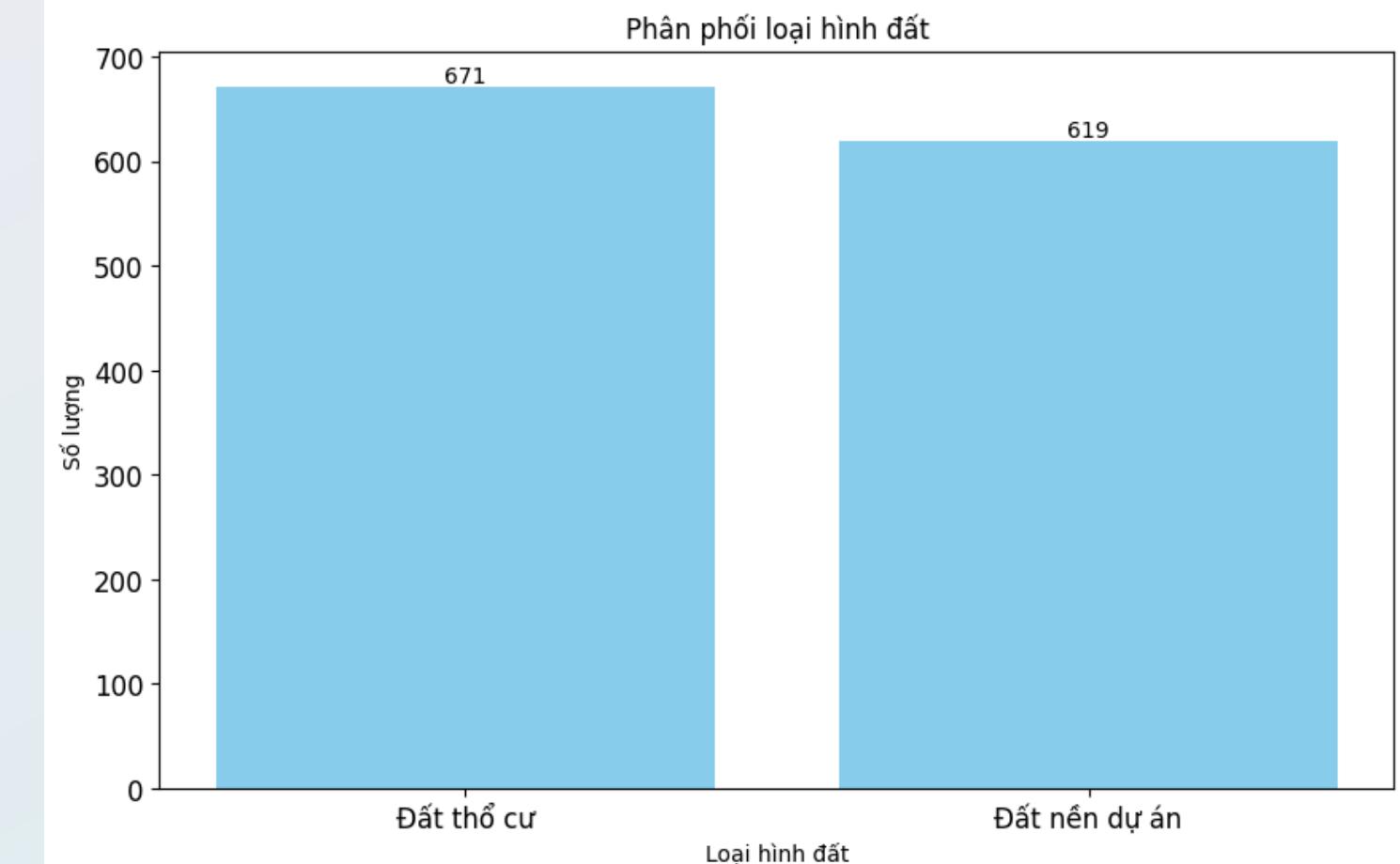
3. Trích Xuất Đặc Trưng

Trực quan dữ liệu trước và sau khi làm sạch:

Trước khi làm sạch



Sau khi làm sạch



4

Mô Hình Hóa Dữ Liệu

4.1 Mô hình Gradient Boosting Regression

Nhập các thư viện

```
from sklearn.ensemble import GradientBoostingRegressor  
from sklearn.model_selection import train_test_split, cross_val_score  
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```

Xác định các đặc trưng(features) và nhãn (target)

```
data = frame  
X = data.drop(columns=['Gia'])  
y = data['Gia']
```

Chia dữ liệu thành tập huấn luyện và kiểm tra

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=365)
```

4.1 Mô hình Gradient Boosting Regression

Khởi tạo mô hình Gradient Boosting Regression

```
reg = GradientBoostingRegressor(random_state=365)
```

Huấn luyện mô hình trên tập huấn luyện

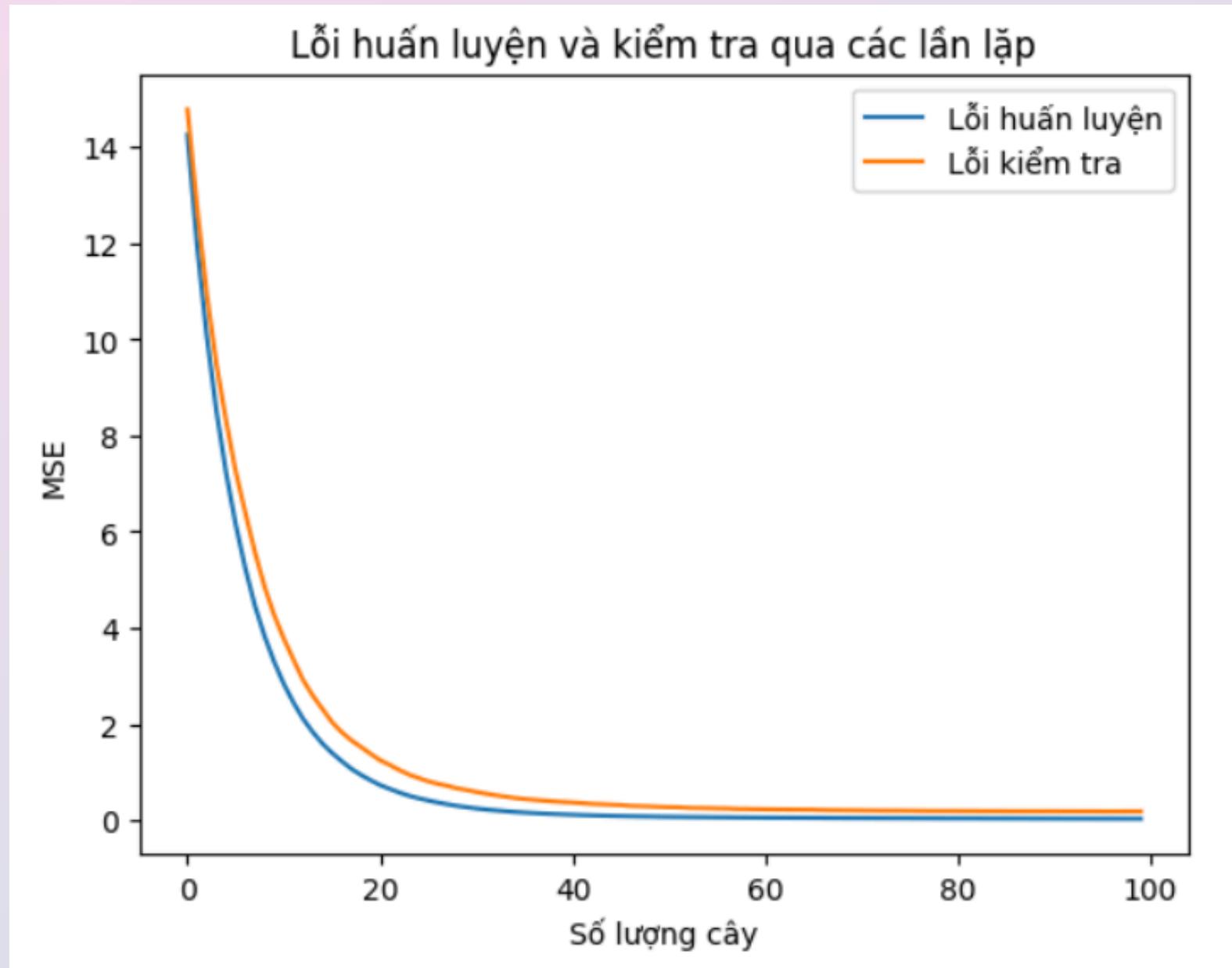
```
reg.fit(X_train, y_train)
```

Dự đoán trên tập kiểm tra

```
y_pred = reg.predict(X_test)
```

4.1 Mô hình Gradient Boosting Regression

Lỗi huấn luyện và kiểm tra qua các lần lặp



Code

```
train_loss = reg.train_score_
test_loss = []
for y_pred_stage in reg.staged_predict(X_test):
    mse = mean_squared_error(y_test, y_pred_stage)
    test_loss.append(mse)
```

4.1 Mô hình Gradient Boosting Regression

So sánh hiệu suất trên tập huấn luyện, tập kiểm tra

	Train	Test
MAE	0.1072	0.1952
MSE	0.0334	0.1717
RMSE	0.1826	0.4143
R ²	0.9980	0.9989

