ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

DỰ ĐOÁN GIÁ THUÊ NHÀ TẠI THÀNH PHỐ HỒ CHÍ MINH

Sinh viên thực hiện:			
STT	Họ tên	MSSV	Ngành
1	Nguyễn Minh Thuận	20520797	Ngành KHMT
2	Đỗ Thị Thu Trang	20520816	Ngành KHMT
3	Nguyễn Đặng Bảo Ngọc	20521663	Ngành KHMT

TP. HỒ CHÍ MINH – 12/2023

1. GIỚI THIỆU

Thành phố Hồ Chí Minh là thành phố có số lượng và mật độ dân cư lớn nhất Việt Nam. Do đó, những vấn đề như chỗ ở cũng trở thành một vấn đề nan giải của sinh viên tỉnh ngoài hoặc dân lao động khi sống tại thành phố này. Là sinh viên thuê trọ, chúng tôi đã phải đóng giá nhà cao hơn sơ với giá thuê mặt bằng chung của khu vực vì thiếu kinh nghiệm. Do đó, để tránh bị lừa sau này, chúng tôi xây dựng mô hình dự đoán giá thuê nhà trên địa bàn thành phố Hồ Chí Minh.

Ý tưởng của nhóm chúng tôi là sẽ dựa vào giá thuê thực tế ứng với các điều kiện của nhà như: diện tích, số phòng ngủ, số nhà vệ sinh, … mô hình học và khi nhập vào một danh sách các điều kiện mà nhà đang có, mô hình sẽ đưa ra dự đoán giá thuê của căn nhà đó.

Bộ dữ liệu và đề tài do nhóm tự phân tích thiết kế và không dựa trên đề tài nào khác. Do đó, để xây dựng mô hình này, chúng tôi đã đặt ra 2 vấn đề cần phải giải quyết: dữ liệu và mô hình. Với dữ liệu, chúng tôi đã tiến hành tự thu thập dữ liệu thực tế trên trang batdongsan.com [1] – trang web về bất động sản lớn nhất Việt Nam. Dữ liệu thu thập được sẽ được đi qua các bước tiền xử lý, phân tích, … để có thể đưa vào mô hình dự đoán. Với mô hình thực hiện, chúng tôi nhận thấy với đặc điểm có nhiều trường thông tin như bộ dữ liệu này, các thuật toán hồi quy sẽ khá phù hợp. Chúng tôi sẽ sử dụng đa dạng các thuật toán để cho ra được mô hình tốt nhất với bài toán này.

Với các bước làm như trên, chúng tôi đã thành công xây dựng mô hình dự đoán này. Kết quả cho ra khá khả quan với độ chính xác là 79,7%, cùng với phép đo để đánh giá hiệu quả của mô hình RMSE đạt 5.21 cho thấy mô hình có chất lượng khá tốt.

2. MÔ TẢ BỘ DỮ LIỆU

Bộ dữ liệu phân tích tự thu thập tại trang batdongsan.com.vn [1]. Trang web này cho phép chúng tôi thu thập trực tiếp những trường thông tin cơ bản của 1 căn hộ như số phòng ngủ, phòng vệ sinh, …. Trên trang web này, chúng tôi chỉ tập trung thu thập thông tin từ 3 loại hình nhà cho thuê: *Căn hộ chung cư, Nhà riêng, Nhà trọ phòng trọ* vì đây là những loại hình phù hợp với nhu cầu của sinh viên chúng tôi.

Với trang web nêu trên, chúng tôi đã sử dụng phương pháp Browser Automation để lấy dữ liệu. Phương pháp này sẽ hạn chế việc chặn IP khi truy cập trang web. Cách làm này yêu cầu chúng tôi phải phân tích cấu trúc html của trang và lấy dữ liệu cần thiết thông qua các element. Do đó, chúng tôi sử dụng thư viện selenium để truy cập vào đường dẫn tới bài đăng chi tiết của mỗi căn hộ và trích xuất thông tin của căn hộ theo cấu trúc trang. Cuối cùng, sẽ lưu thông tin thu được vào file csv. Và đây cũng chính là bộ dữ liệu mà chúng tôi sử dụng cho bài toán này.

