|  |  |  |
| --- | --- | --- |
|  | **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 bán đất ở Hà Nội

|  |  |
| --- | --- |
| Nhóm 5 |  |
| Họ Và Tên Sinh Viên | Lớp Học Phần |
| Nguyễn Hiếu | 2091A |
| Nguyễn Hồng Hạnh |
| Trương Thành Đạt |

ĐÀ NẴNG, 06/2023

**TÓM TẮT**

Hà Nội là một trong những thị trường bất động sản trọng điểm của khu vực miền Bắc nói riêng và Việt Nam nói chung. Thị trường bất động sản Hà Nội 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 Hà Nội. 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.....

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | Các nhiệm vụ | Tự đánh giá theo 3 mức  (Đã hoàn thành/Chưa hoàn thành/Không triển khai) |
| Nguyễn Hiếu | - Crawl dữ liệu  - Trích xuất đặc trưng  - Viết báo cáo  - Viết slide | - Đã hoàn thành  - Đã hoàn thành  - Đã hoàn thành |
| Nguyễn Hồng Hạnh | - Crawl dữ liệu  - Thu thập và mô tả dữ liệu  - Viết báo cáo  - Viết slide | - Đã hoàn thành  - Đã hoàn thành  - Đã hoàn thành  - Đã hoàn thành |
| Trương Thành Đạt | - Crawl dữ liệu  - Mô hình hóa dữ liệu  - Viết báo cáo  - Viết slide | - Đã hoàn thành  - Đã hoàn thành  - Đã hoàn thành |

**MỤC LỤC**

Nội dung

[1. Giới thiệu 5](#_Toc135600701)

[2. Thu thập và mô tả dữ liệu 5](#_Toc135600702)

[2.1. Thu thập dữ liệu 5](#_Toc135600703)

[2.2. Mô tả dữ liệu 7](#_Toc135600704)

[3. Trích xuất đặc trưng 11](#_Toc135600705)

[4. Mô hình hóa dữ liệu 14](#_Toc135600706)

[4.1 Các mô hình 14](#_Toc135600707)

[**4.1.1 Linear Regression** 14](#_Toc135600708)

[**4.1.2 Random Forest** 15](#_Toc135600709)

[4.2. Chia dữ liệu 16](#_Toc135600710)

[4.3 Tham số của quá trình huấn luyện 16](#_Toc135600711)

[4.4 Hiệu suất của mô hình trên tập huấn luyện và tập xác thực dựa trên độ đo MAE 16](#_Toc135600712)

[4.5 Áp dụng mô hình vào thực tế 18](#_Toc135600713)

[5. Kết luận 18](#_Toc135600714)

[5.1 Tổng kết 18](#_Toc135600715)

[5.2 Kết quả đạt được 19](#_Toc135600716)

[5.3 Hướng phát triển 19](#_Toc135600717)

[6. Tài liệu tham khảo 19](#_Toc135600718)

# 1. Giới thiệu

Vấn đề cần giải quyết :

- Hà Nội là nơi có thị trường bất động sản hoạt động rất sôi nổi, song hàng 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 này là rất khó.

Đề xuất giải pháp tổng quan cần thực hiện :

- Để giải quyết vấn đề đó, nhóm em đã đưa ra giải pháp là xây dựng một mô hình học máy sử dụng mô hình Linear Regression và Ramdom forest để đưa ra dự đoán về giá của một bất động sản ở thành phố này dựa trên những dữ liệu thu nhập được.