Code

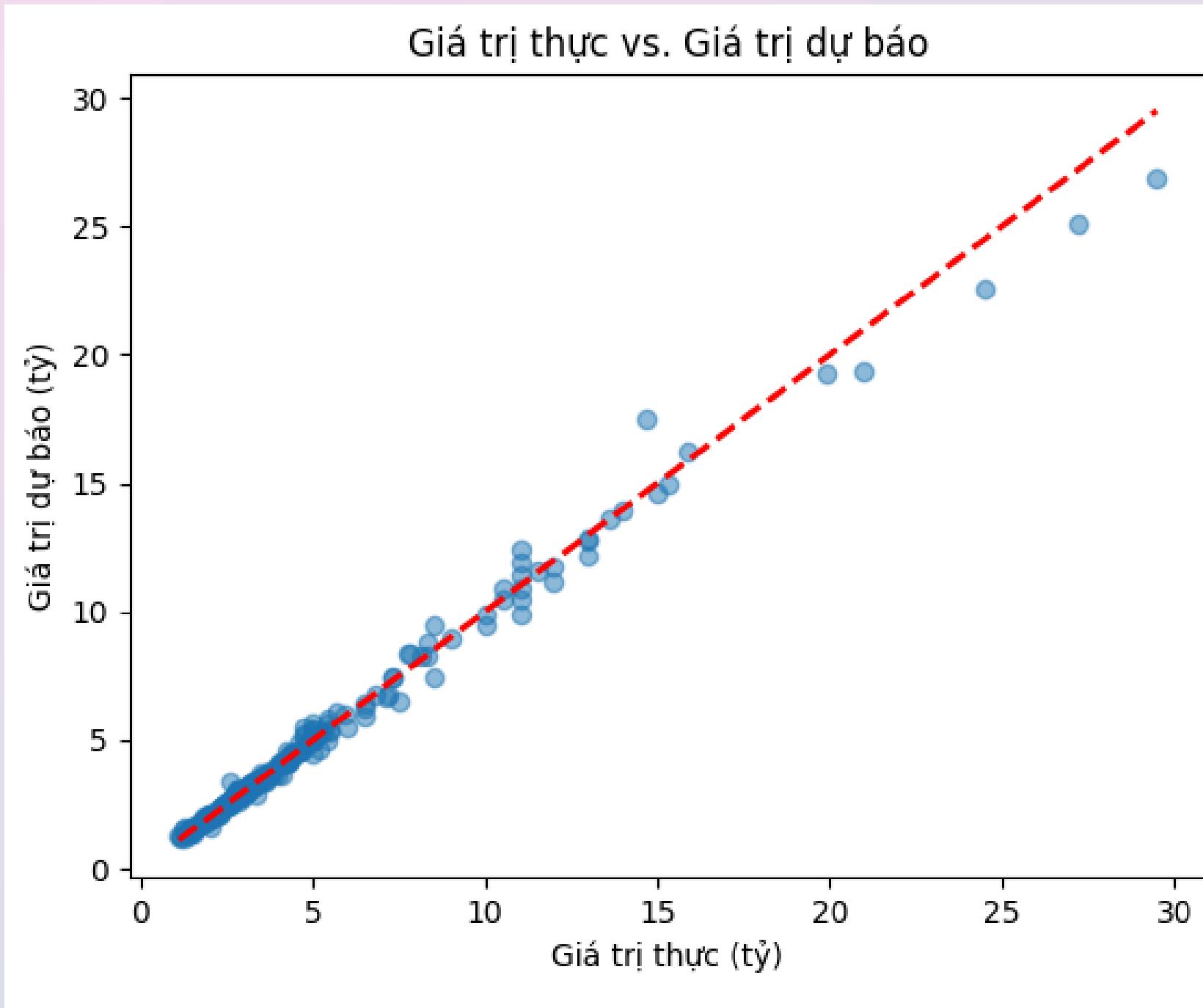
```
y_train_pred = reg.predict(X_train)
train_mae = mean_absolute_error(y_train, y_train_pred)
train_mse = mean_squared_error(y_train, y_train_pred)
train_rmse = np.sqrt(train_mse)
train_r2 = r2_score(y_train, y_train_pred)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

cv_mae = np.abs(cross_val_score(reg, X, y, cv=5, scoring='neg_mean_absolute_error'))
```

4.1 Mô hình Gradient Boosting Regression

Giá dự đoán và giá thực tế



```
perc = np.abs((y_pred - y_test) / y_test) * 100  
  
data = {'Gia du doan': y_pred,  
        'Gia thuc': y_test.values,  
        '% sai lech': perc}  
A = pd.DataFrame(data)  
A['% sai lech'] = A['% sai lech'].round(4)  
B = A.sort_values(by=['% sai lech'])  
B
```

Kết Quả

Gia du doan	Gia thuc	% sai lech
878	1.479976	1.48 0.0016
768	1.349840	1.35 0.0119
344	2.250515	2.25 0.0229
655	2.451150	2.45 0.0470
641	3.497870	3.50 0.0609
...
225	17.520328	14.70 19.1859
368	1.598508	2.00 20.0746
846	1.504245	1.20 25.3538
892	1.588743	1.25 27.0994
1215	3.377514	2.60 29.9044

4.1 Mô hình Gradient Boosting Regression

Thống kê tổng quan về sai lệch giá

```
count      258.000000
mean       3.874496
std        4.501251
min        0.001600
25%        0.860475
50%        2.145600
75%        5.614975
max        29.904400
```

Tính sai lệch giá trung bình

```
(sum((reg.predict(X_test) - y_test)**2)/len(y_test))**0.5
✓ 0.0s
0.41434273919610326
```

4.1 Mô hình Gradient Boosting Regression

Kiểm tra mô hình bằng kỹ thuật multiple test sets

	MAE	MSE	RMSE	R^2
Train	0.1183	0.0442	0.2102	0.9975
Split 1	0.1879	0.1908	0.4368	0.9919
Split 2	0.3002	0.3295	0.574	0.9853
Split 3	0.1915	0.1195	0.3457	0.9859
Split 4	0.1823	0.0998	0.3159	0.9878
Split 5	0.2399	0.3232	0.5685	0.9707

4.1 Mô hình Gradient Boosting Regression

Hiệu chỉnh bằng Early stopping

1. Thiết lập không gian siêu tham số

```
param_grid = {  
    'n_estimators': [100, 500, 1000],  
    'learning_rate': [0.01, 0.05, 0.1],  
    'max_depth': [3, 4, 5],  
    'validation_fraction': [0.1, 0.15, 0.2],  
    'n_iter_no_change': [5, 10, 20],  
    'tol': [1e-4, 1e-3, 1e-2]  
}
```

2. Sử dụng GridSearchCV để tìm kiếm tối ưu hóa siêu tham số

```
grid_search = GridSearchCV(  
    reg,  
    param_grid,  
    cv=5,  
    scoring='neg_mean_absolute_error',  
    verbose=1,  
    n_jobs=-1  
)
```

3. Thực hiện tìm kiếm

```
grid_search.fit(x_train, y_train)
```

4. Lấy bộ siêu tham số tốt nhất

```
best_params = grid_search.best_params_  
print("Bộ siêu tham số tốt nhất:", best_params)
```

Kết quả của bộ siêu tham số

```
Bộ siêu tham số tốt nhất: {'learning_rate': 0.05, 'max_depth': 4, 'n_estimators': 1000, 'n_iter_no_change': 20, 'tol': 0.0001, 'validation_fraction': 0.2}
```

