|  |  |
| --- | --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **VŨ ĐỨC ĐỘ**  **KHÓA 14**  **HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **CHUYÊN NGÀNH: CÔNG NGHỆ DỮ LIỆU**  **NGHIÊN CỨU VÀ ỨNG DỤNG HỌC MÁY**  **CHO ĐỊNH GIÁ NHÀ**  **NĂM 2020** | |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **VŨ ĐỨC ĐỘ**  **KHÓA 14**  **HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**  **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH: CÔNG NGHỆ THÔNG TIN**.  **MÃ SỐ: 52480201**  **NGHIÊN CỨU VÀ ỨNG DỤNG HỌC MÁY**  **CHO ĐỊNH GIÁ NHÀ**  ***Cán bộ hướng dẫn: 2//, GV, TS. Trần Cao Trưởng***  **NĂM 2020** | |
| HỌC VIỆN KỸ THUẬT QUÂN SỰ  **KHOA CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN AN TOÀN THÔNG TIN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc** |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên: Vũ Đức Độ, Lớp: CNDL14, Khóa: 14

Ngành: Công nghệ thông tin, Chuyên ngành: Công nghệ dữ liệu.

1. Tên đề tài: Nghiên cứu và ứng dụng học máy cho định giá nhà.

2. Các số liệu ban đầu:

- Quyết định Giao đồ án tốt nghiệp đại học – Học viện KTQS

- Tài liệu tham khảo

3. Nội dung bản thuyết minh:

- Mở đầu

- Chương 1: Tổng quan về học máy

- Chương 2: Tổng quan về học máy cho định giá nhà

- Chương 3: Học máy cho định giá nhà riêng tại Hà Nội

- Chương 4: Xây dựng chương trình định giá

- Kết luận

- Tài liệu tham khảo

4. Số lượng, nội dung các bản vẽ (ghi rõ loại, kích thước và cách thực hiện các bản vẽ) và các sản phẩm cụ thể (nếu có):

Được sử dụng máy tính và máy chiếu để trình chiếu.

5. Cán bộ hướng dẫn:

- Họ và tên: Trần Cao Trưởng

- Cấp bậc: Trung tá

- Học hàm, học vị: GV, TS

- Đơn vị: Bộ môn Khoa học máy tính - Khoa Công nghệ thông tin

- Hướng dẫn toàn bộ

Ngày giao: 10/01/2020 Ngày hoàn thành: 28/07/2020

*Hà Nội, ngày 28, tháng 07, năm 2020*

**Chủ nhiệm bộ môn Cán bộ hướng dẫn**

(Ký, ghi rõ họ tên, học hàm, học vị)

**Học viên thực hiện**

Đã hoàn thành và nộp đồ án ngày 28 tháng 07 năm 2020

(Ký và ghi rõ họ tên)

**KÝ HIỆU VIẾT TẮT**

|  |  |
| --- | --- |
| **Nội dung** | **Ký hiệu viết tắt** |
| CSDL | Cơ sở dữ liệu |
| DL | Dữ liệu |
| KNN | K-nearest neighbors |
| KPDL | Khai phá dữ liệu |
| SVM | Support Vector Machine |
| SVR | Support Vector Machine - Regression |
| URL | Uniform Resource Locator |

**DANH MỤC HÌNH ẢNH**

[Hình 1.1. Công thức tính khoảng cách trong KNN 9](#_Toc46711178)

[Hình 1.2. Công thức Hamming 9](#_Toc46711179)

[Hình 1.3. SVM chia bộ dữ liệu trong không gian 2 chiều 10](#_Toc46711180)

[Hình 1.4. SVM chia bộ dữ liệu trong không gian 3 chiều 11](#_Toc46711181)

[Hình 1.5. Khoảng cách margin 11](#_Toc46711182)

[Hình 1.6. Độ lệch chuẩn cho 1 thuộc tính 16](#_Toc46711183)

[Hình 1.7. Độ lệch chuẩn cho 2 thuộc tính 17](#_Toc46711184)

[Hình 2.1. Quy trình khai phá dữ liệu 20](#_Toc46711185)