# 2. Thu thập và mô tả dữ liệu

## 2.1. Thu thập dữ liệu

Giải pháp thu thập dữ liệu :

- Nguồn dữ liệu : <https://alonhadat.com.vn/nha-dat/can-ban/dat-tho-cu-dat-o/1/ha-noi/trang--2.html>

- Công cụ thu thập : sử dụng ngôn ngữ Python và các thư viện beautifulsoup và

selenium

- Cách thức sử dụng công cụ :

* Xác đinh yêu cầu thu thập dữ liệu : thu thập thông tin bao gồm diện tích , giá bán đất , địa chỉ , quận huyện , bãi đỗ xe , nhà đất cách đường ,…
* Cấu hình công cụ : cài đặt các thư viện như beautifulsoup và selenium. Công cụ được cấu hình để truy cập vào trang web và lấy thông tin từ các trang chứa thông tin về giá bán đất ở Hà Nội.
* Lấy dữ liệu từ nguồn : sử dụng thư viện selenium để mở trình duyệt và tải nội dung của trang web . Sau đó sử dụng beautifulsoup để phân tích cú pháp
* Xử lý trường hợp xuất hiện reCAPTCHA : sẽ tạm dừng chương trình trong 60 giây để giải quyết reCAPTCHA
* Trích xuất thông tin : công cụ sẽ tự động duyệt qua , tìm kiếm và trích xuất các thông tin cần thiết, chẳng hạn như diện tích, giá bán và các yếu tố liên quan khác.
* Lưu trữ dữ liệu : Kết quả thu thập sẽ được lưu trữ vào một tệp để sử dụng cho việc dự đoán giá bán đất. Công cụ sẽ tự động ghi dữ liệu đã trích xuất vào một tệp csv .

- Đầu vào quá trình thu thập :

* urls : đường dẫn đến trang web muốn lấy dữ liệu
* num\_pages : số trang muốn lấy dữ liệu

- Đầu ra quá trình thu thập : 2 file csv lần lượt là SmallDS.csv và BigDS.csv

- Ví dụ minh họa : Crawl dữ liệu 1 mẫu của trang web : <https://alonhadat.com.vn/nha-dat/can-ban/dat-tho-cu-dat-o/1/ha-noi/trang--2.html> . Mỗi trang sẽ có nhiều bài viết rao bán đất

Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Trang web

Mô tả được tạo tự động

Hình 1: Trang chủ website crawl dữ liệu

Lấy thông tin từ các trang , mỗi bài viết sẽ là 1 mẫu dữ liệu

Ảnh có chứa văn bản, Trang web, Website, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 2 : Một mẫu dữ liệu bán nhà đất

Duyệt qua trang đã cung cấp và trích xuất thông tin cần lấy như diện tích , giá , vị trí , tiêu đề , nhà cách đường ,. Sử dụng python và các thư viện để tải nội dung và phân tích cú pháp HTML và trích xuất thông tin từ các phần tử . Sau khi có được dữ liệu, ta chỉ cần lưu vào file csv là đã hoàn thành.

Hình 3 : Kết quả sau khi lấy được mẫu dữ liệu trong file csv

- Xử lý dữ liệu :

* Chuyển đợn vị cột price về triệu / m2 , chuyển kiểu cột road
* Xử lý dữ liệu bị trùng lặp : dùng drop\_duplicates với các đặc trưng ‘title’,’price’,’square’,’dis’ => Kết quả : 1043 còn 1027 mẫu
* Thêm cột total\_price = price\*square

## 2.2. Mô tả dữ liệu

SV mô tả các thống kê tổng quan về tập dữ liệu (số mẫu, số đặc trưng của 1 mẫu, kiểu dữ liệu của mỗi đặc trưng, số mẫu dữ liệu trống của mỗi đặc trưng,...) và xuất ra các thống kê mô tả trực quan về các đặc trưng (ví dụ: dùng boxplot, histogram, scatter plot về độ tương quan,…).

- Mô tả các thống kê tổng quan về tập dữ liệu :

* Có 1027 mẫu và mỗi mẫu có 10 đặc trưng
* Kiểu dữ liệu và số mẫu dữ liệu trống của mỗi đặc trưng

Bảng 1 : Bảng kiểu dữ liệu và số mẫu dữ liệu trống của mỗi đặc trưng

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Số mẫu dữ liệu bị trống |
| stt | int64 | 0 |
| title | object | 0 |
| road | float64 | 2 |
| parking | int64 | 0 |
| square | int64 | 0 |
| price | float64 | 0 |
| dis\_q | object | 0 |
| dis | object | 0 |
| date | object | 0 |
| total\_price | float64 | 0 |