4.1 Mô hình Gradient Boosting Regression

Thiết lập bộ siêu tham số mới trước khi chạy lại mô hình

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2345)
reg_test = GradientBoostingRegressor(
    random_state=2345,
    n_estimators=1000,
    learning_rate=0.05,
    max_depth=4,
    validation_fraction=0.2,
    n_iter_no_change=20,
    tol=0.0001
)
```

Kết quả

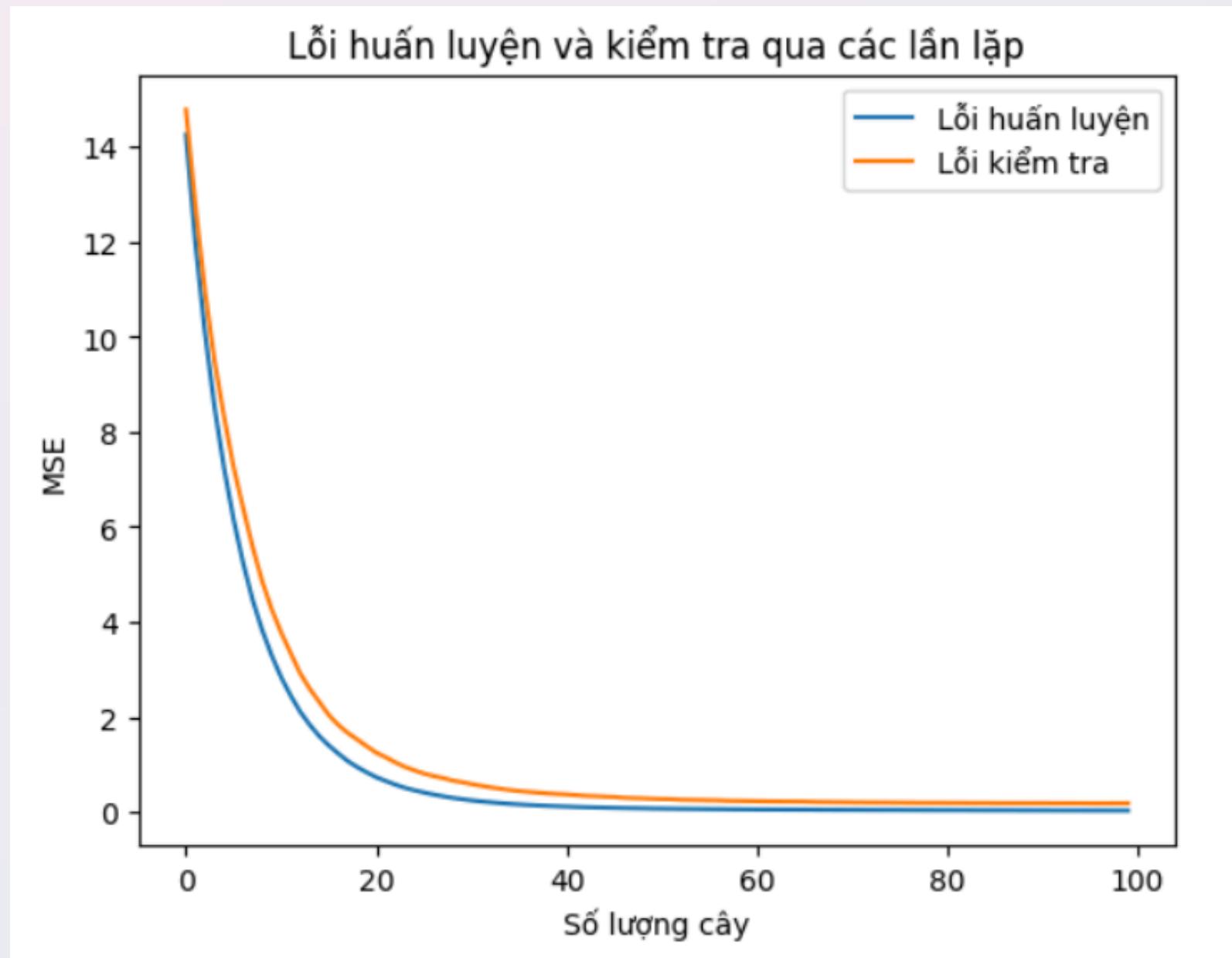
Số vòng lặp trước
khi dừng sớm: 190

	Train	Test
MAE	0.0954	0.1300
MSE	0.1097	0.0874
RMSE	0.3312	0.2956
R^2	0.9940	0.9921

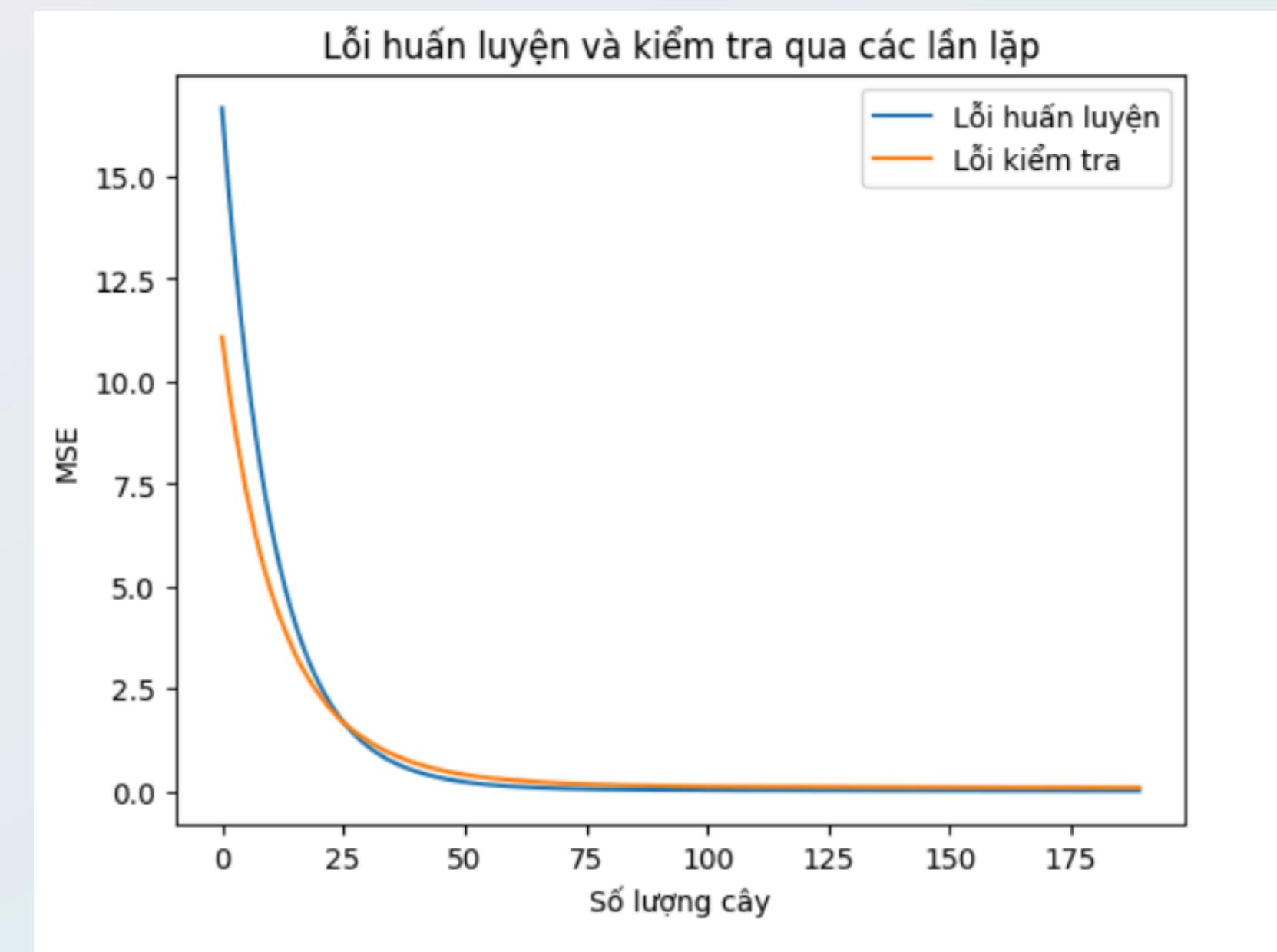
4.1 Mô hình Gradient Boosting Regression

So sánh kết quả trước và sau hiệu chỉnh

Trước khi hiệu chỉnh



Sau khi hiệu chỉnh



4.1 Mô hình Gradient Boosting Regression

So sánh kết quả trước và sau hiệu chỉnh

Trước khi hiệu chỉnh

	Train	Test
MAE	0.1072	0.1952
MSE	0.0334	0.1717
RMSE	0.1826	0.4143
R^2	0.9980	0.9989

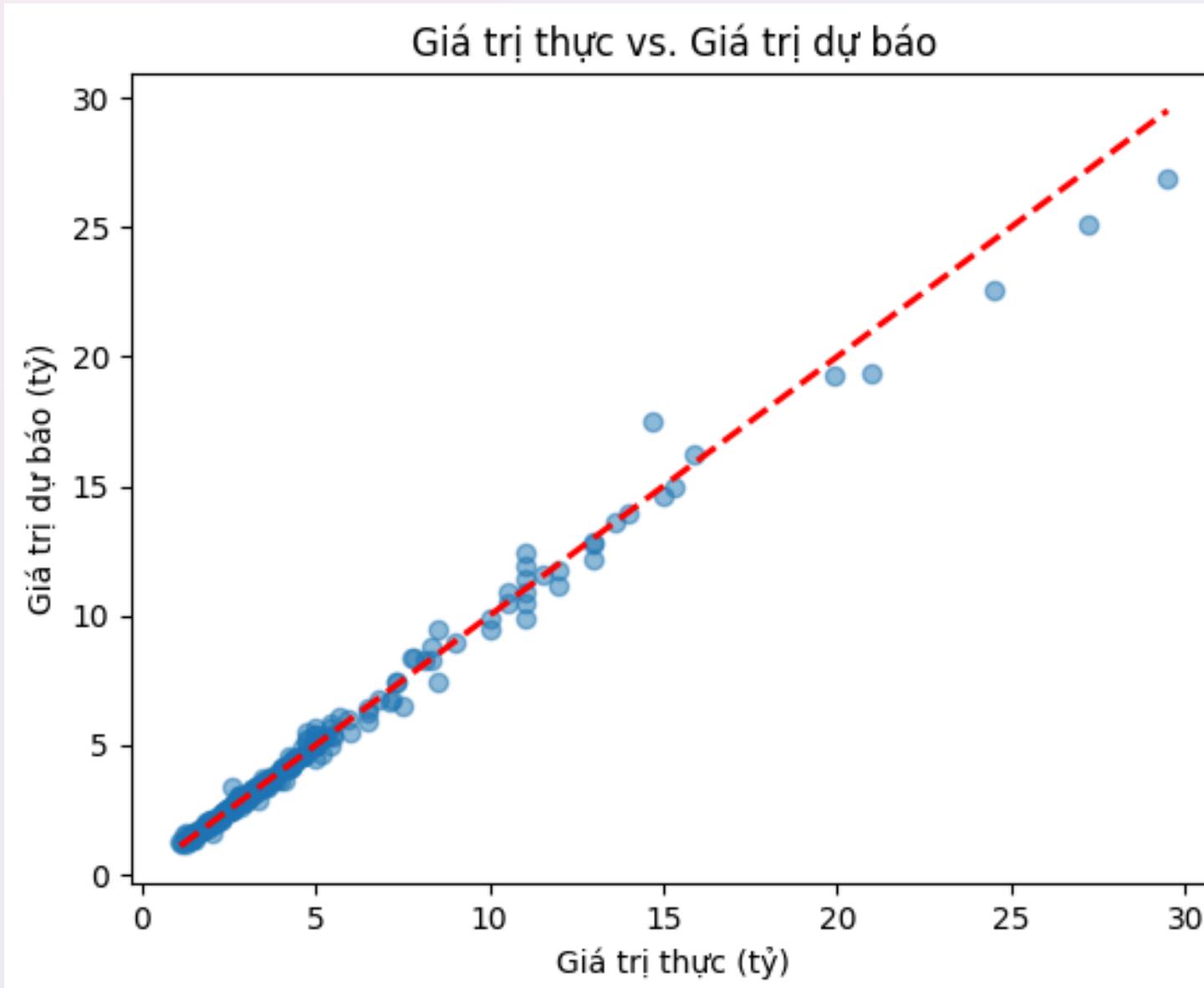
Sau khi hiệu chỉnh

	Train	Test
MAE	0.0954	0.1300
MSE	0.1097	0.0874
RMSE	0.3312	0.2956
R^2	0.9940	0.9921

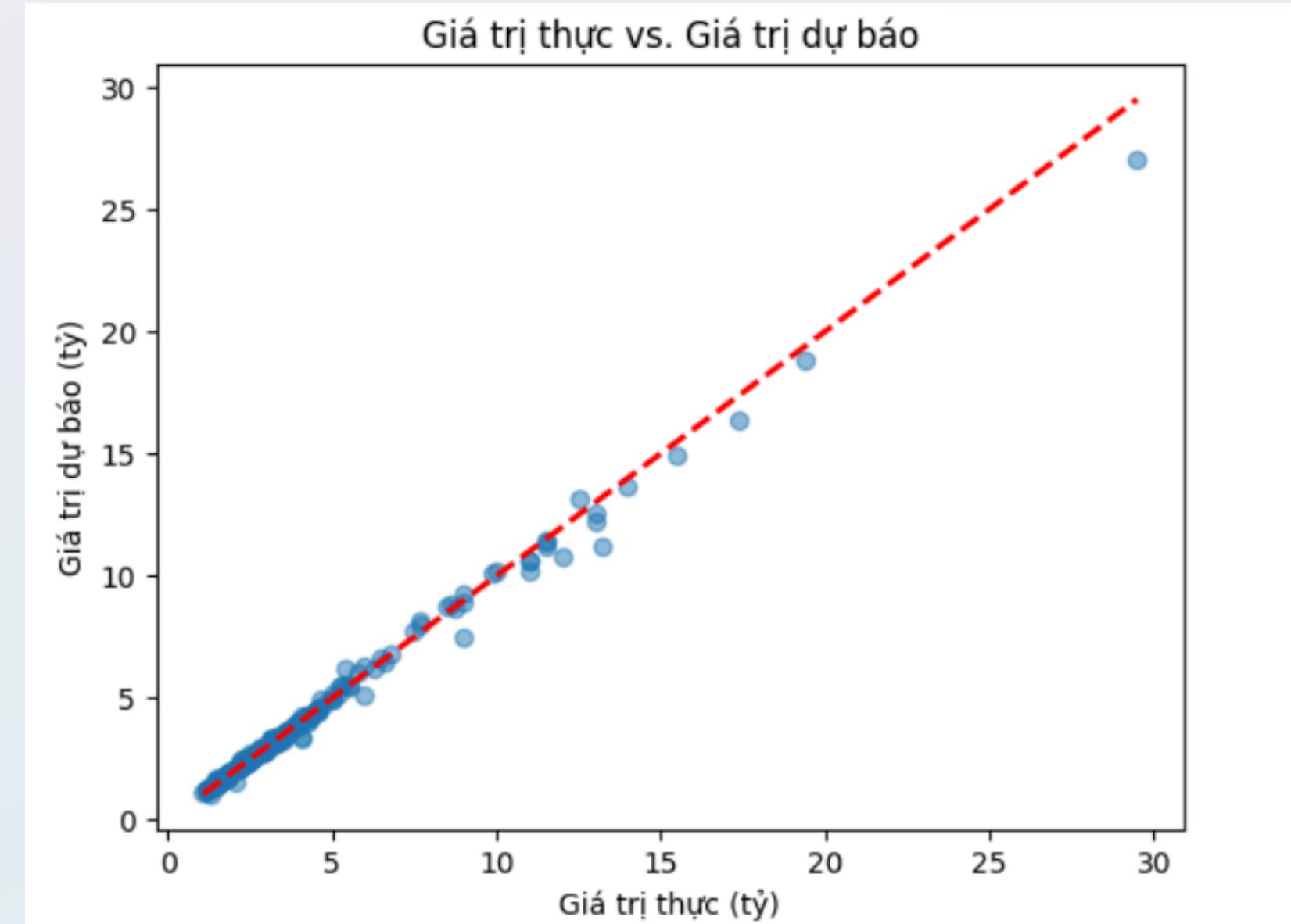
4.1 Mô hình Gradient Boosting Regression

So sánh kết quả trước và sau hiệu chỉnh

Trước khi hiệu chỉnh



Sau khi hiệu chỉnh



4.1 Mô hình Gradient Boosting Regression

So sánh kết quả trước và sau hiệu chỉnh

Trước khi hiệu chỉnh

	Gia du doan	Gia thuc	% sai lech
878	1.479976	1.48	0.0016
768	1.349840	1.35	0.0119
344	2.250515	2.25	0.0229
655	2.451150	2.45	0.0470
641	3.497870	3.50	0.0609
...
225	17.520328	14.70	19.1859
368	1.598508	2.00	20.0746
846	1.504245	1.20	25.3538
892	1.588743	1.25	27.0994
1215	3.377514	2.60	29.9044

Sai lệch giá trung bình

0.41434273919610326

Sau khi hiệu chỉnh

	Gia du doan	Gia thuc	% sai lech
90	3.650806	3.65	0.0221
1146	11.504162	11.50	0.0362
423	1.290542	1.29	0.0420
173	6.803623	6.80	0.0533
479	2.797891	2.80	0.0753
...
518	3.447207	4.10	15.9218
727	7.511253	9.00	16.5416
151	3.356647	4.08	17.7292
851	1.058895	1.30	18.5466
984	1.589819	2.07	23.1972

Sai lệch giá trung bình

0.19287102698237835

4.2 Mô hình Random Forest Regression

Random Forest Regression:

- Hồi quy rừng ngẫu nhiên trong học máy là một kỹ thuật tổng hợp có khả năng thực hiện cả nhiệm vụ hồi quy và phân loại với việc sử dụng nhiều cây quyết định và một kỹ thuật gọi là Bootstrap và Aggregation, thường được gọi là đóng gói.
- Ý tưởng cơ bản đằng sau điều này là kết hợp nhiều cây quyết định trong việc xác định kết quả cuối cùng thay vì dựa vào cây quyết định riêng lẻ.

4.2 Mô hình Random Forest Regression

Random Forest Regression:

Các bước thực hiện:

Bước-1: Nhập thư viện

Ở đây chúng tôi đang nhập tất cả các thư viện cần thiết cần thiết.