[Hình 3.1. Giao diện trang web 29](#_Toc46711186)

[Hình 3.2. Trang cuối không có dữ liệu 30](#_Toc46711187)

[Hình 3.3. Dữ liệu được lọc trên website 31](#_Toc46711188)

[Hình 3.4. Chi tiết tin đăng 31](#_Toc46711189)

[Hình 3.5. Đặc điểm bất động sản 32](#_Toc46711190)

[Hình 3.6. Ví dụ một document được lưu trong collection URL 33](#_Toc46711191)

[Hình 3.7. Ví dụ một document được lưu trong collection rawData 35](#_Toc46711192)

[Hình 3.8. Người dùng ghi thông tin vô lý 36](#_Toc46711193)

[Hình 3.9. Người dùng ghi sai thông tin giá so với mô tả 37](#_Toc46711194)

[Hình 3.10. Quy luật trong cách viết địa chỉ 38](#_Toc46711195)

[Hình 3.11. Thống kê số lượng bản ghi theo từng quận/huyện 39](#_Toc46711196)

[Hình 3.12. Thống kê tỷ lệ missing value của các thuộc tính 42](#_Toc46711197)

[Hình 3.13. Biểu đồ thống kê giá trị thuộc tính Price 44](#_Toc46711198)

[Hình 3.14. Biểu đồ thống kê giá trị thuộc tính Price 44](#_Toc46711199)

[Hình 3.15. Hình so sánh tỷ lệ sai số các mô hình 52](#_Toc46711200)

[Hình 4.1. Giao diện chương trình demo API định giá 72](#_Toc46711201)

[Hình 4.2. Hiển thị kết quả định giá 73](#_Toc46711202)

**DANH MỤC BẢNG BIỂU**

[Bảng 3.1. Các thông tin lưu trữ trong collection URL 33](#_Toc46711203)

[Bảng 3.2. Các thông tin lưu trữ trong collection rawData 34](#_Toc46711204)

[Bảng 3.3. Liệt kê các thuộc tính thu thập được 35](#_Toc46711205)

[Bảng 3.4. Danh sách các thuộc tính có thể đưa vào mô hình 40](#_Toc46711206)

[Bảng 3.5. Bảng thống kê giá trị một số thuộc tính 43](#_Toc46711207)

[Bảng 3.6. Thống kê giá trị lớn nhất, nhỏ nhất của các thuộc tính sau khi lọc theo phân vị ở thuộc tính giá và diện tích 46](#_Toc46711208)

[Bảng 3.7. Thống kê giá trị lớn nhất, giá trị nhỏ nhất của các thuộc tính 47](#_Toc46711209)

[Bảng 3.8. Mô hình sử dụng mode để impute missing value 51](#_Toc46711210)

[Bảng 3.9. Mô hình sử dụng mean để impute missing value 51](#_Toc46711211)

[Bảng 3.10. Mô hình sử dụng median để impute missing value 51](#_Toc46711212)

[Bảng 3.11. Kết quả mô hình sử dụng Z-score standardizes chuẩn hóa dữ liệu 52](#_Toc46711213)

[Bảng 3.12. Kết quả mô hình khi lược bỏ thuộc tính area 53](#_Toc46711214)

[Bảng 3.13. Kết quả mô hình khi lược bỏ thuộc tính làn\_certificate 54](#_Toc46711215)

[Bảng 3.14. Kết quả mô hình khi lược bỏ thuộc tính street\_width 54](#_Toc46711216)

[Bảng 3.15. Thống kê dữ liệu sau khi lọc nhiễu theo từng quận/ huyện 56](#_Toc46711217)

[Bảng 3.16. Kết quả mô hình quận Hà Đông 58](#_Toc46711218)

[Bảng 3.17. Kết quả mô hình quận Nam Từ Liêm 58](#_Toc46711219)

[Bảng 3.18. Kết quả mô hình quận Hoài Đức 59](#_Toc46711220)

[Bảng 3.19. Kết quả mô hình quận Đống Đa 59](#_Toc46711221)