Chú thích :

* + stt : số thứ tự của mẫu dữ liệu
  + title : tiêu đề của bài viết bán nhà đất Hà Nội
  + road : khoảng cách từ nhà đất đến đường ( m)
  + parking : bãi đổ xe ( 1 : có , 0 : không có)
  + square : diện tích của nhà đất ( m2 )
  + price : giá tiền trên mỗi mét vuông (triệu)
  + dis\_q : quận / huyện / thị xã của nhà đất ở Hà Nội
  + dis : địa chỉ chi tiết của nhà đất ở Hà Nội
  + date : ngày bài viết bán đất được đăng lên trang web
  + total\_price : giá tiền nhà đất (triệu)

- Các thống kê mô tả trực quan về các đặc trưng :

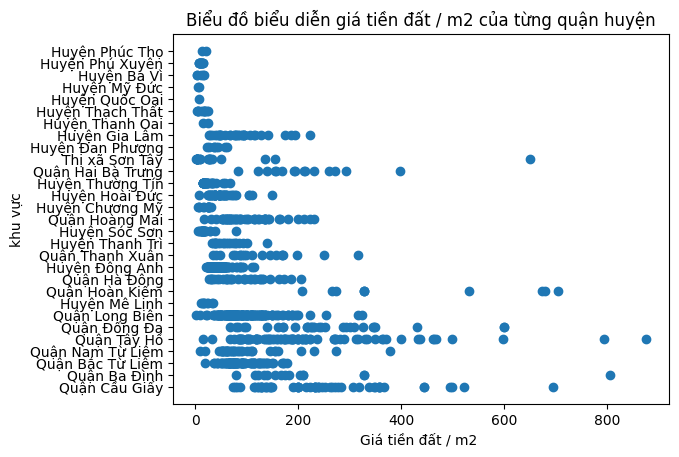
* Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

  Mô tả được tạo tự độngSử dụng hisplot biểu diễn số lượng các bài bán đất của các quận huyện ở Hà Nội

Hình 4 :Biểu đồ số lần xuất hiện của từng quận huyện ở Hà Nội

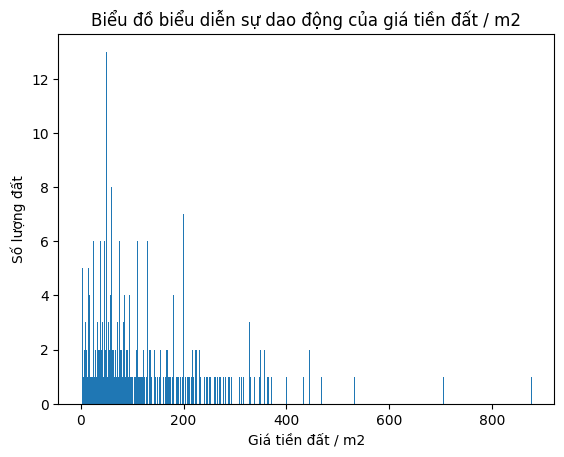
Kết luận : Quận Long Biên chiếm tỉ lệ cao nhât, Huyện Quốc Oai có tỉ lệ thấp nhất

* Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng price và dis\_q



Hình 5 : Biểu đồ biểu diễn giá tiền đất trên mỗi m2 của từng quận huyện

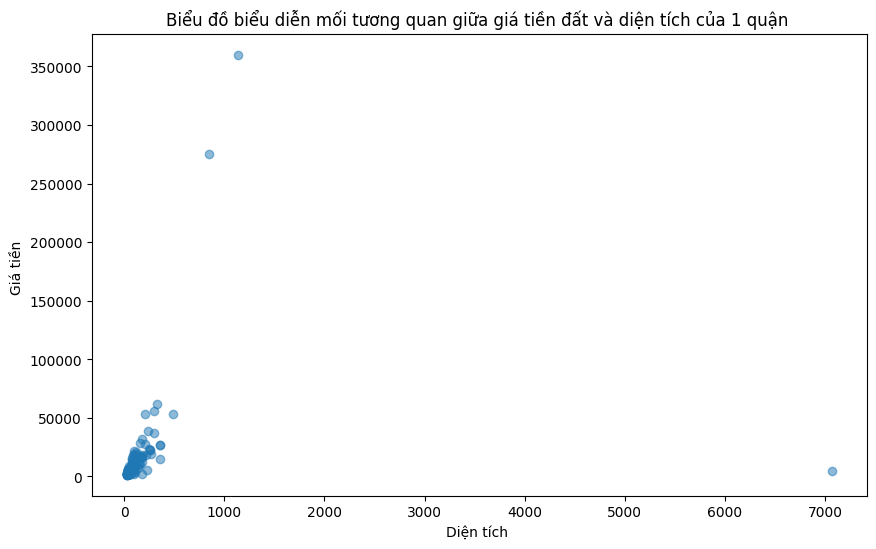
Kết luận : Có một số mẫu đất có giá tiền đất / m2 lớn bất ngờ



Hình 6 : Biểu đồ biểu diễn sự dao động của giá tiền đất trên mỗi mét vuông

Kết luận : Giá đất trung bình từ khoảng 0 đến 250 triệu / m2 chiếm nhiều nhất

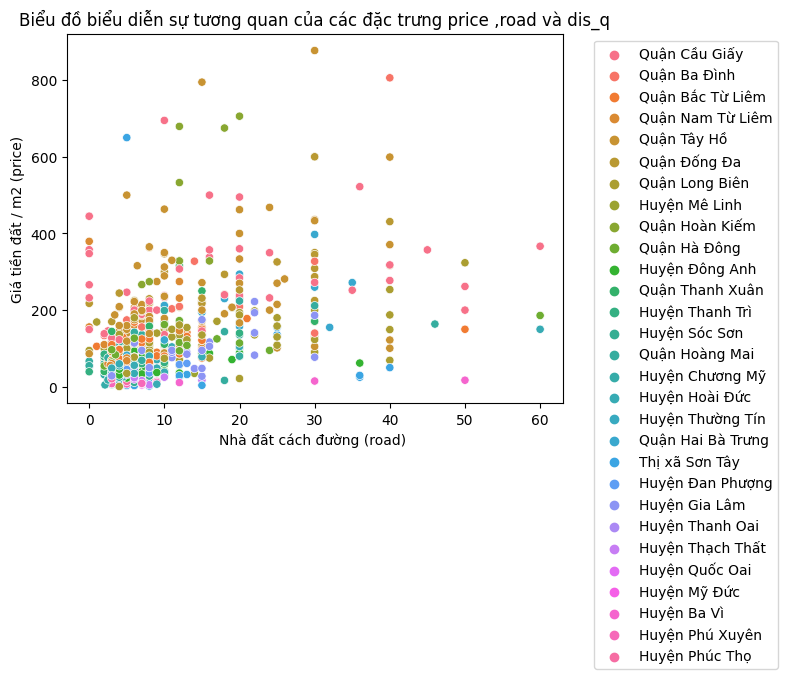
Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng price và square



Hình 7 : Biểu đồ biểu diễn mối tương quan giữa giá tiền đất và diện tích của một quận

Kết luận : Hình trên cho thấy hầu hết diện tích tỉ lệ thuận với giá tiền, diện tích tăng giá cũng tăng ở quận Long Biên