```
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
import pandas as pd
|
```

4.2 Mô hình Random Forest Regression

Random Forest Regression:

Các bước thực hiện:

Bước 2: Nhập tập dữ liệu

```
data = frame
```

	Gia	Dien tích	Gia/m ²	Chieu ngang	Chieu dai	Huong dat_Nam	Huong dat_Tây	Huong dat_Tây Bắc	Huong dat_Tây Nam	Huong dat_Dòng	Huong dat_Dòng Bắc	Huong dat_Dòng Nam
0	3.15	-0.456774	-0.195102	-0.519699	-0.038198	0.0	0.0	1.0	0.0	0.0	0.0	0.0
1	2.50	-0.842025	-0.189120	-0.082630	-1.643005	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2	2.10	0.001859	-0.781291	-0.869355	2.903947	0.0	0.0	0.0	0.0	0.0	1.0	0.0
3	5.90	1.744663	-0.381725	1.665649	0.496737	0.0	0.0	0.0	1.0	0.0	0.0	0.0
4	3.50	-0.676918	0.134679	-0.519699	-0.706867	0.0	0.0	0.0	0.0	0.0	1.0	0.0

4.2 Mô hình Random Forest Regression

Random Forest Regression:

Các bước thực hiện:

Bước 3: Chuẩn bị dữ liệu

- Trích xuất các tính năng: Nó trích xuất các tính năng từ DataFrame và lưu trữ chúng trong một biến có tên X
- Trích xuất biến mục tiêu: Nó trích xuất biến đích từ DataFrame và lưu trữ nó trong một biến có tên y

```
x = data.drop(columns=['Gia'])
y = data['Gia']
```

4.2 Mô hình Random Forest Regression

Random Forest Regression:

Các bước thực hiện:

Bước 4: Mô hình hồi quy rừng ngẫu nhiên

- Mã xử lý dữ liệu phân loại bằng cách mã hóa nó bằng số, kết hợp dữ liệu được xử lý với dữ liệu số và đào tạo mô hình Hồi quy rừng ngẫu nhiên bằng cách sử dụng dữ liệu đã chuẩn bị

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=365)
reg = RandomForestRegressor(random_state=365)
reg.fit(x_train, y_train)
y_pred = reg.predict(x_test)
```

4.2 Mô hình Random Forest Regression

Random Forest Regression:

Các bước thực hiện:

Bước-5: Đưa ra dự đoán và đánh giá

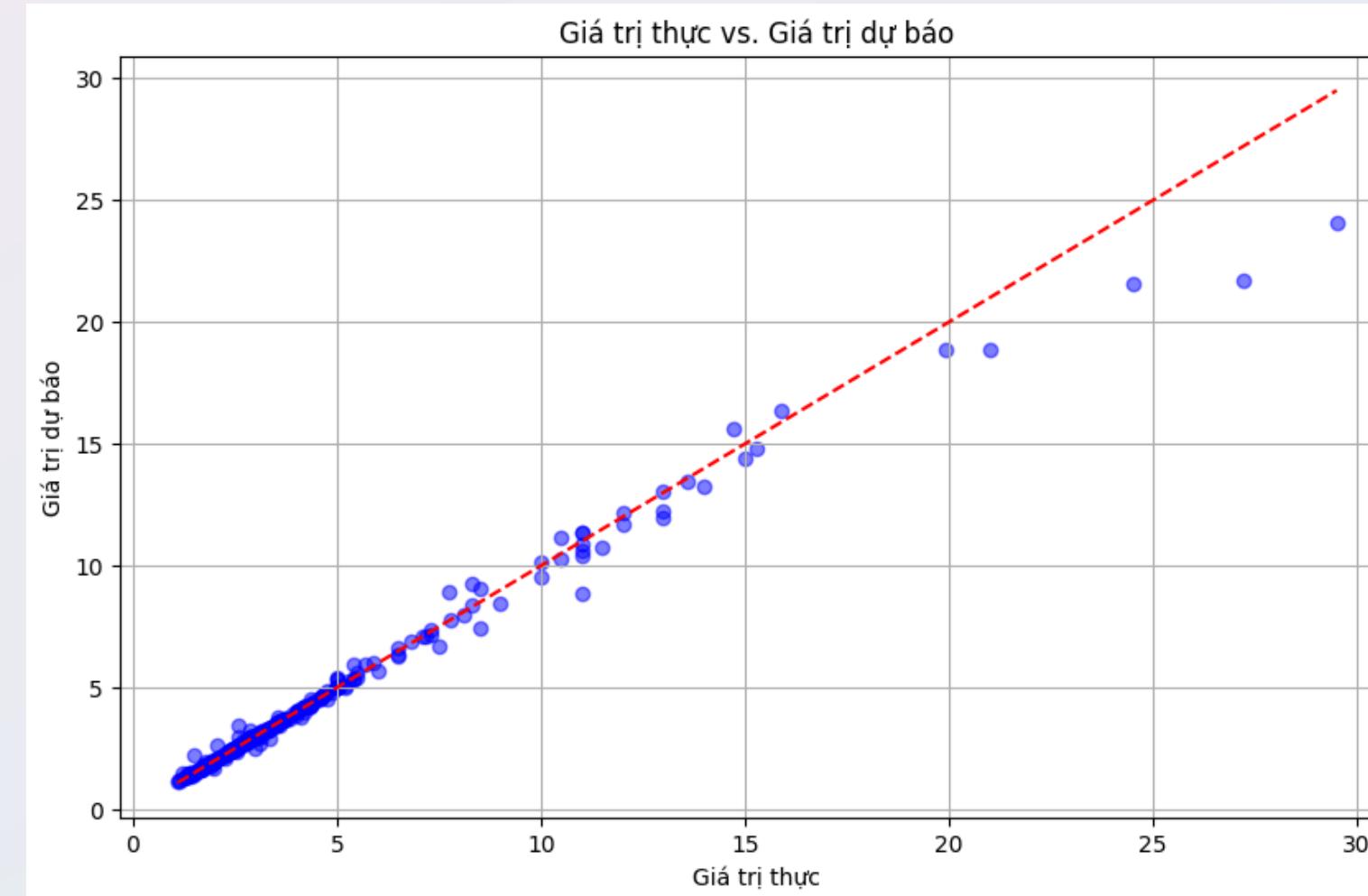
Mã đánh giá mô hình hồi quy rừng ngẫu nhiên được đào tạo:

- điểm out-of-bag (OOB), ước tính hiệu suất khái quát hóa của mô hình.
- Đưa ra dự đoán bằng cách sử dụng mô hình đã đào tạo và lưu trữ chúng trong mảng 'dự đoán'.
- Đánh giá hiệu suất của mô hình bằng cách sử dụng các chỉ số Lỗi bình phương trung bình (MSE) và bình phương R (R2).