[Bảng 3.20. Kết quả các mô hình sử dụng thuật toán Random Forest 60](#_Toc46711222)

[Bảng 3.21. Bảng so sánh kết quả giữa mô hình toàn Hà Nội và mô hình từng quận/ huyện 61](#_Toc46711223)

[Bảng 3.22. Thống kê kết quả sau khi hiệu chỉnh tham số 63](#_Toc46711224)

[Bảng 3.23. Kết quả định giá một số bản ghi tại quận Ba Đình 64](#_Toc46711225)

[Bảng 3.24. Kết quả định giá một số bản ghi tại quận Đống Đa 66](#_Toc46711226)

[Bảng 3.25. Kết quả sau khi bổ sung lọc nhiễu 67](#_Toc46711227)

[Bảng 4.1. Các thuộc tính đầu vào của API định giá 69](#_Toc46711228)

[Bảng 4.2. Danh sách các tham số output của API định giá 70](#_Toc46711229)

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc46711230)

[1. Lý do chọn đề tài 1](#_Toc46711231)

[2. Cơ sở khoa học, tính thực tiễn đề tài 1](#_Toc46711232)

[3. Mục đích nghiên cứu 2](#_Toc46711233)

[4. Đối tượng và phạm vi nghiên cứu 2](#_Toc46711234)

[5. Nhiệm vụ nghiên cứu 2](#_Toc46711235)

[6. Phương pháp nghiên cứu 2](#_Toc46711236)

[Chương 1 4](#_Toc46711237)

[TỔNG QUAN VỀ HỌC MÁY 4](#_Toc46711238)

[1.1 Giới thiệu về Học máy 4](#_Toc46711239)

[1.2 Phân loại các phương pháp học máy 5](#_Toc46711240)

[1.1.1. Phân loại dựa trên phương thức học 5](#_Toc46711241)

[1.1.2. Phân nhóm dựa theo chức năng 6](#_Toc46711242)

[1.3 Học máy cho bài toán hồi quy 7](#_Toc46711243)

[1.1.3. Thuật toán Linear Regression 7](#_Toc46711244)

[1.1.4. Thuật toán KNN 8](#_Toc46711245)

[1.1.5. Thuật toán SVR 10](#_Toc46711246)

[1.1.6. Thuật toán Decision Tree 13](#_Toc46711247)

[1.1.7. Thuật toán Random Forest 17](#_Toc46711248)

[1.4 Tổng kết chương 18](#_Toc46711249)

[Chương 2 19](#_Toc46711250)

[TỔNG QUAN VỀ HỌC MÁY CHO ĐỊNH GIÁ NHÀ 19](#_Toc46711251)

[2.1 Nghiên cứu bài báo định giá 19](#_Toc46711252)

[2.2 Qui trình ứng dụng học máy cho định giá nhà 19](#_Toc46711253)

[2.2.1 Quy trình khai phá dữ liệu 20](#_Toc46711254)

[2.2.2 Quy trình xây dựng mô hình định giá 21](#_Toc46711255)

[2.3 Giới thiệu về cơ sở dữ liệu của bài toán định giá nhà 25](#_Toc46711256)

[2.4 Tổng quan về các mô hình cho định giá nhà 26](#_Toc46711257)

[Chương 3 28](#_Toc46711258)

[HỌC MÁY CHO ĐỊNH GIÁ BÁN NHÀ RIÊNG TẠI HÀ NỘI 28](#_Toc46711259)

[3.1 Thu thập dữ liệu 28](#_Toc46711260)

[3.1.1 Nguồn gốc dữ liệu 28](#_Toc46711261)

[3.1.2 Cách thức thu thập dữ liệu 29](#_Toc46711262)

[3.1.3 Mô tả dữ liệu thô 35](#_Toc46711263)

[3.2 Tiền xử lý dữ liệu 37](#_Toc46711264)

[3.3 Khám phá dữ liệu 39](#_Toc46711265)

[3.4 Xây dựng mô hình học máy cho định giá nhà 45](#_Toc46711266)

[3.4.1 Lọc dữ liệu nhiễu 45](#_Toc46711267)

