



Faculty of Information Technology

Môn học : Khoa Học Dữ Liệu
Lớp CQ2016/2 – Học kỳ I/2019-2020
GV: Trần Trung Kiên

BÁO CÁO ĐỒ ÁN MÔN HỌC CUỐI KỲ -- ĐỀ TÀI DỰ ĐOÁN GIÁ NHÀ --

<Vấn đáp với giảng viên ngày 09-01-2020>

Nhóm 14: Đặng Phương Nam – Lê Minh Nghĩa

NỘI DUNG CHÍNH



Câu hỏi đặt ra?



Thu thập dữ liệu



Tiền xử lý
dữ liệu



Mô hình hóa
dữ liệu



Tổng kết



Cho các thông tin về một căn nhà
(đường, khu vực, diện tích, loại nhà, số
phòng, tình hình thuế và tội phạm...)



Căn nhà này có giá là bao nhiêu tiền?

CÂU HỎI ĐẶT RA?





Người bán có thể dự đoán được giá trị căn nhà mà mình muốn bán.



Người mua có thể ước lượng được căn nhà mình muốn mua có giá cả hợp lý hay không?

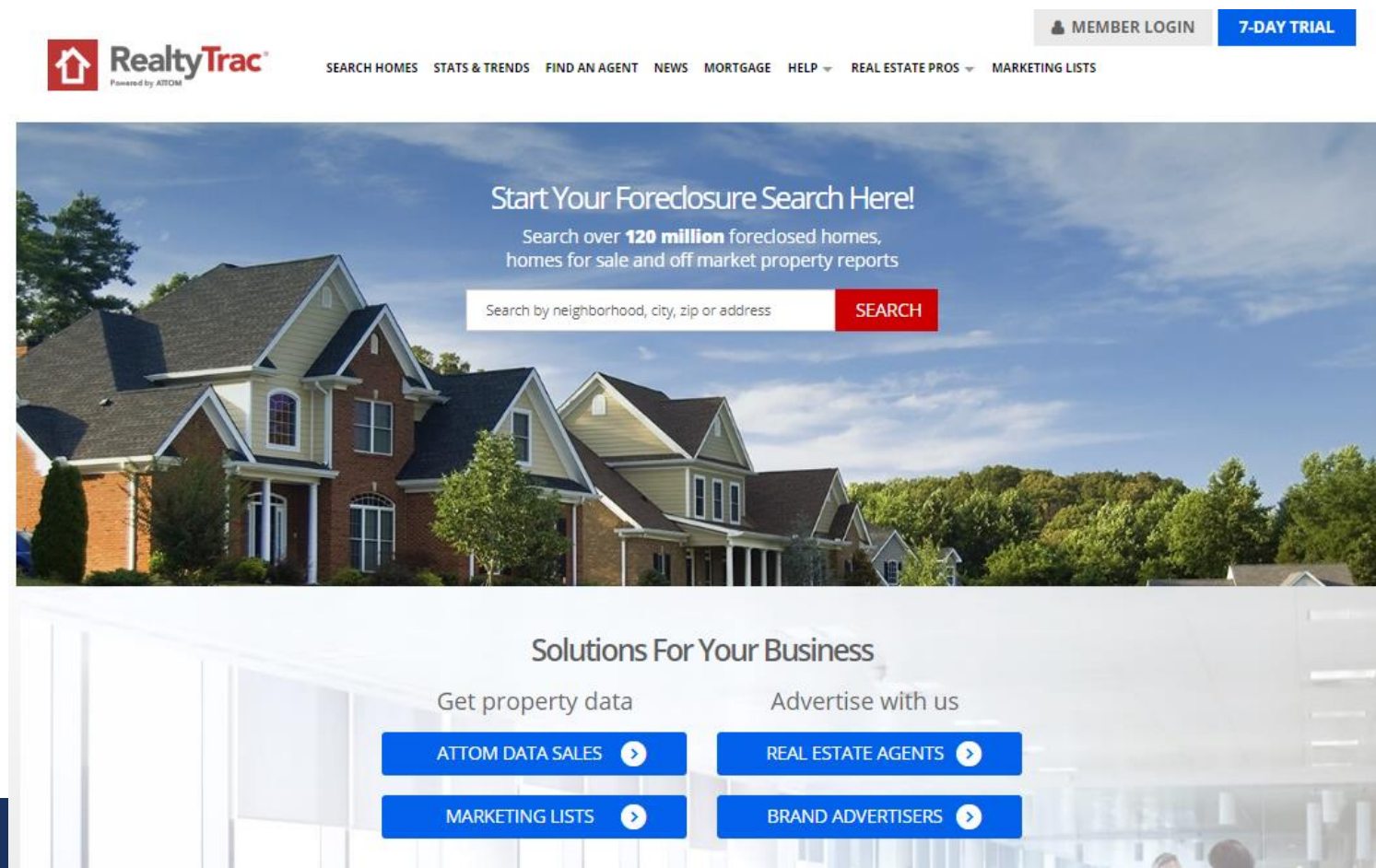


Nguồn gốc câu hỏi: nhóm tự nghĩ ra.

LỢI ÍCH CỦA CÂU HỎI



Trang web thu thập dữ liệu: <https://www.realtytrac.com/>



THU THẬP DỮ LIỆU – PARSE HTML



Thu thập dữ liệu về danh sách “Các căn nhà đã được bán thành công tại quận Cam bang California của Mỹ”

Lưu ý: Dữ liệu mà nhóm thu thập là **hợp pháp** (parse HTML có kiểm tra file robots.txt) và **giá nhà là correct output** (do căn nhà đã bán thành công).

Parse HTML để lấy:

+ Detail url.