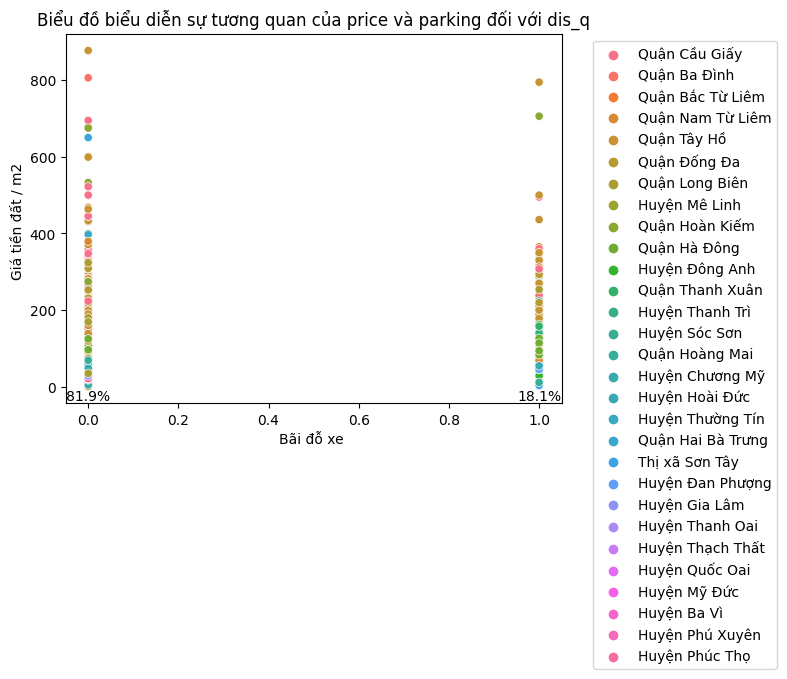
* Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng price, road và dis\_q



Hình 8 : Biểu đồ biểu diễn sự tương quan của price , road , dis\_q

Kết luận : Đa số các mẫu đất cách đường từ 0m đến 20m

* Sử dụng scatterplot để mô tả tương quang giữa các đặc trưng price, parking và dis\_q



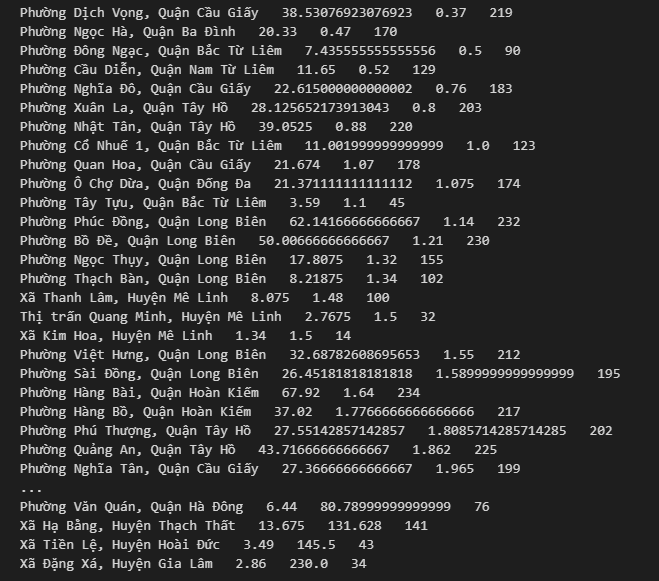
Hình 9 : Biểu đồ biểu diễn sự tương quan của price , parking ,dis\_q

Kết luận :Nhà đất không có bãi đỗ xe chiếm 91,9% cao gấp 4.5 lần có bãi đỗ xe

# 3. Trích xuất đặc trưng

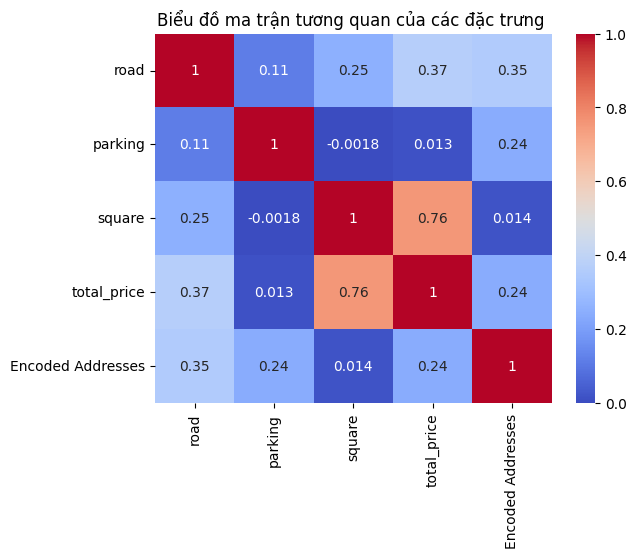
- Tạo mới 1 đặc trưng Encoded Addresses dựa vào đặc trưng dis\_q và đặc trưng price:

+ Bước 1: lấy dữ liệu tên các phường/thị trấn/xã và tính giá đất trung bình trên 1m2 thuộc phường/thị trấn/xã đó.

 + Bước 2: Sắp xếp phường/thị trấn/xã theo giá đất trung bình trên 1m2 tăng dần và đánh nhãn cho các phường/thị trấn/xã.

Hình 10 : Danh sách phường , thị xã trên một mét vuông sau khi được sắp xếp

- Lựa chọn đặc trưng theo biểu đồ ma trận tương quan của các đặc trưng:

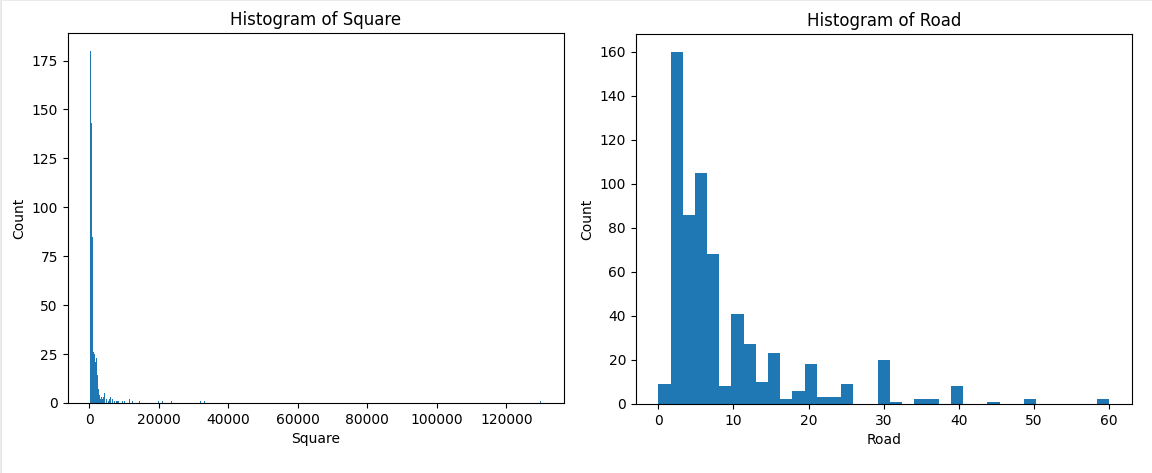


Hình 11 : Biểu đồ ma trận tương quan của các đặc trưng

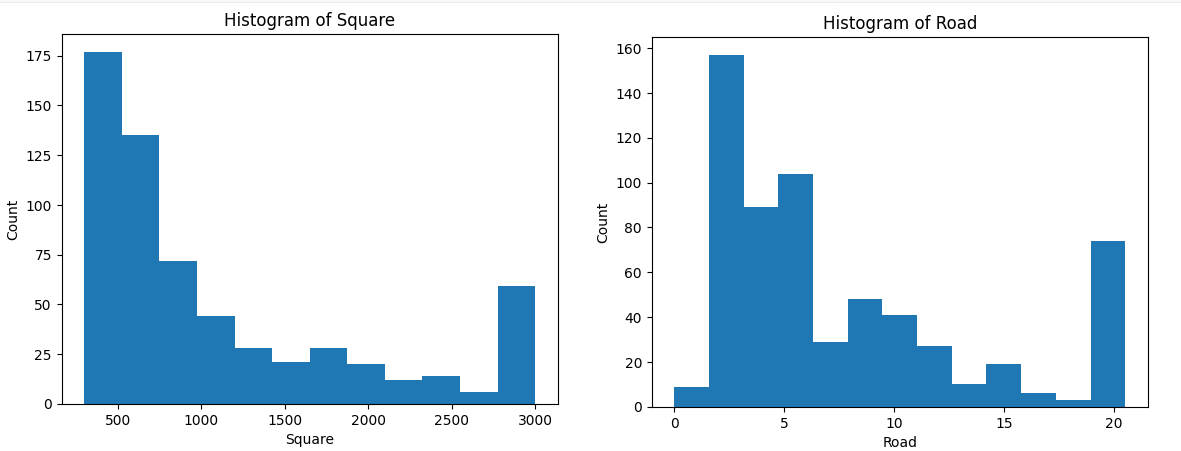
- Dựa theo biểu đồ trên lấy 3 đặc trưng để huấn luyện: road, square, Encoded Addresses.