[3.4.2 Impute missing value 47](#_Toc46711268)

[3.4.3 Feature selection 48](#_Toc46711269)

[3.4.4 Chuẩn hóa dữ liệu 48](#_Toc46711270)

[3.4.5 Lựa chọn thuật toán 48](#_Toc46711271)

[3.4.6 Xây dựng các mô hình học máy 49](#_Toc46711272)

[3.5 Tổng kết và đánh giá kết quả 68](#_Toc46711273)

[Chương 4 69](#_Toc46711274)

[XÂY DỰNG CHƯƠNG TRÌNH ĐỊNH GIÁ 69](#_Toc46711275)

[4.1 Phân tích yêu cầu 69](#_Toc46711276)

[4.2 Xây dựng API định giá 69](#_Toc46711277)

[4.3 Xây dựng chương trình minh họa 71](#_Toc46711278)

[KẾT LUẬN 74](#_Toc46711279)

[1. Kết luận 74](#_Toc46711280)

[2. Hướng phát triển 74](#_Toc46711281)

[TÀI LIỆU THAM KHẢO 75](#_Toc46711282)

# MỞ ĐẦU

## Lý do chọn đề tài

Thị trường bất động sản là nơi thực hiện các giao dịch mua bán, chuyển nhượng, trao đổi bất động sản và các quyền về bất động sản. Đây là một trong những thị trường quan trọng trong nền kinh tế quốc dân bởi bất động sản không chỉ là các tài sản có giá trị rất lớn về mặt kinh tế mà còn là loại tài sản có ý nghĩa xã hội sâu sắc.

Đó là một mặt hàng có giá trị rất lớn những thông thường để định giá chính xác được nó thì cần đền chuyên giá có nhiều năm kinh nghiệm. Chính vì thế giá trị của bất động sản thương bị các đối tượng lợi dụng cố tình đẩy giá lên cao hoặc ép giá xuống thấp nhằm chuộc lợi gây thiệt hại to lớn cho cả người bán lần người mua. Để giải quyết vẫn đề này cần có một công cụ giúp cho cả người bán và người mua có thể định giá được căn nhà của mình.

Vi thế đề tài **“Nghiên cứu và ứng dụng học máy cho định giá nhà”** được ra đời nhằm nghiên cứu các giải pháp công nghệ sử dụng các kĩ thuật học máy để có thể dự báo một cách tương đối nhất giá trị của căn nhà. Điều này giúp cho việc định giá trở lên công bằng hơn giữa người bán với người mua, giữa các chủ đầu tư, môi giới, khách hàng hay với ngân hàng khi định giá một căn nhà.

## Cơ sở khoa học, tính thực tiễn đề tài

Cơ sở khoa học:

* Cơ sở lý thuyết về các phương pháp học máy.

Ý nghĩa thực tiễn đề tài:

Theo Tiêu chuẩn thẩm định giá Việt Nam số 02 Giá trị thị trường làm cơ sở cho thẩm định giá [1] ta có định nghĩa:

"Giá trị thị trường là mức giá ước tính của tài sản tại thời điểm, địa điểm thẩm định giá, giữa một bên là người mua sẵn sàng mua và một bên là người bán sẵn sàng bán, trong một giao dịch khách quan, độc lập, có đủ thông tin, các bên tham gia hành động một cách có hiểu biết, thận trọng và không bị ép buộc."

Như vậy ta có thể thấy giá trị của một sản phẩm nói chung và giá trị của căn nhà nói riêng là một thỏa thuận giữa bên bán và bên mua, điều này làm cho giá trị căn nhà có tính tương đối và rất khó xác định. Việc này dẫn đến hệ lụy là người mua có thể bị mua với giá cao và người bán có thể bán với giá thấp hơn giá trị thực sự của căn nhà.

Hiện nay, với sự phát triển của công nghệ đặc biệt là các công nghệ trí tuệ nhân tạo, học máy hoàn toàn có thể giải quyết được những khó khăn này bằng cách đưa ra dự báo giá nhà một cách khách quan và chân thực nhất.