Ở mỗi lần parse, nhóm đều thực kiểm tra việc lấy dữ liệu có hợp pháp hay không tại file robots.txt của trang web:

<https://m.realtytrac.com/robots.txt>

[Home](#) > [Sold](#) > [California](#) > [Orange County](#)

Orange County, CA Recently Sold Properties

View sold price, comparable sales and detailed property information on 19,468 properties in Orange County, CA.

515 Pecan Ave, Huntington Beach, CA 92648



SOLD

Sold Price: **\$1,450,000**

3 Beds | 3.5 Baths | 2,387 Sq/Ft

Sold: 12/19/2019

[View Details](#)

17 Long Bay Dr, Newport Beach, CA 92660



SOLD

Sold Price: **\$1,998,000**

3 Beds | 3.5 Baths | 2,917 Sq/Ft

Sold: 12/19/2019

[View Details](#)

3116 W Olinda Ln, Anaheim, CA 92804



SOLD

Sold Price: **\$625,000**

3 Beds | 2 Baths | 1,318 Sq/Ft

Sold: 12/19/2019

[View Details](#)

Ở mỗi bản tin bán nhà thành công, nhóm muốn lấy thêm một số thông tin chi tiết:

Parse HTML để lấy:

- + Date Sold.
- + Mortgage.
- + Address.

+ Description.

+ Property Infor:

- Bedroom.
- Bathroom.
- Size.
-

Ở mỗi lần parse, nhóm đều thực kiểm tra việc lấy dữ liệu có hợp pháp hay không tại file robots.txt của trang web:

<https://m.realtytrac.com/robots.txt>

SOLD ON 11/25/2019

\$1,360,000 (sold price)

Est. Mortgage: \$6,178/mo

240 Colgate Ave

Kensington, CA 94708

4 bd | 2 ba | 1,478 sq ft

8101 Barrington Dr is a single family residence located in La Mirada, CA 90638. Built in 1960, this property features 3 bedrooms, 3 bathrooms, 7,200 sq ft lot, and 1,627 sq ft of living space. This property recently sold for \$605,000 on 12/19/2019.

For the surrounding community of La Mirada, CA 90638, the average sale price for similar homes to 8101 Barrington Dr is \$628,885. The nearby schools are excellent and include Charles G Emery Elementary, Sunny Hills High and Buena Park Junior High. The overall crime risk for this area is low with 3 criminal and sex offenders residing within 1 mile. The natural disaster risk for this area includes very high earthquake risk, low tornado risk, and minimal flood risk.

Property Info

Type	Single Family Residence
Bedrooms	4
Bathrooms	2
Size	1,478 sqft
Lot Size	5,000 sqft
Year Built	1938
Sold Price	\$1,360,000
Property ID	1102358090
County	Contra Costa County, CA
Parcel Number	5701800079

Ở mỗi bản tin bán nhà thành công, nhóm muốn lấy thêm một số thông tin chi tiết:

Parse HTML để lấy:

+ Taxes.

+ Price History.

+ Number school.

Property Taxes

Land	\$32,498
Improvements	\$27,856
Total	\$60,354
Taxes	\$2,173 (3.60 %)

Price History

Sold (11/25/19)	\$1,360,000
Sold (3/29/19)	\$680,000

Local Schools



Kensington Elementary (assigned)

90 Highland Blvd, Kensington, CA 94708

Students: 514

Grades: KG-06



El Cerrito High (assigned)

540 Ashbury Ave, El Cerrito, CA 94530

Students: 1364

Grades: 09-12



Fred T Korematsu Middle (assigned)

1021 Navellier St, El Cerrito, CA 94530

Students: 539

Grades: 07-08

Ở mỗi lần parse, nhóm đều thực kiểm tra việc lấy dữ liệu có hợp pháp hay không tại file robots.txt của trang web:

<https://m.realtytrac.com/robots.txt>

Ở mỗi bản tin bán nhà thành công, nhóm muốn lấy thêm một số thông tin chi tiết:

Parse HTML để lấy:

+ Info crime.

+ Number near foreclosures.

Ở mỗi lần parse HTML ở mỗi trang nhóm đều thực kiểm tra việc lấy dữ liệu có hợp pháp hay không tại file robots.txt của trang web:

<https://m.realtytrac.com/robots.txt>

Local Crime Index



Total Crime = 25



Violent Crime = 16



Property Crime = 47

National average = 100

Nearby Foreclosures

000

Kenilworth Dr
Kensington, CA 94707
2 Bd 1 Ba 972 SqFt
Auction - \$784,389
[View Details](#)