Bộ dữ liệu với 21 cột và 1740 dòng. Trong đó có 4 biến số và 17 biến phân loại. Thống kê sơ bộ ban đầu, bộ dữ liệu này vẫn thiếu khá nhiều giá trị ở tất cả các trường, đòi hỏi nhóm phải xử lý dữ liệu một cách kỹ càng. Cụ thể như sau:

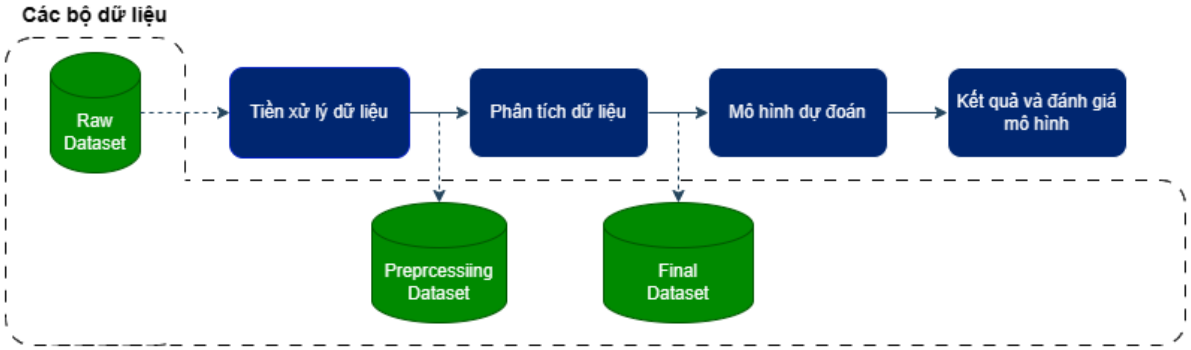
Thuộc tính	Định nghĩa	Kiểu	Ví dụ
post_id	Mã số bài đăng	int64	38205009
post_title	Nội dung bài đăng	object	Cho thuê căn 88m2 (3PN) giá 9.5tr …
post_verified	Tin được xác thực	int64	0
post_type	Loại bài đăng (Tin vip Kim cương, Vàng, Bạc; Tin thường)	object	vip-diamond
upload_date	Thời gian đăng bài	object	06/10/2023
owner_id	Mã số người đăng bài	int64	2524547
owner_name	Tên người đăng bài	object	Thiên Phúc
building_type	Loại bất động sản	object	Căn hộ chung cư
squares	Diện tích	object	88 m²
nums_bedroom	Số phòng ngủ	object	3 phòng
nums_wc	Số nhà vệ sinh	object	2 phòng
nums_floor	Số tầng	object	2 tầng
path_size	Độ rộng đường trước nhà	object	3 m
front_size	Độ rộng mặt tiền	object	2 m
building_direction	Hướng cửa chính	object	Tây - Nam
window_direction	Hướng cửa sổ	object	Đông - Nam
furniture	Nội thất trong nhà	object	Đầy đủ
legal	Giấy tờ	object	Hợp đồng mua bán
others	Thông tin khác	float64	
location	Vị trí căn hộ	object	Dự án The Western Capital, Đường Lý …
price	Giá tiền	object	9,5 triệu/tháng

3. PHƯƠNG PHÁP PHÂN TÍCH

Đánh giá tổng quan ban đầu cho thấy, bộ dữ liệu này tồn tại các khuyết điểm như sau:

- Các trường có kiểu dữ liệu chưa đúng và tồn tại ký tự dư thừa.
- Một số trường bị thiếu dữ liệu nhiều.
- Một số dòng bị lặp lại nội dung do được đăng nhiều lần.
- Các giá trị trong 1 trường có độ phân bổ không đều, gây mất cân bằng dữ liệu.
- Tồn tại một số trường được dự đoán sẽ không ảnh hưởng đến giá thuê nhà, cần được loại bỏ.