Với việc áp dụng công nghệ vào việc dự báo giá sẽ giúp cho người bán có thể đưa ra một mức giá phù hợp nhất tránh tình trạng người bán đưa ra mức giá qua cao dẫn đến không tìm được người mua, còn người mua sẽ mua được căn nhà với mức giá phải chăng.

## Mục đích nghiên cứu

* Tìm hiểu một số phương pháp học máy ứng dụng trong các bài toán định giá nhà.
* Xây dựng bộ dữ liệu cho định giá nhà riêng trên địa bàn một số quận/ huyện Hà Nội.

## Đối tượng và phạm vi nghiên cứu

Đối tượng và phạm vi nghiên cứu của đề tài là nghiên cứu, thử nghiệm các phương pháp học máy áp dụng cho định giá nhà sử dụng bộ dữ liệu nhà riêng tại Hà Nội.

## Nhiệm vụ nghiên cứu

* Xây dựng bộ dữ liệu cho định giá nhà riêng tại Hà Nội.
* Xây dựng mô hình định giá nhà riêng tại một số quận, huyện Hà Nội.

## Phương pháp nghiên cứu

* Phương pháp nghiên cứu tài liệu:
* Đọc, tìm hiểu các bài báo về định giá nhà trên thế giới.
* Đọc, tìm hiều cơ bản về học máy, các phương pháp sử dụng cho định giá, các bước thực hiện đề tài.
* Tìm hiểu các công cụ lập trình hỗ trợ cho đề tài.
* Phương pháp nghiên cứu thực hành: Lập trình thử nghiệm, đánh giá kết quả.

# Chương 1

# TỔNG QUAN VỀ HỌC MÁY

## Giới thiệu về Học máy

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, …, chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

Học máy ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): machine learning) là một tập con của AI. Theo định nghĩa học máy là một lĩnh vực của [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại [thư điện tử](https://vi.wikipedia.org/wiki/Th%C6%B0_%C4%91i%E1%BB%87n_t%E1%BB%AD) xem có phải [thư rác (spam)](https://vi.wikipedia.org/wiki/Th%C6%B0_r%C3%A1c) hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với [suy diễn thống kê](https://vi.wikipedia.org/w/index.php?title=Suy_di%E1%BB%85n_th%E1%BB%91ng_k%C3%AA&action=edit&redlink=1) (statistical inference) tuy có khác nhau về thuật ngữ.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning (Học Sâu). Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc

## Phân loại các phương pháp học máy

Có hai cách phổ biến phân nhóm các thuật toán Machine learning. Một là dựa trên phương thức học (learning style), hai là dựa trên chức năng (function) (của mỗi thuật toán).

### Phân loại dựa trên phương thức học

Theo phương thức học, các thuật toán Machine Learning thường được chia làm 4 nhóm: Supervised learning, Unsupervised learning, Semi-supervised lerning và Reinforcement learning. Có một số cách phân nhóm không có Semi-supervised learning hoặc Reinforcement learning.

Supervised Learning (Học có giám sát) là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning. Trong học có giám sát có hai loại thuật toán phổ biến nhất là:

* + Classification (phân loại): giá trị của output cần dự đoán là các giá trị rời rạc, ví dụ như màu sắc, đúng/sai... Ví dụ về việc phân loại spam/non-spam email là 1 bài toán Classification.
  + Regression (hồi quy): giá trị của output cần dự đoán là các giá trị thực và liên tục, ví dụ như giá tiền, cân nặng, số lượng... Một số thuật toán phổ biến trong bài toán hồi quy là Linear Regression, Logistic Regression. Nội dung của đề tài chính là một bài toán hồi quy.

Unsupervised Learning (Học không giám sát): Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Semi-Supervised Learning (Học bán giám sát):Các bài toán khi chúng ta có một lượng lớn dữ liệu X nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.

Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học chẳng hạn). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.

Reinforcement Learning (Học Củng Cố)

Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

### Phân nhóm dựa theo chức năng

Dựa vào sự tương đồng về chức năng hay cách thức hoạt động mà các thuật toán sẽ được gom nhóm với nhau. Sau đây là danh sách các nhóm và các thuật toán theo từng nhóm.