000

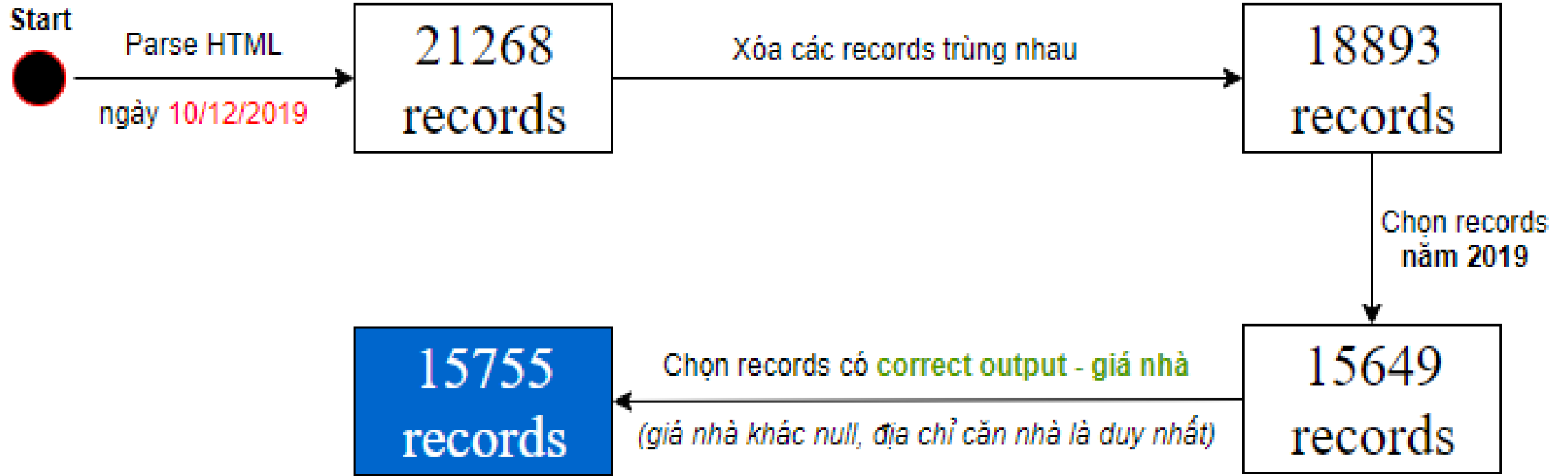
Grizzly Peak Blvd
Berkeley, CA 94708
2 Bd 2 Ba 1,664 SqFt
Pre-foreclosure - \$1,333,000
[View Details](#)

000

Coventry Rd
Kensington, CA 94707
3 Bd 3 Ba 1,923 SqFt
Pre-foreclosure - \$1,441,000
[View Details](#)

000

Menlo Pl
Berkeley, CA 94707
3 Bd 3.5 Ba 2,458 SqFt
Pre-foreclosure - \$1,604,000
[View Details](#)