Với các vấn đề đặt ra như trên, đòi hỏi chúng tôi phải tiến hành các bước tiền xử lý và phân tích cho bộ dữ liệu này trước khi đưa vào mô hình để tránh trường hợp mô hình học vẹt hoặc dự đoán với độ tin cây thấp. Chúng tôi đã xây dựng luồng xử lý phù hợp với bộ dữ liệu này như sau:



Hình 1. Quy trình phân tích dữ liệu

3.1. Tiền xử lý dữ liệu:

Dựa vào thông tin tổng quan của bộ dữ liệu, có thể thấy thực tế của bộ dữ liệu đòi hỏi chúng tôi phải có các bước xử lý trước khi đưa vào phân tích.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1739 entries, 0 to 1738
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   post_id             1739 non-null   int64
 1   post_title          1739 non-null   object
 2   post_verified       1739 non-null   int64
 3   post_type           1739 non-null   object
 4   upload_date         1739 non-null   object
 5   building_type       1739 non-null   object
 6   squares             1739 non-null   object
 7   nums_bedroom        1325 non-null   object
 8   nums_wc             1301 non-null   object
 9   nums_floor          488 non-null    object
 10  path_size           213 non-null    object
 11  front_size          224 non-null    object
 12  building_direction  259 non-null    object
 13  window_direction    209 non-null    object
 14  furniture           872 non-null    object
 15  legal               484 non-null    object
 16  others              0 non-null      float64
 17  location            1739 non-null   object
 18  owner_id            1739 non-null   int64
 19  owner_name          1739 non-null   object
 20  price               1739 non-null   object
dtypes: float64(1), int64(3), object(17)
memory usage: 285.4+ KB
```

Hình 2. Thông tin tổng quan của bộ dữ liệu

3.1.1. Chuyển đổi kiểu dữ liệu và tách cột:

Để giúp mô hình sau này được học một cách tốt nhất, chúng tôi sẽ tối ưu hóa bộ dữ liệu bằng cách chỉ giữ lại các thuộc tính giá trị rõ ràng và đầy đủ. Dựa vào *Hình 2,* có thể thấy rõ, cột *'others'* không mang giá trị nào cả, vì thế chúng tôi sẽ loại bỏ nó. Bên

cạnh đó, tồn tại một số dòng bị trùng lặp nhau do người đăng đăng nhiều bài cùng nội dung cũng sẽ được chúng tôi loại bỏ.

Khi mô hình học, các kiểu dữ liệu dạng string và object sẽ cần phải qua bước mã hóa, vì thế, chúng tôi sẽ ưu tiên kiểu dữ liệu là int64 hoặc float và chuyển đổi các thuộc tính về kiểu dữ liệu int64 và float này khi có thể. Đối với bộ dữ liệu tự thu thập này, chúng tôi đã chuyển đổi các cột: *'squares'*, *'nums_floor'*, *'nums_wc'*, *'nums_bedroom'* và *'price'* về kiểu dữ liệu số.

Ngoài ra, dữ liệu còn gặp phải tình trạng là 1 thuộc tính lại mang quá nhiều thông tin và sẽ không thể khai thác được nhiều với thuộc tính này. Chúng tôi nhận thấy ở thực tế, giá thuê nhà cao hay thấp còn phụ thuộc khá nhiều vào vị trí của nó. Vì thế, để tận dụng tốt thông tin này, chúng tôi sẽ chia cột *'location'* thành 4 cột là *'project_name'* (tên dự án), *'streets'* (Tên đường), *'wards'* (Tên phường) và *'districts'* (Tên quận).

3.1.2. Xử lý dữ liệu bị thiếu

Sau khi đã tách cột *'location'* ở phía trên, đã xáy ra trường hợp là các cột mới hình thành bị thiếu ở một số bất động sản như nhà riêng sẽ không thuộc dự án nào cả. Vì thế, chúng tôi cũng đã xử lý dữ liệu thiếu của phần này ứng với từng thuộc tính:

- *streets, wards:* dựa vào các bất động sản không bị thiếu, sẽ thu thập thông tin mode của đường và phường, ứng với từng quận sẽ điền giá trị mode này vào và tương tự, ứng với từng phường sẽ điền giá trị của đường tương ứng.