- Xử lý dữ liệu trống bằng phương pháp Mean

- Xử lý dữ liệu ngoại lệ bằng phương pháp Skewed của 2 đặc trưng: road, square



Hình 12 : Biểu đồ các đặc trưng road,square trước khi xử lý ngoại lệ



Hình 13 : Biểu đồ các đặc trưng road,square sau khi xử lý ngoại lệ

# 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 của Python.

### **4.1.1 Linear Regression**

* Lý thuyết:

Linear Regression (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 = [𝑥1 , 𝑥2 , … , 𝑥𝑛] là vector chưa thông tin input, y là một vô hướng biểu diễn output. y sẽ có dạng

𝑦 ≈ 𝑓(𝑥) = 𝑦̂

𝑓(𝑥) = 𝑤1𝑥1 + 𝑤2𝑥2 + ⋯ + 𝑤𝑛𝑥𝑛 + 𝑤0 (1)

Đặt 𝑤 = [𝑤0 , 𝑤1 , … , 𝑤𝑛] 𝑇 là vector (cột) hệ số cần phải tối ưu và 𝑥̅= [1, 𝑥1 ,𝑥2 , … , 𝑥𝑛] là vector dữ liệu đầu vào mở rộng thì (1) có thể được viết lại dưới dạng: 𝑦 ≈ 𝑥̅𝑤 = 𝑦̂

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

* Tham số của mô hình

a. 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.

b. normalize: Có chuẩn hóa hay không, tham số này sẽ bị bỏ qua khi fit\_intercept bằng False. Mặc định là deprecated.

c. copy\_X: Nếu bằng True thì dữ liệu sẽ được copy, ngược lại sẽ bị ghi đè. Mặc định là True.

d. n\_jobs: Số job cần dùng để tính toán, dùng để tăng tốc độ trong trường hợp dữ liệu lớn. Mặc định là None.

e. 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 Random Forest**

* Lý 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 sốc 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. 22
* 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. o 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 CostComplexity. 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.

## 4.2. Chia dữ liệu

Dữ liệu sau khi xử lí dữ liệu trống gồm 1027 mẫu được chia thành 3 tập: huấn luyện (train), xác thực (validate) và kiểm thử (test) theo tỉ lệ 60: 20: 20.

## 4.3 Tham số của quá trình huấn luyện

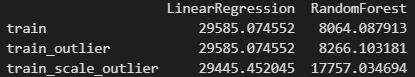
Quá trình huấn luyện diễn ra với 2 mô hình

+ LinearRegression()

+ RandomForestRegressor()

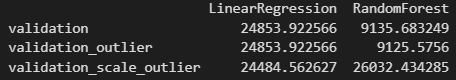
## 4.4 Hiệu suất của mô hình trên tập huấn luyện và tập xác thực dựa trên độ đo MAE