THU THẬP DỮ LIỆU – LỰA CHỌN DỮ LIỆU NĂM 2019



Dữ liệu có 15649 dòng và 26 cột

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15649 entries, 0 to 15648
Data columns (total 26 columns):
address_street      15603 non-null object
address_locality    15649 non-null object
address_region      15649 non-null object
address_code        15649 non-null int64
date_sold           15649 non-null object
mortgage            15649 non-null float64
info_type            15649 non-null object
info_bedrooms       14222 non-null float64
info_bathrooms      14232 non-null float64
info_size            14937 non-null float64
info_lot_size        10605 non-null float64
info_year_built     14461 non-null float64
info_sold_price      15649 non-null float64
info_property_id    15649 non-null int64
info_county          15649 non-null object
info_parcel_number   15649 non-null int64
taxes_land           15408 non-null float64
taxes_improvements   15408 non-null float64
taxes_total          15408 non-null float64
taxes_taxes          15408 non-null object
school              15649 non-null int64
total_crime          15166 non-null float64
violent_crime        15166 non-null float64
property_crime       15166 non-null float64
foreclosures         15649 non-null int64
year_sold            15649 non-null int64
dtypes: float64(13), int64(6), object(7)
memory usage: 3.1+ MB
```

1. address_street: tên đường.
2. address_locality: tên địa phương.
3. address_region: tên vùng.
4. address_code: mã bưu điện.
5. date_sold: ngày bán thành công ngôi nhà.
6. mortgage: định giá ngôi nhà cho thuê theo tháng trước khi bán đi.
7. info_type: loại nhà.
8. info_bedrooms: số lượng phòng ngủ.
9. info_bathrooms: số lượng phòng tắm.
10. info_size: kích thước ngôi nhà (sqft).
11. info_lot_size: kích thước lô đất (sqft).
12. info_year_build: năm ngôi nhà được xây dựng.
- 13. info_sold_price: giá bán thành công.**
14. info_property_id: id căn nhà.
15. info_county: tên quận.
16. info_parcel_number: số bưu kiện của căn nhà.

17. taxes_land: tiền đất.
18. taxes_improvements: tiền các tiện ích có trên đất đó (hàng rào, đường đi từ cổng vào nhà, ...).
19. taxes_total: tổng của tiền land và tiền improvements.
20. taxes_taxes: tiền thuế phải đóng ứng với taxes_total.
21. school: số lượng trường học gần đó.
22. total_crime: tỉ lệ tội phạm (%) so với tội phạm cả nước.
23. violent_crime: tỉ lệ tội phạm bạo lực (%) so với tội phạm cả nước.
24. property_crime: tỉ lệ tội phạm về tài sản (%) so với tội phạm cả nước.
25. foreclosures: số lượng các căn nhà bị tịch thu gần đó.
26. year_sold: năm bán nhà.

Có 15 cột có giá trị thiếu là:

- address_street thiếu 46 giá trị.
- info_bedrooms thiếu 1427 giá trị.
- info_bathrooms thiếu 1417 giá trị.
- info_size thiếu 712 giá trị.
- info_lot_size thiếu 1334 giá trị.
- info_year_built thiếu 5044 giá trị.
- info_sold_price thiếu 934 giá trị.
- info_property_id thiếu 151 giá trị.
- taxes_land, taxes_improvements, taxes_total và taxes_taxes đều thiếu 241 giá trị.
- total_crime, violent_crime và property_crime đều thiếu 483 giá trị.

Lý do có nhiều giá trị thiếu tại các cột bên là do người muốn bán nhà lúc đăng tin không nhập đầy đủ các trường thông tin trên mà thường chỉ mô tả chi tiết tại phần description của bản tin hay vì lý do nào đó mà bản thân người bán cũng không biết rõ các thông tin trên.

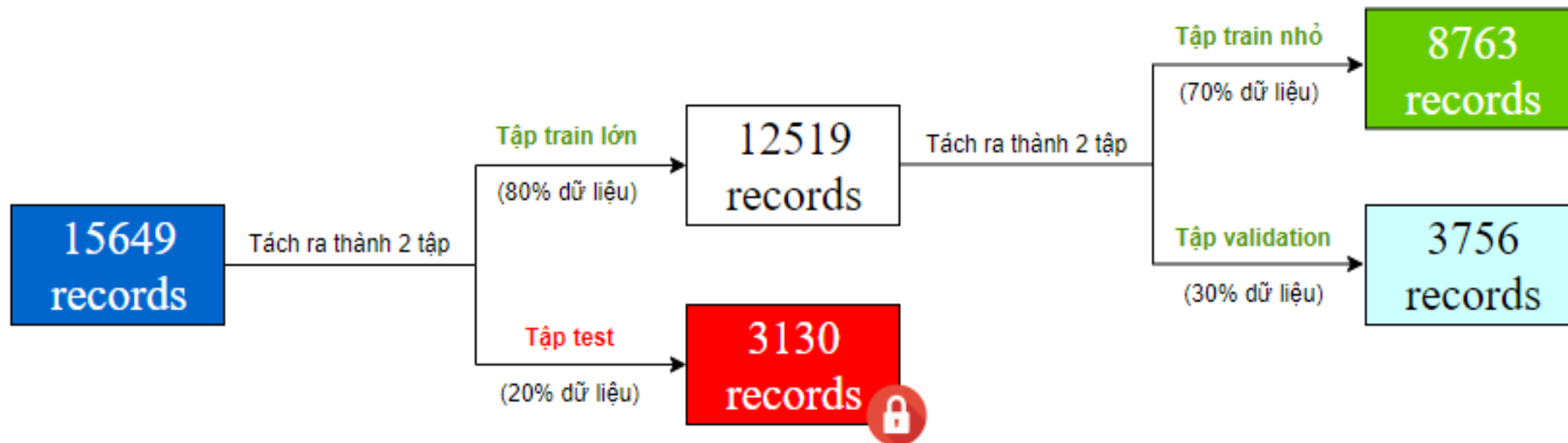