- *projects_name:* với thuộc tính này, chúng tôi sẽ dựa vào cột 'post_title' để tìm kiếm thêm thông tin, nếu trong bài đăng có 1 trong cái tên trong danh sách thu được từ các bất động sản không bị khuyết, chúng tôi sẽ gán giá trị tại cột projects_name là tên của dự án được tìm ra đó. Trường hợp không tìm thấy, chúng tôi sẽ gán giá trị *'Không thuộc dự án'*.

Trong danh sách các dữ liệu bị thiếu ban đầu, chúng tôi sẽ chỉ xử lý các cột *'nums_bedroom', 'nums_wc', 'nums_floor'* và *'funiture'* vì đây là những thuộc tính ảnh hưởng lớn đến giá nhà (theo thực tế). Bên cạnh đó, việc lựa chọn thuộc tính để xử lý nó còn phụ thuộc vào khả năng của nhóm. Cụ thể như sau:

- *part_size và font_size:* không thể xét được đối với loại hình chung cư.
- *building_direction và window_direction:* không thể xét đối với căn hộ, nhà có nhiều cửa số hoặc căn hộ chung cư nhiều mặt tiền.
- *legal:* đa phần chỉ ảnh hưởng đến giá bán, hầu như người thuê nhà sẽ không để ý đến điều này.

Và để xử lý trường hợp thiếu của các thuộc tính trên, ứng với mỗi thuộc tính, chúng tôi đã thực hiện như sau:

- *nums_bedroom, nums_wc:* có thể nhận thấy điểm chung giữa các thông tin này là đa phần đều phụ thuộc vào diện tích nhà/căn hộ. Chúng tôi đã tiến hành chia nhóm diện tích, với mỗi nhóm sẽ tìm mode '*nums_bedroom*', '*num_wc'* của nhóm đó và thay thế vào các cột có giá trị NaN tương ứng của nhà/căn hộ.
- *nums_floor:* chúng tôi đã tiến hành phân tích sơ bộ và thấy được rằng chủ yếu số tầng chỉ có ở loại hình nhà riêng, còn về nhà trọ phòng trọ chỉ có một số có 1 tầng trở lên. Do đó chúng tôi sử dụng cách điền khuyết theo mode của loại diện tích và bất động sản.
- *funiture*: mô tả về nội thất khá đa dạng, để tiện cho việc điền khuyết chúng tôi sẽ gom các nhóm nội thất đó lại thành 5 mốc thường thấy: *'Cơ bản', 'Đầy đủ', 'Cao cấp', 'Có tiện nghi cơ bản'* và *'Không nội thất'*. Sau đó tiến hành điền khuyết theo mode của loại diện tích và bất động sản.

3.1.3. Xóa thuộc tính không cần thiết

Sau khi tiến hành các bước trên, bộ dữ liệu của chúng tôi có 23 thuộc tính và cần phải loại bỏ một số thuộc tính này. Đây là cũng bước quan trọng để tăng chất lượng dữ liệu chúng tôi có trước khi đưa vào mô hình. Chúng tôi sẽ loại bỏ 10 trường thông tin, đây là những trường có dữ liệu thiếu sót như: *part_size, font_size,…* hay các thuộc tính có không mang lại thông tin như *'post_id'*, *'post_title',…*đều sẽ được loại bỏ.

Kết quả cuối cùng, chúng tôi có bộ dữ liệu gồm 14 cột và 1597 dòng, trong đó, không tồn tại bất kỳ giá trị NaN nào.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1597 entries, 0 to 1596
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   post_verified  1597 non-null   int64
 1   post_type      1597 non-null   object
 2   building_type  1597 non-null   object
 3   squares        1597 non-null   float64
 4   nums_bedroom   1597 non-null   int64
 5   nums_wc        1597 non-null   int64
 6   nums_floor     1597 non-null   int64
 7   furniture      1597 non-null   object
 8   price          1597 non-null   float64
 9   project_names  1597 non-null   object
 10  streets        1597 non-null   object
 11  wards          1597 non-null   object
 12  districts      1597 non-null   object
 13  type_squares   1597 non-null   object
dtypes: float64(2), int64(4), object(8)
memory usage: 174.8+ KB
```