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

4.2 Mô hình Random Forest Regression

Biểu đồ:



4.2 Mô hình Random Forest Regression

Kết quả:

Nhận xét:

	Huấn luyện	Kiểm thử
MAE	0.0777	0.1952
MSE	0.0612	0.3647
RMSE	0.2475	0.6039
R^2	0.9964	0.9786

- MAE của tập huấn luyện là 0.0777, thấp hơn đáng kể so với MAE của tập kiểm tra là 0.1952
- MSE của tập huấn luyện là 0.0612, rất thấp so với MSE của tập kiểm tra là 0.3647.
- Giá trị R^2 của tập huấn luyện là 0.9964, rất gần với giá trị hoàn hảo là 1. Trong khi đó, R^2 của tập kiểm tra là 0.9786, vẫn là một giá trị cao nhưng thấp hơn so với tập huấn luyện.

4.3 So Sánh Hai Mô Hình Hồi Quy

	Gradient Boosting	Random Forest
MAE	0.1098	0.0777
MSE	0.0328	0.0612
RMSE	0.1812	0.2475
R^2	0.9981	0.9964

Nhận xét:

- Giá trị MAE của Gradient Boosting cao hơn Random Forest.
- Giá trị MSE của Gradient Boosting thấp gần gấp đôi Random Forest.
- Giá trị RMSE của Gradient Boosting thấp hơn Random Forest.
- Giá trị R^2 của Gradient Boosting cao hơn Random Forest. Nên sẽ chọn mô hình Gradient Boosting

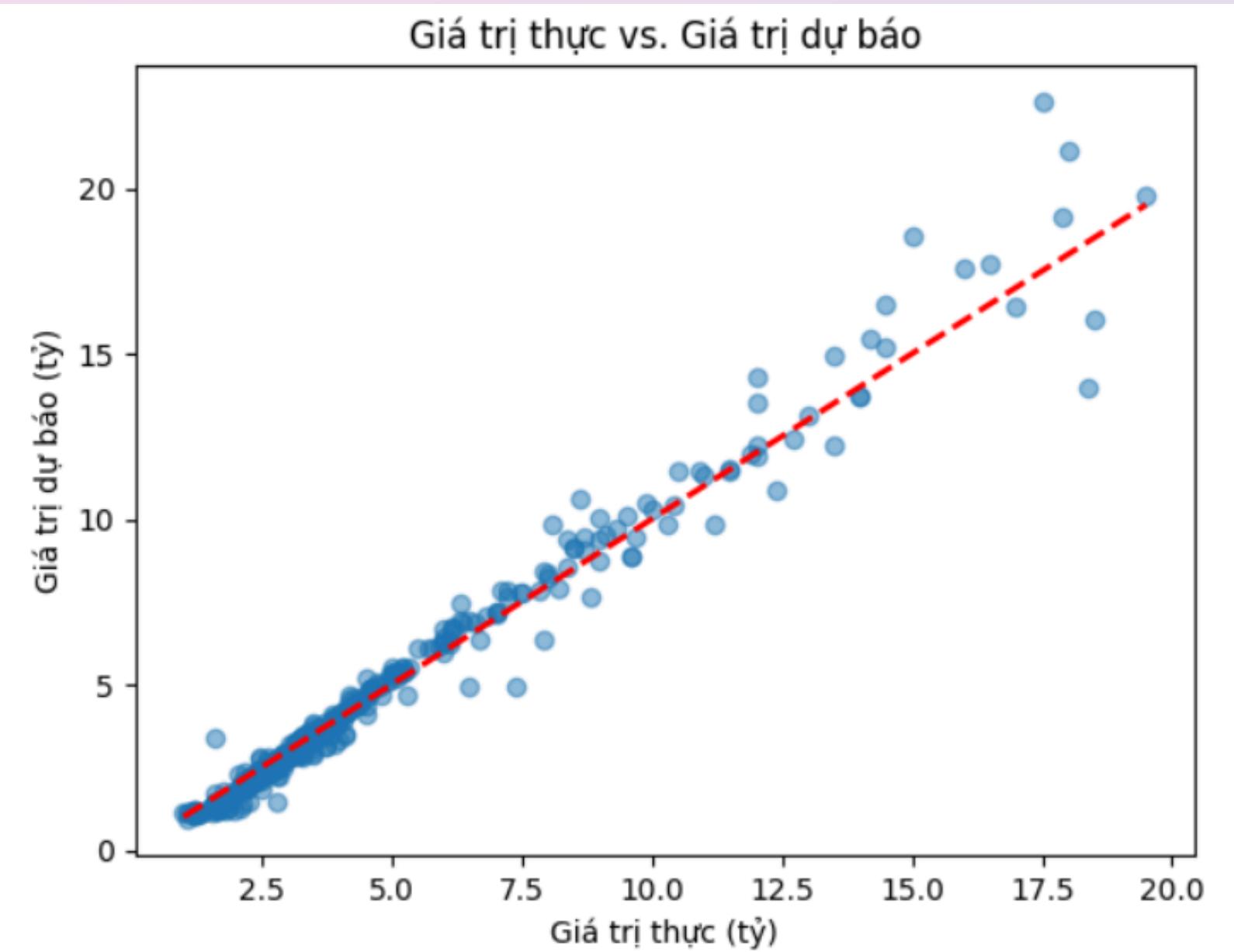
4.4 Kiểm tra mô hình Gradient Boosting Regression

Kiểm tra trên tập dữ liệu test	
MAE	0.3478
MSE	0.2917
RMSE	0,6259
R^2	0.966

Nhận xét:

- Giá trị R^2 giảm xuống
- Các chỉ số còn lại đều tăng lên, cho thấy mô hình đã dự đoán sai lệch một vài giá trị trên tập dữ liệu mới hoàn toàn

4.4 Kiểm tra mô hình Gradient Boosting Regression



	Gia du doan	Gia thuc	% sai lech
18	11.497606	11.50	0.0208
181	7.847283	7.85	0.0346
35	10.404360	10.40	0.0419
43	3.402039	3.40	0.0600
9	2.948058	2.95	0.0658
...
167	1.436509	2.25	36.1552
106	1.188117	1.95	39.0709
16	1.240495	2.09	40.6462
245	1.434340	2.80	48.7736
267	3.407458	1.59	114.3056
419 rows × 3 columns			

count	419.000000
mean	8.454273
std	9.045015
min	0.020800
25%	3.028600
50%	6.233300
75%	11.340650
max	114.305600

Giá trị sai lệch trung bình: 0.6258750122433303

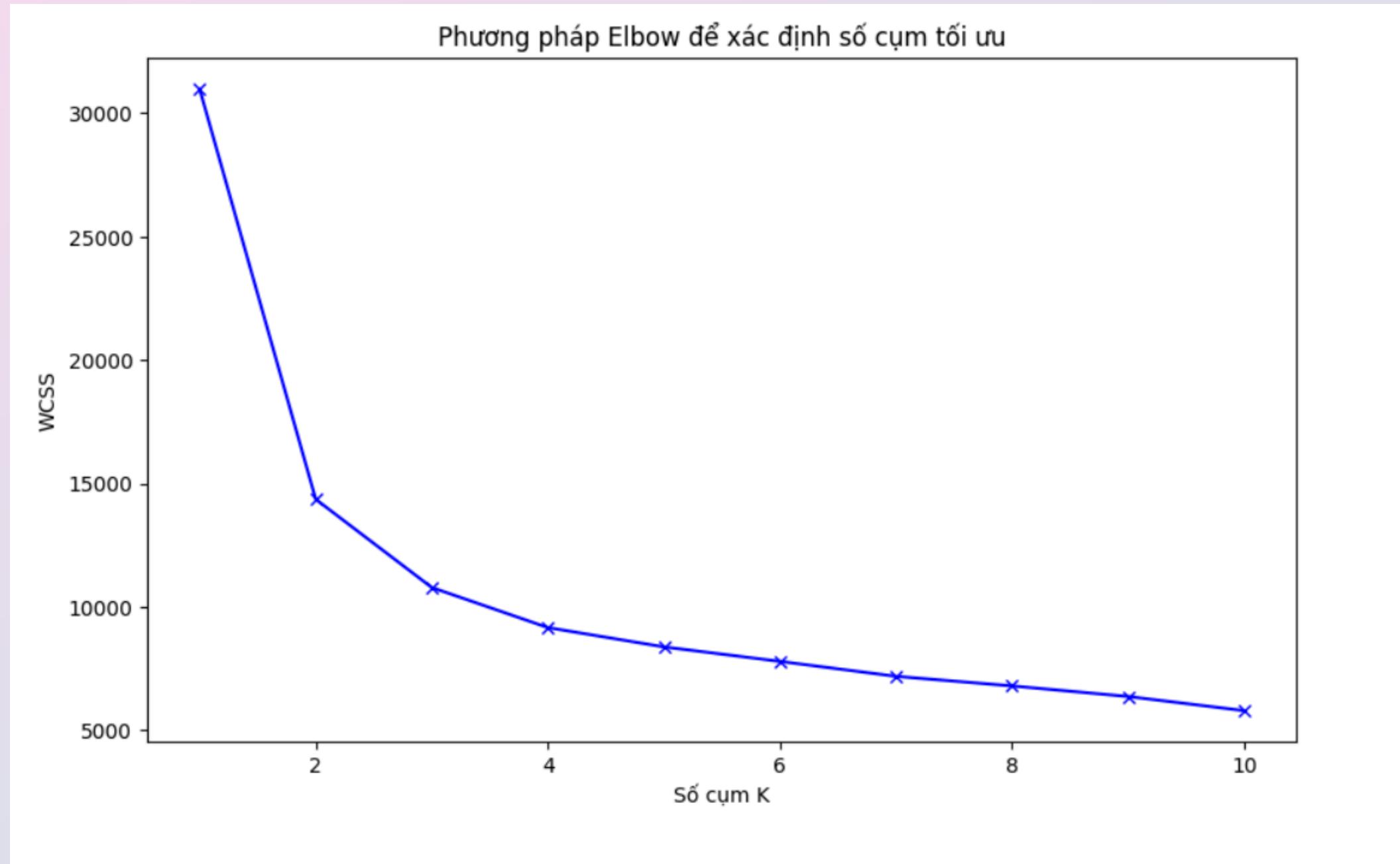
4.5 Thuật Toán K-means

Cách thức hoạt động của thuật toán kmeans như sau:

- Xác định số lượng các cụm K .
- Khởi tạo các trung tâm bằng cách xáo trộn tập dữ liệu trước và sau đó chọn ngẫu nhiên K điểm dữ liệu cho các trung tâm mà không cần thay thế.
- Tiếp tục lặp lại cho đến khi không có thay đổi đối với centroid. tức là việc gán các điểm dữ liệu cho các cụm không thay đổi.
- Tính tổng bình phương khoảng cách giữa các điểm dữ liệu và tất cả các centroid.
- Gán mỗi điểm dữ liệu cho cụm gần nhất (centroid).
- Tính toán trọng tâm cho các cụm bằng cách lấy giá trị trung bình của tất cả các điểm dữ liệu thuộc mỗi cụm.