Giải pháp: Sẽ bỏ đi các cột thực sự không cần thiết mà có nhiều giá trị thiếu. Và phần giá trị thiếu còn lại, nhóm sẽ tiếp tục thử tìm trong phần description của mỗi bản tin từ phần dữ liệu đã lấy được ban đầu. Nếu vẫn còn thiếu thì sẽ lấp đầy bằng các phương pháp như mean, mode, median, KNN.



	address_street	address_locality	address_region	address_code	date_sold	mortgage	info_type	info_bedrooms	info_bathrooms	info_size	...	taxes_land
0	4 Silver Crk # 30	Irvine	CA	92603	11/22/2019	3543.0	Condominium	3.0	2.0	1576.0	...	211708.0
1	28459 Alava	Mission Viejo	CA	92692	11/22/2019	2430.0	Single Family Residence	2.0	2.0	1503.0	...	281600.0
2	19441 Hansen Ln	Huntington Beach	CA	92646	11/22/2019	6268.0	Single Family Residence	4.0	3.0	2926.0	...	459321.0
3	23308 Copante # 103	Mission Viejo	CA	92692	11/22/2019	2158.0	Condominium	2.0	2.0	1308.0	...	326318.0
4	1301 Burwood St	La Habra	CA	90631	11/22/2019	3816.0	Multi-Family Dwellings	NaN	NaN	3756.0	...	233089.0
5	10281 Overhill Dr	Santa Ana	CA	92705	11/22/2019	7268.0	Single Family Residence	5.0	3.0	2814.0	...	525268.0
6	52 Wild Horse	Irvine	CA	92602	11/22/2019	5669.0	Single Family Residence	4.0	3.0	2393.0	...	360108.0
7	8597 Valley View St	Buena Park	CA	90620	11/22/2019	2839.0	Single Family Residence	3.0	2.0	1250.0	...	437818.0
8	13887 La Jolla Plz	Garden Grove	CA	92844	11/22/2019	2226.0	Single Family Residence	3.0	3.0	1637.0	...	332109.0
9	163 Jaripol Cir	Rancho Mission Viejo	CA	92694	11/22/2019	1881.0	Condominium	1.0	2.0	921.0	...	NaN

THU THẬP DỮ LIỆU – 10 DÒNG ĐẦU TIÊN CỦA DỮ LIỆU ĐƯỢC THU THẬP





Mọi thao tác tiền xử lý đều được thực hiện trên **tập train nhỏ (8763 records)**, sau đó mới dùng các giá trị ước lượng từ **tập train nhỏ** để tiền xử lý **tập validation (3756 records)**.

Cuối cùng, áp dụng toàn bộ thao tác tiền xử lý tìm được (*từ ở trên*) áp dụng tương tự cho **tập train lớn** và **tập test (3130 records)**.

TIỀN XỬ LÝ DỮ LIỆU





Phần description thu thập được không giúp ích gì cho việc điền giá trị thiếu trong dữ liệu.

1. Cột address_street được ghi theo format:

<số nhà> <hướng của đường> <tên đường> <chung cư (unit, #)>

ví dụ: 2875 S Fairview St Unit B

Nhóm chỉ lấy giá trị có nằm trong chung cư hay không? (1 là có, 0 là không). Xóa cột address_street.

2. Xóa 2 cột address_locality và cột address_region, giữ lại cột address_code.

3. Từ cột date_sold: rút trích time_sold (là month hay season) ra và xóa đi cột date_sold.

4. Để nguyên cột mortgage.

5. Cột info_type: sử dụng biến num_top_types để xử lý như BT03.

6. Giữ nguyên các cột info_bedroom, info_bathroom, info_size và info_lot_size.

7. Cột info_year_build vẫn giữ lại hay chuyển qua giai đoạn: <1900, [1990; 1950), [1950; 2000), >2000.

8. Xóa 3 cột info_property_id, info_county và info_parcel_number vì chúng khác biệt 100% và không có ý nghĩa khi huấn luyện.

9. Giữ nguyên hai cột taxes_land, taxes_improvements. Xóa cột taxes_total. Rút trích tiền từ taxes_taxes vì nó chứa cả thông tin dạng tiền và %, *ví dụ: 6969 (1.23 %)*.

10. Giữ nguyên các cột school, foreclosures, total_crime, violent_crime và property_crime.

11. Xóa cột year_sold vì tất cả đều là 2019.

Các dạng cột:

- Cột **dạng số (numerical)**, có 15 cột: apt_unit, mortgage, info_bedroom, info_bathroom, info_size, info_lot_size, taxes_land, taxes_improvements, taxes_taxes, school, foreclosures, total_crime, violent_crime và property_crime.