Hình 3. Thông tin tổng quan của bộ dữ liệu sau khi tiền xử lý

3.2. Phân tích dữ liệu:

Một bộ dữ liệu tốt chúng tôi không chỉ xem xét về lượng của nó mà còn xem xét về chất. Do đó, chúng tôi cần phải đánh giá các giá trị của mỗi thuộc tính.

	post_verified	squares	nums_bedroom	nums_wc	nums_floor	price
count	1597.00	1597.00	1597.00	1597.00	1597.00	1597.00
mean	0.01	71.67	2.29	2.11	1.69	16.03
std	0.11	69.12	1.66	1.57	1.12	19.87
min	0.00	8.00	1.00	1.00	1.00	1.00
25%	0.00	30.00	1.00	1.00	1.00	5.70
50%	0.00	58.00	2.00	2.00	1.00	10.00
75%	0.00	86.00	3.00	3.00	2.00	18.00
max	1.00	800.00	20.00	20.00	7.00	266.00

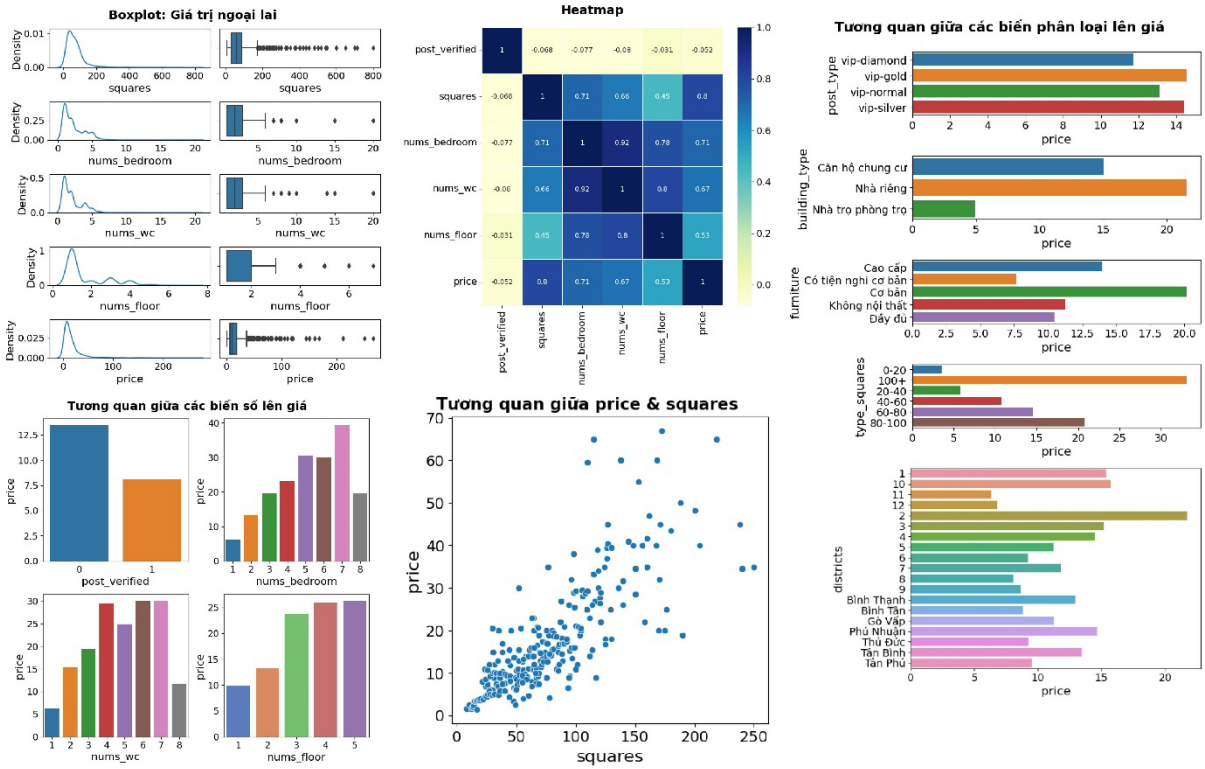
Hình 4. Mô tả dữ liệu

Nhìn vào bảng thống kê trên, có thể thấy với thuộc tính *'squares'* và *'price'* có độ lệch chuẩn khá cao. Điều này là điều dĩ nhiên khi phổ của 2 giá trị này khá rộng với 'squares' dao động từ 8 đến 800, còn 'price' dao động từ 1 đến 266. Vì thế, chênh lệch giữa giá trị mode và đầu cuối khá cao.