4.5 Thuật Toán K-means

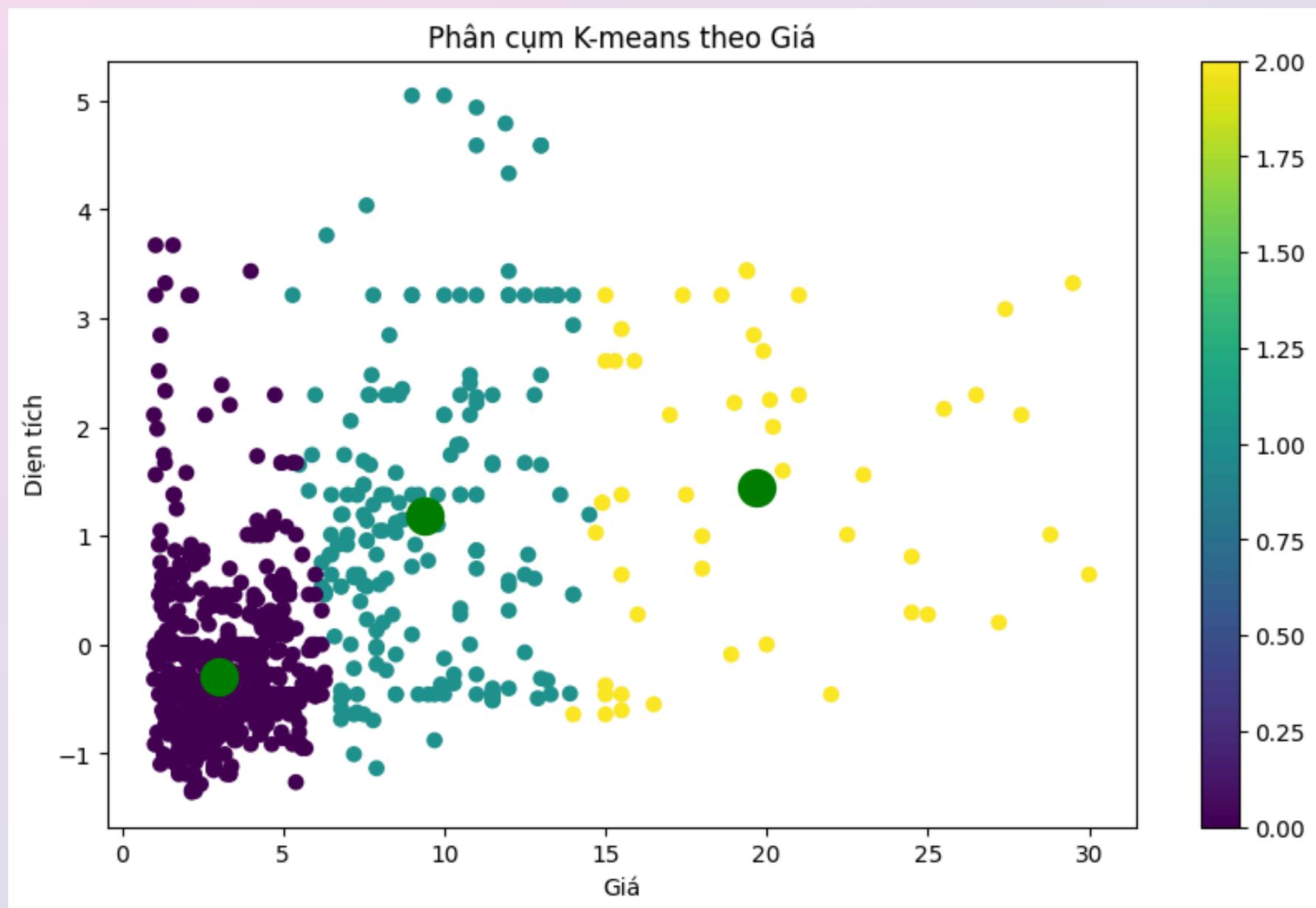
Lựa chọn tham số K



- Điểm khuỷu tay là điểm mà ở đó tốc độ suy giảm của hàm biến dạng sẽ thay đổi nhiều nhất. Tức là kể từ sau vị trí này thì gia tăng thêm số lượng cụm cũng không giúp hàm biến dạng giảm đáng kể. Suy ra chọn $K = 3$.

4.5 Thuật Toán K-means

Kết quả của phân cụm

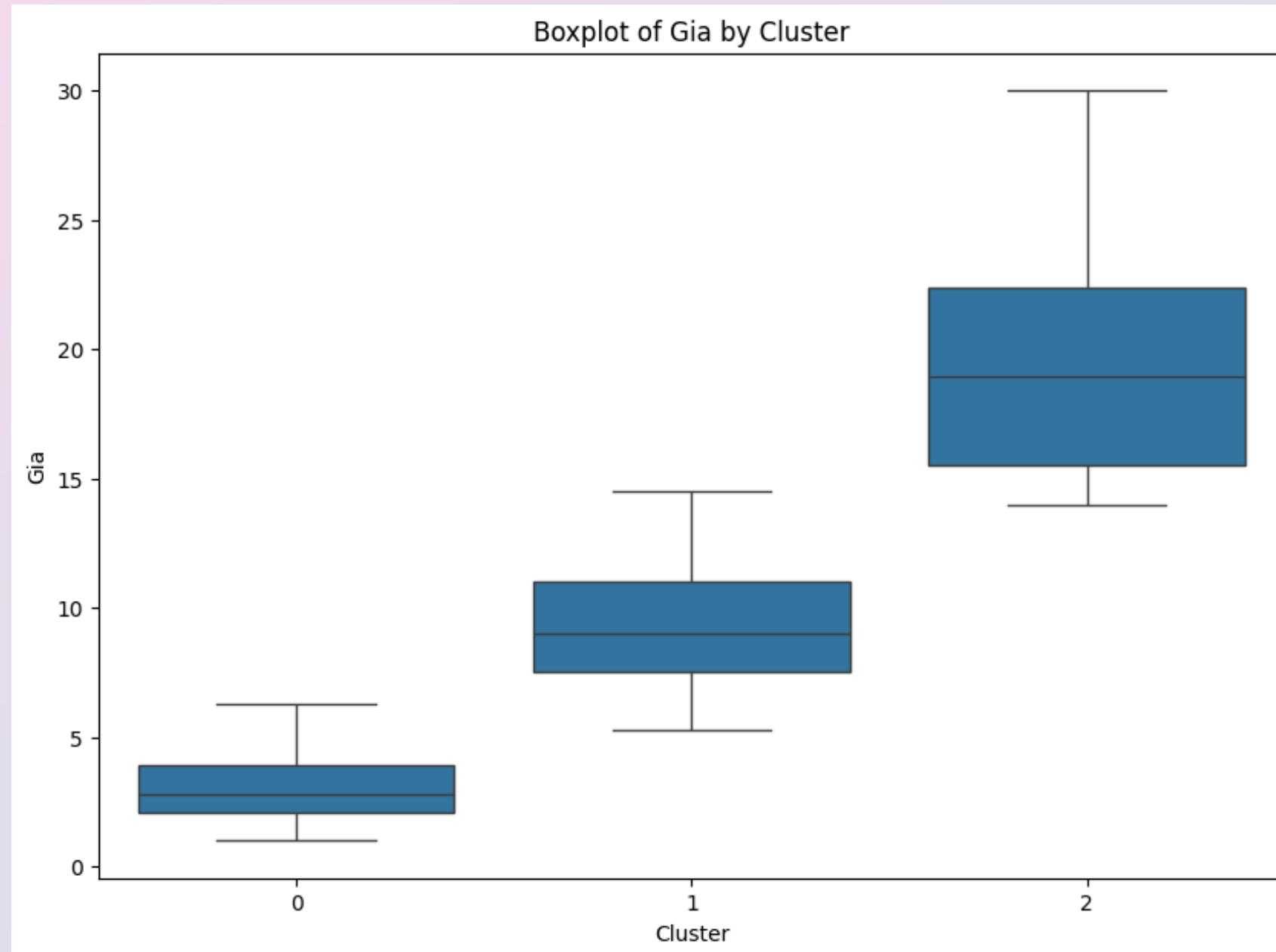


Nhận xét:

- Thông tin cụm được bảo toàn và giữa các cụm có ranh giới rõ ràng.
- Dữ liệu được chia thành 3 cụm

4.5 Thuật Toán K-means

Kết quả của phân cụm



Nhận xét:

- Cụm 0 có số lượng quan sát nhiều nhất và giá trị trung bình thấp nhất, chỉ ra một nhóm lớn với các giá trị thấp và ít biến động.
- Cụm 1 có ít quan sát hơn với giá trị trung bình cao hơn và biến động vừa phải, cho thấy một nhóm rõ ràng với các giá trị trung bình.
- Cụm 2 là nhóm nhỏ nhất nhưng có giá trị trung bình cao nhất và biến động lớn nhất, chỉ ra một nhóm rõ ràng với các giá trị cao nhất.