- Cột **dạng chuỗi có giá trị rời rạc không thứ tự (categorical)**, có 3 cột: address_code, time_sold và info_type. Các cột này đưa vào huấn luyện sẽ sử dụng phương pháp one-hot để chuyển thành số.

Điền giá trị thiếu:

- Dùng **KNN (n = 5)** cho các cột: info_size và info_lot_size.
- Dùng **most** cho các cột: address_code, time_sold, info_type, info_bedroom và info_bathroom.
- Dùng **mean** cho các cột: mortgage, taxes_land, taxes_improvements và taxes_taxes.
- Dùng **median** cho các cột: school, foreclosures, total_crime, violent_crime và property_crime.

Các siêu tham số cần lưu ý:

- **month_to_season**: có chuyển tháng bán nhà thành bán nhà theo bốn mùa? (True/False).
- **year_to_period**: có chuyển thời gian xây nhà thành các khoảng thời gian mà nhóm cho trước? (True/False).
- **num_top_types**: số lượng top types như BT03 (số nguyên dương).

Các siêu tham số này cũng sẽ được học trong mô hình máy học.



Điều bất thường là khi thử sử dụng mô hình Linear Regression (*month_to_seasons=True*, *year_to_period=True* và *num_top_types=5*) để dự đoán giá nhà thì thu được kết quả với **độ chính xác trên tập validation cực cao 99.99%**.

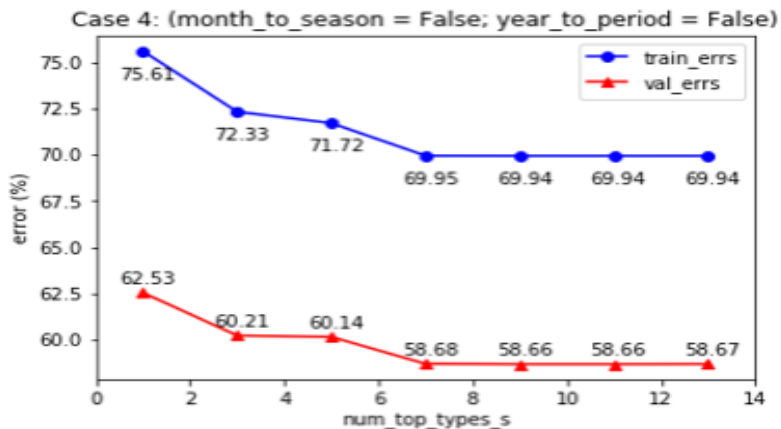
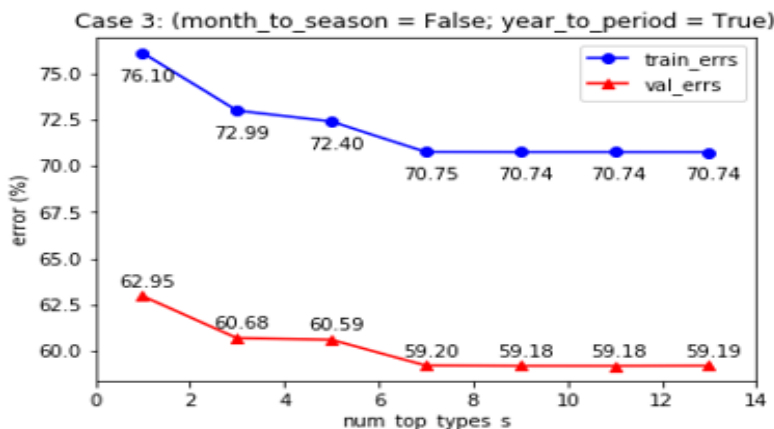
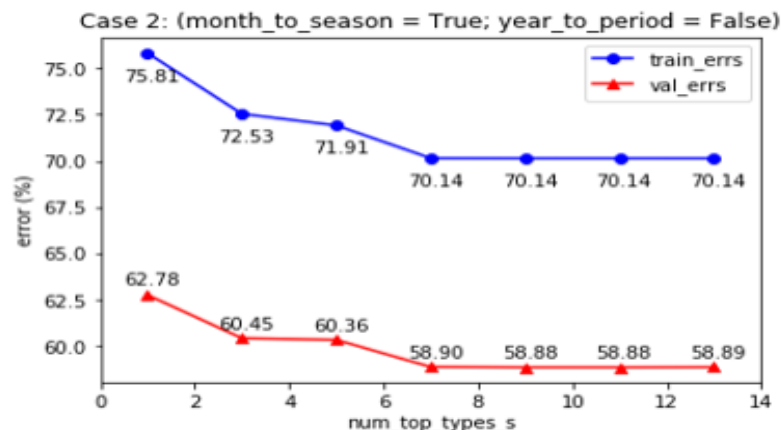
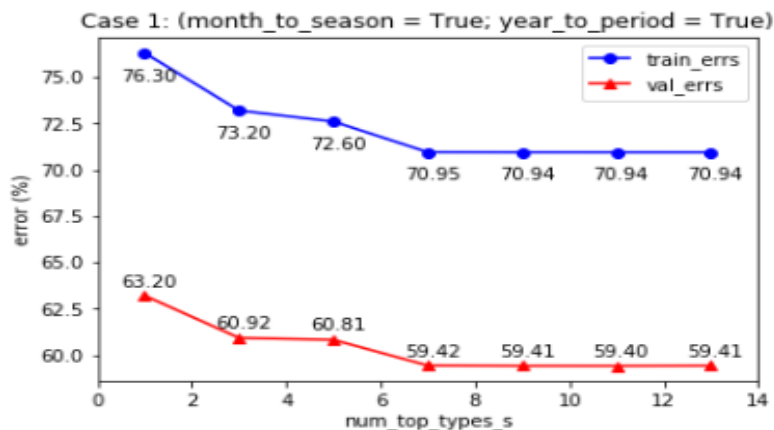
Phát hiện thấy vấn đề nằm ở cột mortgage, vì sau một vài lần thử thì thấy rằng: ta chỉ cần lấy **mortgage nhân cho 220** thì sẽ ra được con số rất gần với giá nhà được bán thành công.



Bây giờ sẽ xóa đi cột mortgage và sử dụng lại mô hình Linear Regression (*month_to_seasons=True*, *year_to_period=True* và *num_top_types=5*) để dự đoán giá nhà thì thu được kết quả với **độ chính xác trên tập validation chỉ 36.76%**.

MÔ HÌNH HÓA DỮ LIỆU





Mô hình **Linear Regression** quá yếu trong tình huống này nên dự đoán bị underfitting, khi tăng chiều dữ liệu lên (đó là cho các siêu tham số bằng False, tăng số lượng num_top_types) thì thấy giúp học tốt hơn một chút, nhưng vẫn bị underfitting.

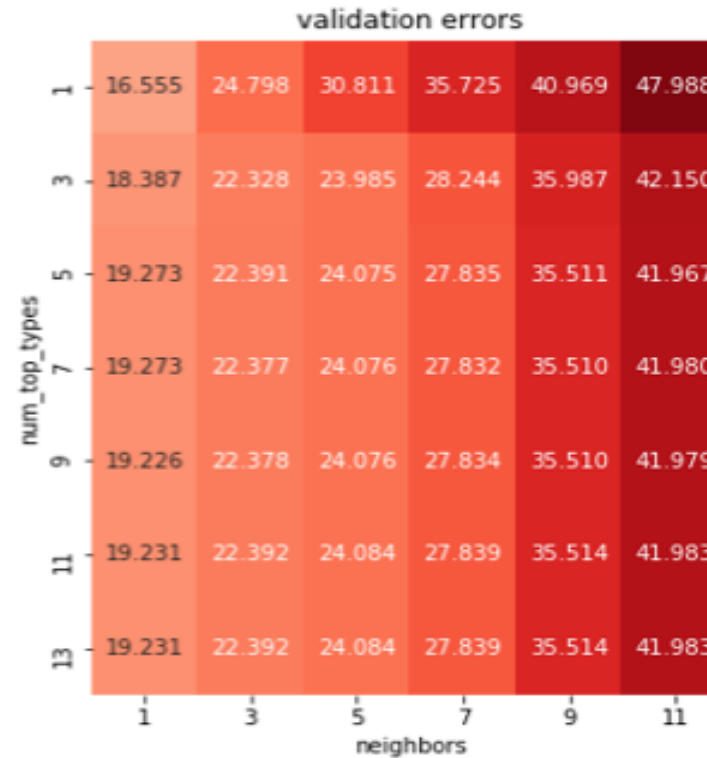
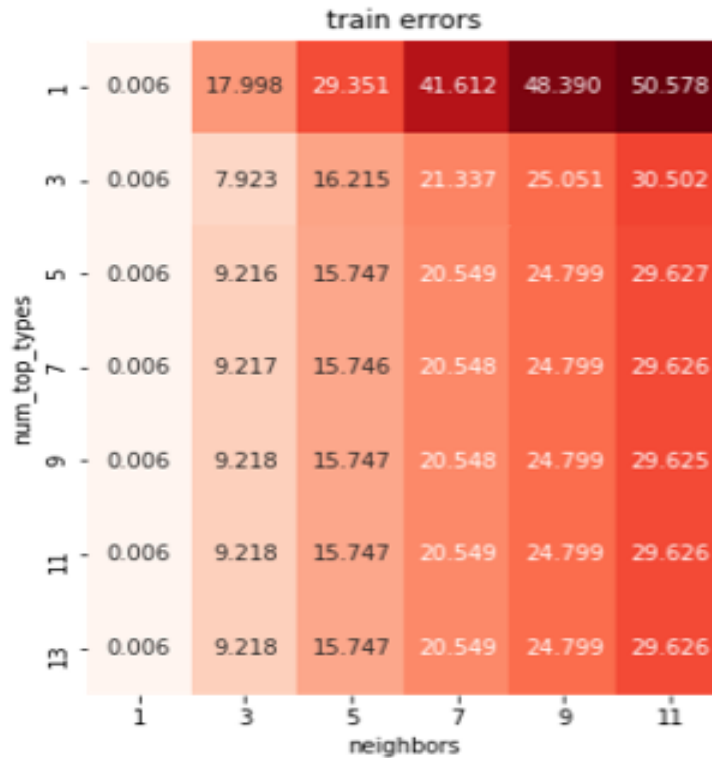
Độ lỗi trên tập validation thấp nhất là **58.66%**, với month_to_season = False, year_to_period = False và num_top_types = 11.

MÔ HÌNH HÓA DỮ LIỆU – LINEAR REGRESSION



K-Neighbors Regressor

Case 1: *month_to_seasons = True* và *year_to_period = True*



- *neighbors* càng tăng thì khiến mô hình càng bị underfitting.

- *num_top_types* càng tăng thì càng giúp giảm độ lỗi trên cả 2 tập, nhưng đến một ngưỡng nào đó thì tăng thêm độ lỗi trên 2 tập vẫn vậy.

Case 1: độ lỗi trên tập validation thấp nhất là **16.56%**, với *num_top_types = 1* và *neighbors = 1*

MÔ HÌNH HÓA DỮ LIỆU – K-NEIGHBORS REGRESSOR



Nhóm cũng tiến hành thử nghiệm cho 3 trường hợp còn lại với tập giá trị `num_top_types = [1, 3, 5, 7, 9, 11, 13]` và `neighbors = [1, 3, 5, 7, 9, 11]`:

- **Case 2:** `month_to_season = True` và `year_to_period = False`, độ lỗi thấp nhất thu được ở tập validation là **16.47%** ứng với `num_top_types = 3` và `neighbors = 1`.
- **Case 3:** `month_to_season = False` và `year_to_period = True`, độ lỗi thấp nhất thu được ở tập validation là **15.71%** ứng với `num_top_types = 1` và `neighbors = 1`.
- **Case 4:** `month_to_season = False` và `year_to_period = False`, độ lỗi thấp nhất thu được ở tập validation là **16.47%** ứng với `num_top_types = 3` và `neighbors = 1`.