Hình kết quả trên tập train



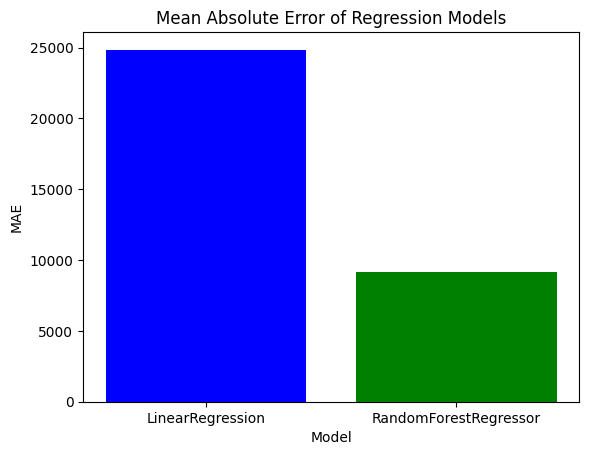
Hình 14 : Kết quả của MAE trên tập train

Hình kết quả trên tập valid



Hình 15 : Kết quả của MAE trên tập validation

* Qua 2 hình trên ta thấy dữ liệu kết hợp xử lý ngoại lệ và chuẩn hóa dữ liệu có giá trị MAE chênh lệch lớn so với phần còn lại. Vì vậy không dùng loại dữ liệu này vào mô hình thực tế.



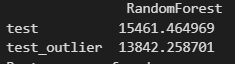
Hình 16 : Hình so sánh giá trị MAE trên tập validation

* Mô hình RandomForestRegressor() cho kết quả tốt hơn nên sẽ chọn mô hình này để thực hiện trên tập kiểm thử.



Hình 17 : Kết quả MAE trên tập test

Dùng Grid Search SV của thử viện scikit learn để tìm ra bộ siêu tham số tốt nhất cho mô hình vừa tìm được

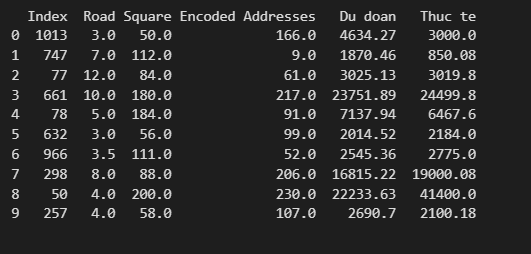


Hình 18 : Kết quả MAE sau khi áp dụng siêu tham số trên tập test

## 4.5 Áp dụng mô hình vào thực tế

Sau khi tìm được bộ siêu tham số, áp dụng vào mô hình Random Forest Regressor, chọn ngẫu nhiên 10 bất động sản trên tập dữ liệu kiểm thử và đưa vào dự đoán.

Bảng kết quả :



Hình 19 : Bảng kết quả dự đoán của 10 dữ liệu ngẫu nhiên

# 5. Kết luận

## 5.1 Tổng kết

- Tạo ra thêm được 1 đặc trưng mới để dự đoán giá đất theo giá đất trên 1m2 thuộc phường/thị trấn/xã của Hà Nội

## - Từ 2 mô hình Linear Regression và Random Forest Regression chọn ra mô hình tốt nhất cho bài toán, dựa vào mô hình đó chọn ra siêu tham số để cải thiện mô hình.

## 5.2 Kết quả đạt được

- Theo mô hình đã xây dựng, được kết quả dự đoán được chênh lệch với giá thực tế còn sai số nhiều. Hơi ấy

## 5.3 Hướng phát triển

Để cải thiện kết quả hiện tại của mô hình dự đoán bán đất ở Hà Nội , có thể thực hiện một số hướng phát triển như sau :

- Bổ sung thêm dữ liệu để mô hình dự đoán được chính xác hơn

- Mở rộng đặc trưng có thể bao gồm thông tin về tiện ích xung quanh (như gần trường học, bệnh viện, công viên),các yếu tố kinh tế xã hội khác để nâng cao khả năng dự đoán

- Viết them 1 câu nữa …

# 6. Tài liệu tham khảo

* Thư viện scikit learn:<https://scikit-learn.org/stable/>
* LinearRegression:<https://viblo.asia/p/linear-regression-hoi-quy-tuyen-tinh-trong-machine-learning-4P856akRlY3>
* RandomForestRegressor:<https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html>