4.6 Mô hình GaussianNB

- Là một mô hình học máy được sử dụng rộng rãi trong phân loại. Trong mô hình này, chúng ta giả định rằng các đặc trưng (features) của dữ liệu được phân phối theo phân phối Gaussian
- Cách hoạt động:
 1. Giả định đặc trưng độc lập: Mô hình giả định rằng các đặc trưng (features) trong dữ liệu là độc lập có điều kiện
 2. Huấn luyện: Trong quá trình huấn luyện, mô hình ước lượng các tham số của phân phối Gaussian cho mỗi lớp dựa trên dữ liệu huấn luyện.
 3. Dự đoán: Khi có một mẫu mới cần dự đoán, mô hình tính toán xác suất cho mẫu thuộc vào mỗi lớp bằng cách sử dụng phân phối Gaussian đã học được và giả định đặc trưng độc lập.
 4. Chọn lớp: Mô hình chọn lớp có xác suất cao nhất cho mẫu mới. Cụ thể, nó chọn lớp mà mẫu có xác suất cao nhất thuộc vào lớp đó.

4.6 Mô hình GaussianNB

	precision	recall	f1-score
0	1.00	0.97	0.98
1	0.84	0.93	0.88
2	0.74	0.93	0.82
accuracy			0.96
macro avg	0.86	0.94	0.90
weighted avg	0.97	0.96	0.96

- Độ chính xác mô hình: 0.9623

Nhận xét:

- Mô hình hoạt động rất tốt cho lớp 0 với độ chính xác tuyệt đối và độ nhạy cao.
- Đối với lớp 1, mô hình cũng có hiệu suất cao nhưng có một chút giảm về độ chính xác (precision).
- Đối với lớp 2, mặc dù số lượng mẫu ít, mô hình vẫn có độ nhạy cao nhưng độ chính xác (precision) thấp hơn so với các lớp khác.

4.7 Mô hình Random Forest Classifier

- Là tạo một tập hợp các cây quyết định từ một tập hợp con được chọn ngẫu nhiên của tập huấn luyện.
- Nó là một tập hợp các cây quyết định từ một tập hợp con được chọn ngẫu nhiên của tập hợp đào tạo và sau đó nó thu thập các phiếu bầu từ các cây quyết định khác nhau để quyết định dự đoán cuối cùng.
- Chia tập huấn luyện: kiểm thử theo tỉ lệ 63:37

4.7 Mô hình Random Forest Classifier

	precision	recall	f1-score
0	1.00	1.00	1.00
1	0.99	1.00	0.99
2	1.00	1.00	1.00
accuracy			1.00
macro avg	1.00	1.00	1.00
weighted avg	1.00	1.00	1.00

- Độ chính xác mô hình: 0.9979

Nhận xét:

- Mô hình hoạt động rất tốt cho lớp 0,1,2 với độ chính xác tuyệt đối và độ nhạy cao.

4.7 Mô hình Random Forest Classifier

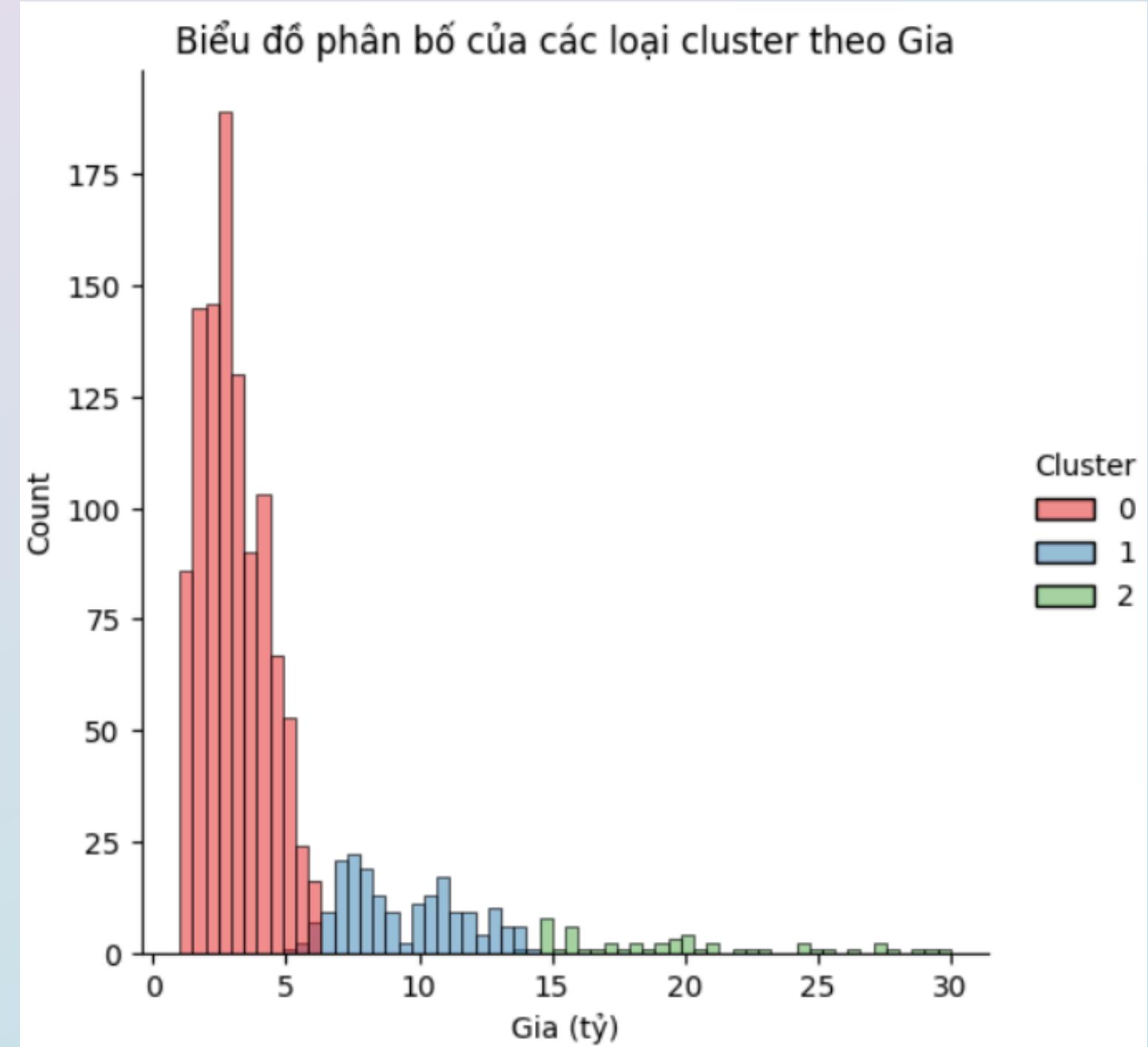
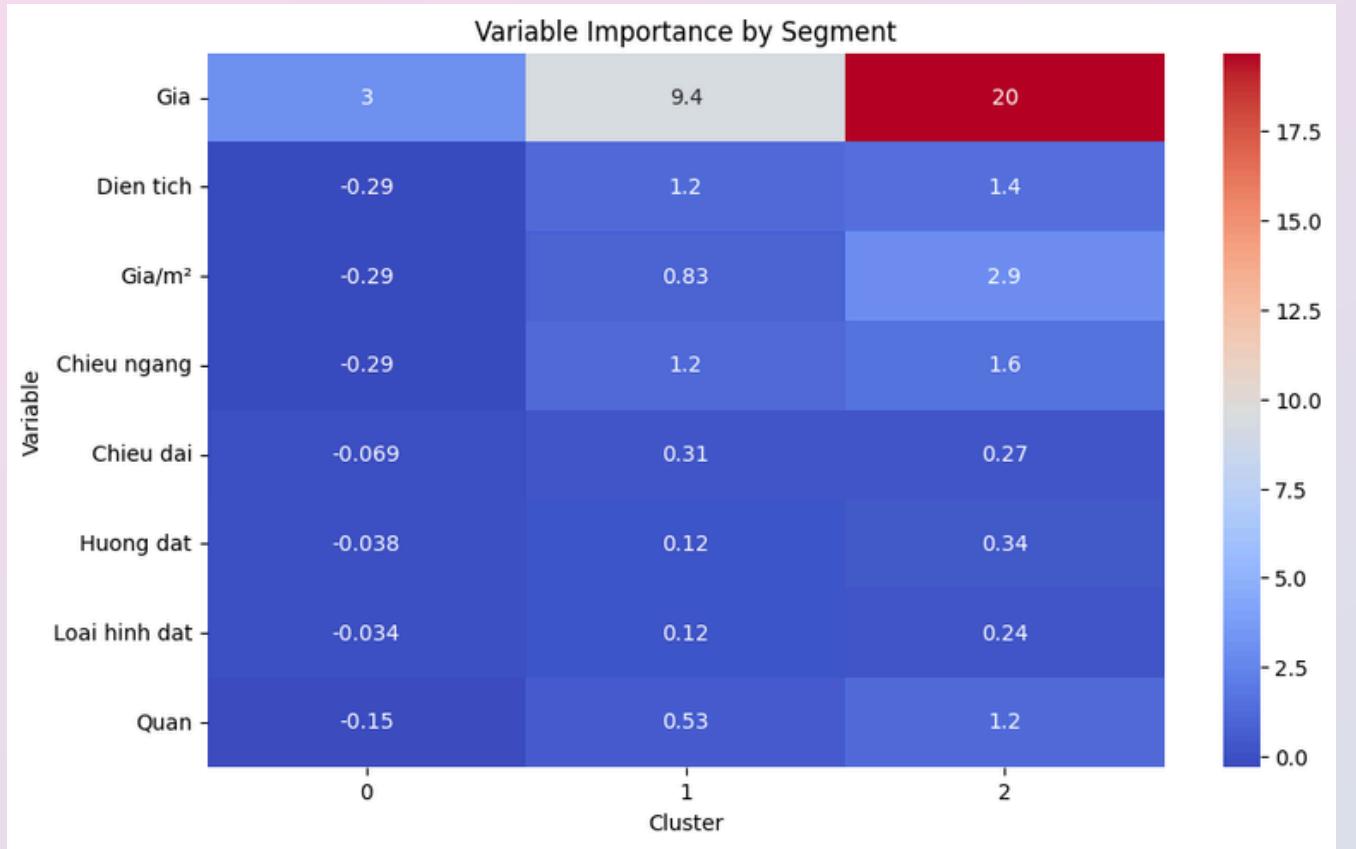
Kiểm tra mô hình bằng kỹ thuật cross-validation