Trong trường hợp dùng **K-Neighbors Regressor** fit vào tập train và dự đoán trên tập validation thì thấy rằng: càng tăng neighbors thì càng khiến mô hình dễ bị underfitting do có quá nhiều láng giềng gây nhiễu ảnh hưởng đến kết quả.

Ghi nhận độ lỗi thấp nhất trên tập validation khi dùng mô hình **K-Neighbors Regressor** là **15.71%** ứng với các siêu tham số `month_to_season = False`, `year_to_period = True`, `num_top_types = 1` và `neighbors = 1`.

Decision Tree Regressor

Case 1: *month_to_seasons = True* và *year_to_period = True*

		train errors					
num_top_types	1	38.960	4.929	2.186	0.758	0.192	0.104
	3	12.406	6.419	4.244	1.641	0.262	0.083
	5	12.406	6.419	4.244	1.640	0.261	0.081
	7	12.406	6.419	4.243	1.640	0.259	0.081
	9	12.406	6.419	4.244	1.640	0.261	0.082
	11	12.406	6.419	4.244	1.640	0.261	0.082
	13	12.406	6.419	4.244	1.640	0.261	0.082
		3	5	7	9	11	13
		max_depth					

		validation errors					
num_top_types	1	35.375	33.723	33.263	33.614	34.363	34.326
	3	10.685	9.994	10.077	5.143	9.702	10.592
	5	23.204	9.994	9.567	5.473	7.411	21.290
	7	10.685	9.994	10.279	4.986	9.733	8.689
	9	10.685	23.228	22.597	4.930	9.446	7.959
	11	10.685	9.994	9.376	5.249	10.323	8.447
	13	10.685	10.784	9.376	5.772	9.254	7.724
		3	5	7	9	11	13
		max_depth					

- *num_top_types* bằng 1
→ *max_depth* quá nhỏ bị underfitting, về sau có tăng *max_depth* lên ở giá trị bao nhiêu thì cũng lại bị overfitting.

- *num_top_types* tăng dần lên:
→ càng tăng *max_depth* thì tỏ ra có hiệu quả nhưng tăng một hồi sẽ có dấu hiệu bị overfitting.

- *num_top_types* ban đầu tăng thì giúp giảm độ lỗi trên cả 2 tập, một lúc sau sẽ bị overfitting và sau đó chỉ làm giảm rồi lại tăng độ lỗi trên tập validation (xen kẽ).

Case 1: độ lỗi trên tập validation thấp nhất là **4.93%**, với *num_top_types* = 9 và *max_depth* = 9

MÔ HÌNH HÓA DỮ LIỆU – DECISION TREE REGRESSOR



Nhóm cũng tiến hành thử nghiệm cho 3 trường hợp còn lại với tập giá trị `num_top_types` = [1, 3, 5, 7, 9, 11, 13] và `neighbors` = [1, 3, 5, 7, 9, 11]:

- **Case 2:** `month_to_season` = `True` và `year_to_period` = `False`, độ lỗi thấp nhất thu được ở tập validation là 6.04% ứng với `num_top_types` = 7 và `max_depth` = 11.
- **Case 3:** `month_to_season` = `False` và `year_to_period` = `True`, độ lỗi thấp nhất thu được ở tập validation là 5.14% ứng với `num_top_types` = 13 và `max_depth` = 9.
- **Case 4:** `month_to_season` = `False` và `year_to_period` = `False`, độ lỗi thấp nhất thu được ở tập validation là 6.37% ứng với `num_top_types` = 7 và `max_depth` = 11.

Trong trường hợp dùng **Decision Tree Regressor** để fit vào tập train và dự đoán trên tập validation thì thấy rằng: càng tăng `max_depth` thì mô hình càng fit vào dữ liệu do đó có thể dẫn đến trường hợp bị overfitting. Lưu ý việc dùng `num_top_types` khó kiểm soát nhưng vẫn giúp mô hình học hiệu quả.

Ghi nhận độ lỗi thấp nhất trên tập validation khi dùng mô hình **Decision Tree Regressor** là 4.93% ứng với các siêu tham số `month_to_season` = `True`, `year_to_period` = `True`, `num_top_types` = 9 và `max_depth` = 9.

Random Forest Regressor

Case 1: *month_to_seasons = True* và *year_to_period = True*

		train errors					
num_top_types	1	27.793	7.018	4.481	3.715	3.178	3.743
	3	12.750	7.148	4.302	3.643	3.405	3.439
	5	12.693	7.520	4.574	3.228	2.959	3.030
	7	12.695	7.169	4.456	3.282	3.024	2.977