Đối với các thuộc tính khác như '*nums_bedroom'*, *'nums_wc'* số lượng phòng đa phần nằm trong khoảng 1-3 phòng, chênh lệch khá lớn so với giá trị max là 20. Nhìn nhận với nums_bedrooms, có thể thấy độ tương quan lớn giữa *'nums_bedroom'* và *'nums_floor'* khi mà số lượng phòng ngủ sẽ bằng hoặc chênh lệch không đánh kể với số lượng nhà vệ sinh.

Đối với thuộc tính *'nums_floor',* số lượng tầng đa phần cũng nằm trong 1-2 tầng. Tuy nhiên, không quá khó hiểu khi mà loại hình chung cư và phòng trọ đã chiếm tới 1/2

tổng số dữ liệu, những loại hình này có số tầng mặc định là 1. Vì thế, chúng tôi sẽ không xét tới trường hợp ngoại lai của thuộc tính này.



Hình 5. Kết quả thăm dò dữ liệu

Để có thể lựa chọn giữ lại hay loại bỏ các giá trị ngoại lai này, chúng tôi sẽ xét tới biểu đồ *Boxplot : Giá trị ngoại lai* để kiểm tra mật độ của nó. Có thể thấy ở cả 4 thuộc tính đều tồn tại giá trị ngoại lai có giá trị lớn nhưng không nhiều. Chúng tôi sẽ loại bỏ các giá trị này.

	post_verified	squares	nums_bedroom	nums_wc	nums_floor	price
count	1542.00	1542.00	1542.00	1542.00	1542.00	1542.00
mean	0.01	62.59	2.17	1.99	1.63	13.43
std	0.11	37.33	1.43	1.29	1.07	11.27
min	0.00	8.00	1.00	1.00	1.00	1.00
25%	0.00	30.00	1.00	1.00	1.00	5.50
50%	0.00	56.00	2.00	2.00	1.00	9.50
75%	0.00	81.00	3.00	2.00	2.00	18.00
max	1.00	250.00	8.00	8.00	5.00	67.00

Hình 6. Dữ liệu sau khi loại bỏ giá trị ngoại lai

Để phân tích được rõ ràng hơn, chúng tôi sẽ sử dụng thêm phương pháp EDA, mục tiêu là tìm ra các feature quan trọng. Để có cái nhìn tổng quan nhất và không bỏ sót bất kỳ thông tin nào, chúng tôi sẽ chia thuộc tính ra 2 loại là biến số và biến phân loại.

Đối với biến số, chúng tôi sử dụng biểu đồ heatmap để xem xét mức độ tương quan giữa các cặp biến trong một bảng dữ liệu. Biến mục tiêu của chúng tôi là *'price'*, do đó,

chúng tôi sẽ tập trung quan sát các thuộc tính với *'price'*. Có thể loại trừ thuộc tính *'post_verified'* vì có mối quan hệ yếu với *'price'*.

Bên canh đó, chúng tôi sử dụng thêm biểu đồi barplot *Tương quan giữa các biến số lên giá.* Thông qua biểu đồ này, chúng tôi đã nhận thấy các biến số so với biến mục tiêu có quan hệ tuyến tính với nhau. Điều này có thể dễ dàng nhận thấy khi giá trị *'nums_bedroom', 'nums_wc'* hay *'nums_floor'* tăng thì 'price' cũng sẽ tăng theo. Đối với giá trị *'squares'* tuy không quá rõ ràng, nhưng vẫn cho thấy dấu hiệu của tuyến tính.

Đối với biến phân loại, chúng tôi cũng sử dụng barplot để biểu diễn. Có thể nhận thấy rằng, các biến phân loại cũng phần nào ảnh hưởng đến giá, điển hình như với district, nhà ở Quận 2 sẽ có giá trung bình cao hơn nhà ở Quận 11.