	Accuracy
Fold 1	1.0
Fold 2	0.9961
Fold 3	0.9883
Fold 4	0.9961
Fold 5	0.9844

Nhận xét:

- Mô hình cho kết quả tốt trên các Fold khác nhau, gần với giá trị hoàn hảo 1

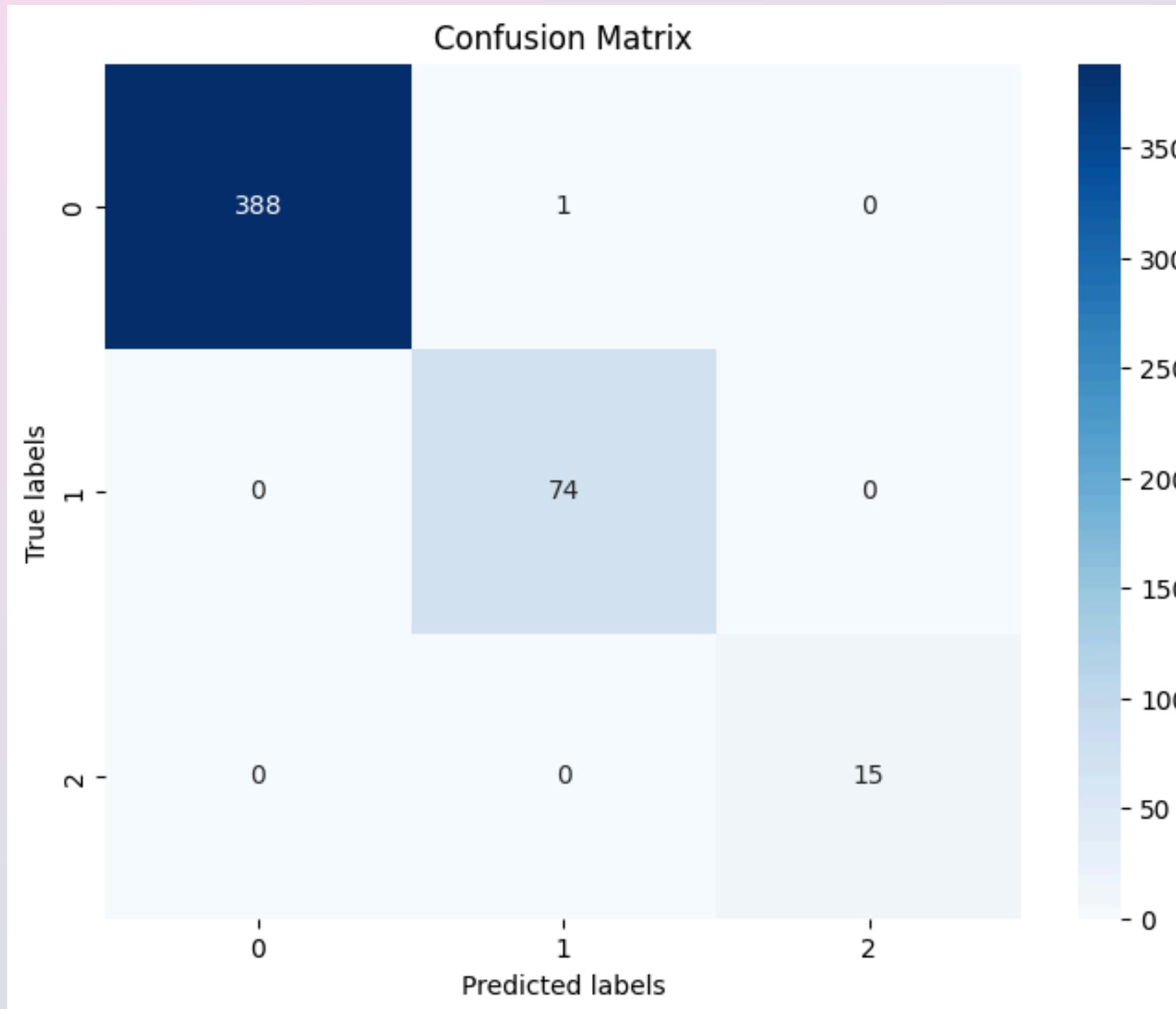
4.7 Mô hình Random Forest Classifier



Nhận xét:

- biến Gia có mức độ quan trọng cao nhất đối với mỗi loại
- biến loai hinh dat có mức độ quan trọng thấp nhất đối với mỗi loại

4.7 Mô hình Random Forest Classifier



Nhận xét:

- Mô hình dự đoán tốt, chỉ nhầm lẫn rất ít mẫu

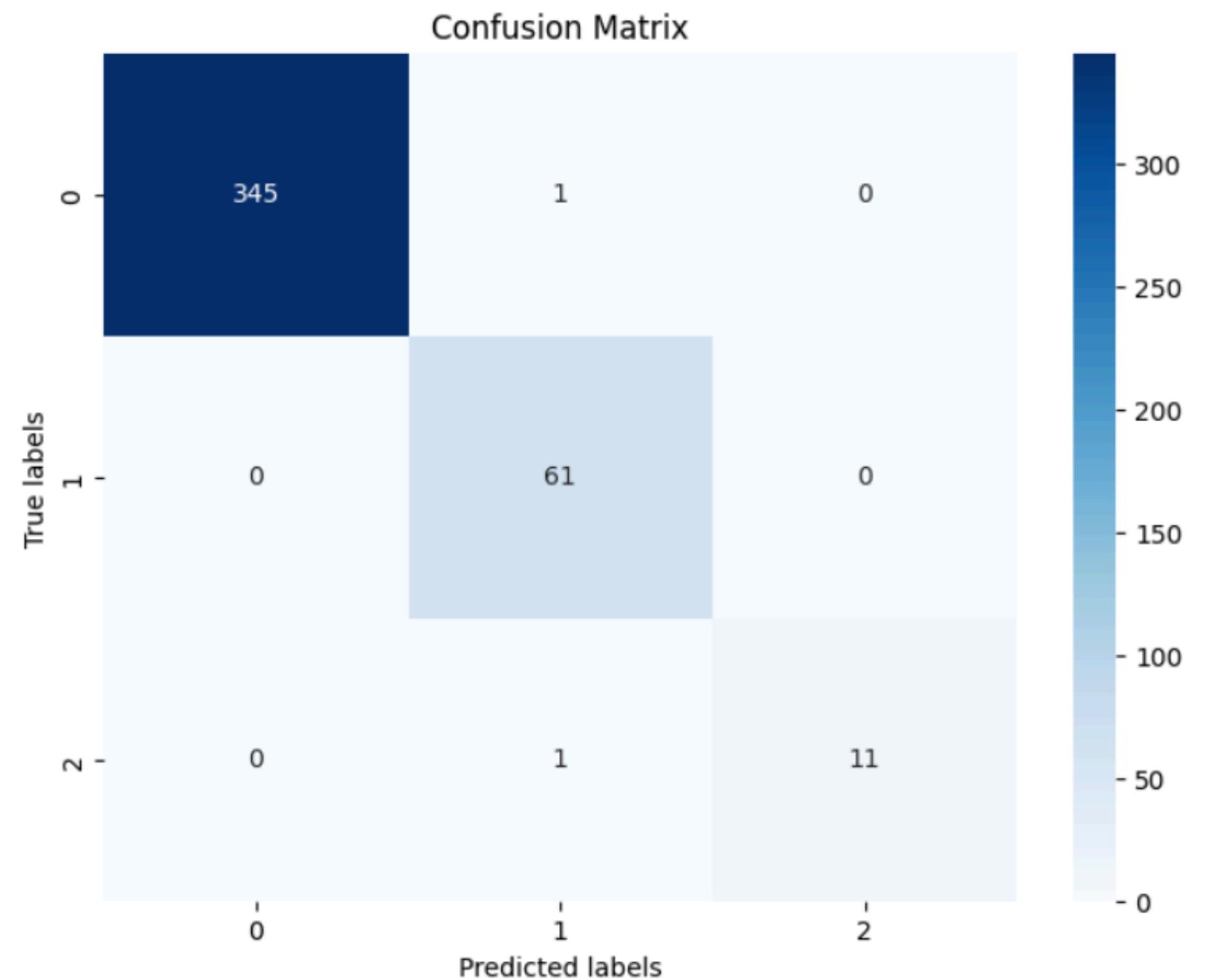
4.8 So Sánh Hai Mô Hình Phân Loại

	GaussianNB	Random Forest Classifier
Accuracy	0.9623	0.9979
Weighted avg precision	0.97	1
Weighted avg recall	0.96	1
Weighted avg f1-score	0.96	1

Nhận xét:

- Tất cả giá trị của Random Forest Classifier đều tốt hơn GaussianNB. Nên ta sẽ chọn mô hình Random Forest Classifier

4.9 Kiểm Tra Mô Hình Random Forest Classifier



- Giá trị Accuracy: 0.9928

Nhận xét:

- Mô hình dự đoán chính xác, nhầm lẫn rất ít

5

KẾT LUẬN

5. Kết Luận

Kết quả đạt được:

- Đạt được một mô hình dự đoán giá đất nền với độ chính xác cao
- Mô hình có thể hiển thị được mối quan hệ giữa các biến độc lập và giá đất nền, giúp người dùng hiểu rõ hơn về yếu tố ảnh hưởng đến giá đất.
- Xây dựng một mô hình phân loại giá đất nền với độ chính xác cao
- Mô hình có khả năng phân loại các khu vực thành các nhóm giá đất khác nhau, giúp người dùng hiểu rõ hơn về phân bố giá đất trên thị trường.

5. Kết Luận

Hướng phát triển:

- Tăng cường thu thập dữ liệu: Thu thập thêm dữ liệu từ nhiều nguồn khác nhau để tăng tính đa dạng và độ chính xác của mô hình.
- Cải thiện tiền xử lý dữ liệu: Xử lý dữ liệu thiếu và dữ liệu nhiễu một cách kỹ lưỡng để tăng chất lượng dữ liệu.
- Tích hợp tính năng dự báo thời gian thực: Phát triển các ứng dụng hoặc hệ thống có khả năng dự báo giá đất nền theo thời gian thực dựa trên dữ liệu đầu vào cập nhật liên tục.
- Phân tích sâu hơn về yếu tố ảnh hưởng: Thực hiện phân tích sâu hơn về từng yếu tố ảnh hưởng đến giá đất nền để hiểu rõ hơn về mối quan hệ giữa các biến số và giá đất.

**THANK
YOU**