	9	12.657	7.141	4.643	3.500	3.223	3.045
	11	12.699	7.609	4.372	3.456	3.073	3.161
	13	12.663	7.272	4.491	3.211	3.583	2.922
		3	5	7	9	11	13
		max_depth					

		validation errors					
num_top_types	1	31.741	26.488	20.572	23.363	28.585	22.535
	3	9.860	8.766	8.063	7.823	7.791	8.240
	5	9.674	9.084	8.095	7.289	6.837	7.089
	7	9.626	9.013	8.368	7.544	7.000	7.230
	9	9.580	8.472	8.287	7.543	7.655	8.409
	11	10.275	9.890	8.060	8.055	7.089	7.932
	13	10.213	8.811	8.068	6.865	7.755	7.222
		3	5	7	9	11	13
		max_depth					

- Độ lỗi bên tập train tăng (giảm) thì độ lỗi bên tập validation cũng tăng (giảm) theo.

- Còn sự thay đổi của *num_top_types* và *max_depth* thì có biểu hiện tương tự như bên mô hình Decision Tree Regressor.

Case 1: độ lỗi trên tập validation thấp nhất là **6.84%**, với *num_top_types* = 5 và *max_depth* = 11

MÔ HÌNH HÓA DỮ LIỆU – RANDOM FOREST REGRESSOR



Nhóm cũng tiến hành thử nghiệm cho 3 trường hợp còn lại với tập giá trị `num_top_types` = [1, 3, 5, 7, 9, 11, 13] và `neighbors` = [1, 3, 5, 7, 9, 11]:

- **Case 2:** `month_to_season` = `True` và `year_to_period` = `False`, độ lỗi thấp nhất thu được ở tập validation là 5.91% ứng với `num_top_types` = 5 và `max_depth` = 11.
- **Case 3:** `month_to_season` = `False` và `year_to_period` = `True`, độ lỗi thấp nhất thu được ở tập validation là 6.96% ứng với `num_top_types` = 7 và `max_depth` = 9.
- **Case 4:** `month_to_season` = `False` và `year_to_period` = `False`, độ lỗi thấp nhất thu được ở tập validation là 5.98% ứng với `num_top_types` = 3 và `max_depth` = 13.

Trong trường hợp dùng **Random Forest Regressor** để fit vào tập train và dự đoán trên tập validation thì thấy rằng: sự ảnh hưởng của các siêu tham số đến độ lỗi trên 2 tập là khá tương tự như trong mô hình **Decision Tree Regressor**, chỉ khác ở đây là dùng nhiều cây (mặc định 10 cây) và độ lỗi trên tập train tăng (giảm) thì độ lỗi trên tập validation cũng tăng (giảm) theo.

Ghi nhận độ lỗi thấp nhất trên tập validation khi dùng mô hình **Random Forest Regressor** là 5.91% ứng với các siêu tham số `month_to_season` = `True`, `year_to_period` = `False`, `num_top_types` = 5 và `max_depth` = 11.

Bảng thống kê Validation Error nhỏ nhất của các mô hình hồi quy được sử dụng trong Project.

Model	month_to_season	year_to_period	num_to_types	neighbors	max_depth	Validation Error
Linear Regression	False	False	11	None	None	58.66%
K-Neighbors Regressor	False	True	1	1	None	15.71%
Decision Tree Regressor	True	True	9	None	9	4.93%
Random Forest Regressor	True	False	5	None	11	5.91%



Mô hình chiến thắng là Decision Tree Regressor với độ lỗi trên tập validation là 4.93%.

MÔ HÌNH HÓA DỮ LIỆU – MÔ HÌNH TỐT NHẤT



Decision Tree
Model

fit vào toàn bộ tập Train

month_to_seasons = True, year_to_period = True
num_top_type = 9, max_depth = 9

Toàn bộ tập Train

12519
records

Dự đoán và đánh giá trên tập Test

3130
records

Tập test

Độ lỗi thu được trên tập test là **5.13%**.

MÔ HÌNH HÓA DỮ LIỆU – DỰ ĐOÁN TRÊN TẬP TEST

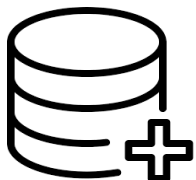




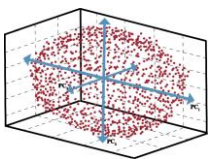
Tìm hiểu kỹ dữ liệu thu thập và các thuộc tính để chọn đưa vào huấn luyện.



Bắt đầu thực hiện đồ án sớm, mỗi ngày một ít.



Thu thập thêm dữ để tăng độ chính xác khi dự đoán.



Thử dùng thêm các kỹ thuật giảm chiều dữ liệu như PCA xem như thế nào.

TỔNG KẾT – BÀI HỌC RÚT RA VÀ MONG MUỐN





THANK YOU FOR WATCHING