Tuy nhiên, qua đây chúng tôi vẫn chưa thể đánh giá mức độ ảnh hưởng của tất các biến phân loại, chúng tôi cần một độ đo khác để có thể đánh giá một cách rõ ràng hơn. Phương pháp chúng tôi lựa chọn là ANOVA. Kết quả như sau:

```
Ảnh hưởng của post_type lên price:
F_onewayResult(statistic=1.5099324839606794, pvalue=0.21011259440996502)

Ảnh hưởng của project_names lên price:
F_onewayResult(statistic=3.6123410578681634, pvalue=3.5716499377237383e-35)

Ảnh hưởng của districts lên price:
F_onewayResult(statistic=12.226518418160817, pvalue=5.150241232154634e-34)

Ảnh hưởng của streets lên price:
F_onewayResult(statistic=1.912201171784475, pvalue=4.364601628103122e-18)

Ảnh hưởng của wards lên price:
F_onewayResult(statistic=4.017242755326885, pvalue=3.804746165720014e-33)
```

Hình 7. ANOVA của các biến phân loại lên giá

Dựa vào các giá trị statistic và pvalue của từng thuộc tính, có thể thấy, biến *'post_type'* không ảnh hưởng đáng kể, cho nên chúng tôi sẽ loại bỏ thuộc tính *'post_type'*.

3.3. Mô hình:

Có thể thấy, dữ liệu của chúng tôi là bộ dữ liệu có tính tuyến tính khi giá bán cuối cùng phụ thuộc vào các biến độc lập. Vì thế, nhóm chúng tôi lựa chọn thuật toán Hồi quy nhiều tuyến tính cho bài toán này.

Tuy nhiên, thử nghiệm ban đầu cho ra giá tri không khả quan, đòi hỏi chúng tôi cần phải xử lý thêm cho bô dữ liệu của mình và lựa chọn sử dụng một thuật toán khác cho bài toán này. Với yêu cầu này, chúng tôi đã thử nghiệm thêm các thuật toán khác cũng dùng cho bài toán hồi quy.

Tất cả các thuật toán chúng tôi sử dụng đều yêu cầu dữ liệu số hóa, vì thế các thuộc tính có tính phân loại sẽ được chúng tôi mã hóa: *'building_type', 'furniture', 'project_names', 'streets', 'wards', 'districts'* bằng OneHotEncoder.

3.4. Kết quả và đánh giá mô hình:

Kết quả của các mô hình của chúng tôi như sau:

	model	r2	mse	rmse	mae
0	catboost	7.979083e-01	2.714002e+01	5.209609e+00	3.070878e+00
1	lightGBM	7.834234e-01	2.908527e+01	5.393076e+00	3.297596e+00
2	histogram	7.802012e-01	2.951800e+01	5.433047e+00	3.291720e+00
3	xgboost	7.782595e-01	2.977877e+01	5.456993e+00	3.123120e+00
4	SVR_linear	7.367221e-01	3.535706e+01	5.946180e+00	3.415363e+00
5	SVR_rbf	6.612607e-01	4.549119e+01	6.744716e+00	4.231545e+00
6	SVR_poly	3.106042e-01	9.258280e+01	9.621996e+00	5.097984e+00
7	linear	-7.850790e+17	1.054326e+20	1.026804e+10	1.578550e+09

Hình 8. Độ chính xác của kết quả dự đoán khi training thông qua các mô hình

Chúng tôi đã sử dụng các độ đo R^2 *(Coefficient of determination)* đánh giá mức độ chính xác của mô hình hồi quy. Giá trị R^2 càng cao, cho thấy mô hình dự đoán với độ chính xác cao. Ngoài ra, để có thể xem chất lượng dự đoán của mô hình, chúng tôi đã sử dụng một số độ đo lỗi như MSE *(Mean Squared Error),* RMSE *(Root Mean Squared Error)* hay MAE *(Mean Absolute Error)*.

Dựa vào kết quả được tổng hợp trên, chúng tôi đã rút ra được một số kết luận về mô hình như sau:

- Các mô hình ensemble như CatBoost, LightGBM, Histogram Gradient Boosting và XGBoost cho thấy hiệu suất tốt.
- Mô hình SVR với kernel là linear cũng cho kết quả khả quan.
- Mô hình Linear Regression lại không cho kết quả tốt. Có thể giải thích rằng, với bộ dữ liệu sau khi được mã hóa các biến phân loại lên đến 1519 dòng và 765 cột thì mô hình đơn giản như Linear Regression sẽ không thể nào xử lý đủ tốt.

4. KẾT LUẬN

Để thực hiện đồ án môn học này, chúng tôi đã thực hiện từ những điều cơ bản nhất như thu thập dữ liệu, phân tích và xử lý. Thông qua các bước trên, chúng tôi đã có thêm rất nhiều kinh nghiệm về ý tưởng và cách xử lý khi gặp phải những dữ liệu khó. Bên cạnh đó, việc thực nghiệm các mô hình dự cũng đã giúp chúng tôi có cơ hội được tìm hiểu thêm và nắm rõ các thuật toán hồi quy bên cạnh Linear Regression – thuật toán cơ bản được sử dụng bấy lâu nay.

Đây là lần đầu tiếp cận đến dữ liệu bất động sản, nhưng chúng tôi đã hoàn thành khá tốt với mục tiêu đề ra ban đầu. Dữ liệu sau khi phân tích và xử lý đã được rút gọn và giảm bớt nhiều rất nhiều. Kết quả dự đoán khá cao và khi nhập vào dữ liệu thực tế mô hình cũng đã cho ra giá thuê không mấy chênh lệch với thị trường. Điều này có được là nhờ việc áp dụng tốt những kiến thức đã được học trên lớp.

Tuy nhiên, mô hình này chỉ có thể sử dụng để dự đoán trong khoảng thời gian ngắn (1 – 2 tháng tính từ lúc hoàn thành). Dữ liệu bất động sản là dữ liệu biến động liên tục tùy vào thời điểm. Do đó, đây là 1 nhược điểm của mô hình chúng tôi khi không thể thích ứng với thay đổi thực tế. Chúng tôi sẽ cố gắng hoàn thiện nó nếu có cơ hội trong khoảng thời gian tới.

TÀI LIỆU THAM KHẢO

[1] batdongsan.com.vn. Link: https://batdongsan.com.vn/(Ngày truy cập 10/10/2023).

[2] Thư viện Selenium. Link: https://www.selenium.dev/ (Ngày truy cập 10/10/2023).

[3] Thư viện sklearn. Link: https://scikit-learn.org/stable/index.html (Ngày truy cập 10/10/2023).

[4] Thư viện pandas. Link: https://pandas.pydata.org/ (Ngày truy cập 10/10/2023).

[5] Thư viện numpy. Link: https://numpy.org/ (Ngày truy cập 10/10/2023).

[6] Thư viện seaborn. Link: https://seaborn.pydata.org/ (Ngày truy cập 10/10/2023).

[7] Thư viện LightGBM. Link: https://lightgbm.readthedocs.io/en/latest/index.html# (Ngày truy cập 10/10/2023).

[8] Thư viện CatBoost. Link: https://catboost.ai/ (Ngày truy cập 10/10/2023).

[9] Thư viện XGBoost. Link: https://xgboost.readthedocs.io/en/stable/ (Ngày truy cập 10/10/2023).

[10] Thư viện matplotlib. Link: https://matplotlib.org/ (Ngày truy cập 10/10/2023).

[11] Thư viện Scipy. Link: https://scipy.org/ (Ngày truy cập 10/10/2023).

[12] Thư viện statsmodels. Link: https://www.statsmodels.org/stable/index.html (Ngày truy cập 10/10/2023).

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Nguyễn Minh Thuận	- Thu thập dữ liệu - Tiền xử lí dữ liệu - Viết báo cáo phần 1, 2, 3.1, 1 phần tài liệu tham khảo. - Hoàn thành 100/100
2	Đỗ Thị Thu Trang	- Phân tích dữ liệu - Tìm hiểu và áp dụng các phương pháp phân tích EDA - Viết báo cáo phần 3.2, 4, 1 phần tài liệu tham khảo. - Hoàn thành 100/100
3	Nguyễn Đặng Bảo Ngọc	- Làm slide báo cáo - Huấn luyện mô hình và kiểm thử mô hình - Viết báo cáo phần 3.3, 3.4, 1 phần tài liệu tham khảo. - Hoàn thành 100/100