Dưới đây là một số thuật toán được phân chia theo nhóm chức năng:

* Các thuật toán hồi quy (Regression Algorithms) : đây là nhóm các thuật toán dùng trong các bài toán hồi quy. Ví dụ một số thuật toán:
* [Linear Regression](https://machinelearningcoban.com/2016/12/28/linearregression/)
* [Logistic Regression](https://machinelearningcoban.com/2017/01/27/logisticregression/#sigmoid-function)
* Các thuật toán phân lớp (Classification Algorithms) : đây là nhóm các thuật toán dùng trong các bài toán phân lớp. Ví dụ một số thuật toán:
* Linear Classifier
* [Support Vector Machine (SVM)](https://machinelearningcoban.com/2017/04/09/smv/)
* [Kernel SVM](https://machinelearningcoban.com/2017/04/22/kernelsmv/)
* Sparse Representation-based classification (SRC)
* Thuật toán cây quyết định (Decision Tree Algorithms) : Đây là phương pháp xây dựng mô hình ra quyết định dựa trên các giá trị thực của những thuộc tính trong dữ liệu. Sự quyết định được rẽ nhánh trong cấu trúc cây cho đến khi quyết định dự đoán được đưa ra cho một mẫu nhất định. Phương pháp này được sử dụng trong việc huấn luyện dữ liệu cho bài toán phân lớp và hồi quy. Vì sự nhanh chóng, chính xác nên phương pháp này rất được ưa chuộng trong ML. Các thuật toán cây quyết định phổ biến nhất bao gồm:
* Chi-squared Automatic Interaction Detection (CHAID)
* Classification và Regression Tree – CART
* Conditional Decision Trees
* C4.5 và C5.0
* Thuật toán Bayes (Bayesian Algorithms): Đây là nhóm các thuật toán áp dụng Định lý Bayes cho bài toán phân loại và hồi quy. Một vài thuật toán phổ biến là:
* Averaged One-Dependence Estimators (AODE)
* Bayesian Belief Network (BBN)
* Bayesian Network (BN)
* Gaussian Naive Bayes
* Multinomial Naive Bayes
* Naive Bayes

## Học máy cho bài toán hồi quy

Một số thuật toán học máy áp dụng cho bài toán hồi quy như : linear regression, KNN, SVR, Decision Tree, Random forest

### Thuật toán 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. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục.

Ví dụ: Một căn nhà rộng , có phòng ngủ và cách trung tâm thành phố  km có giá là bao nhiêu. Giả sử chúng ta đã có số liệu thống kê từ 1000 căn nhà trong thành phố đó, liệu rằng khi có một căn nhà mới với các thông số về diện tích, số phòng ngủ và khoảng cách tới trung tâm, chúng ta có thể dự đoán được giá của căn nhà đó không? Nếu có thì hàm dự đoán sẽ có dạng như thế nào. Ở đây là một vector hàng chứa thông tin *input*, ***y*** là một số vô hướng (scalar) biểu diễn *output* (tức giá của căn nhà trong ví dụ này).

Một cách đơn giản nhất, chúng ta có thể thấy rằng:

* Diện tích nhà càng lớn thì giá nhà càng cao;
* Số lượng phòng ngủ càng lớn thì giá nhà càng cao;
* Càng xa trung tâm thì giá nhà càng giảm.

Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa giá nhà và 3 đại lượng đầu vào là:

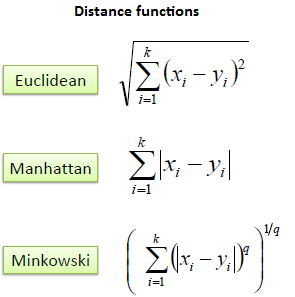
trong đó :

* là các hằng số,
* được gọi là bias.
* Mối quan hệ  bên trên là một mối quan hệ tuyến tính (linear).

Bài toán chúng ta đang làm là một bài toán thuộc loại regression. Bài toán đi tìm các hệ số tối ưu

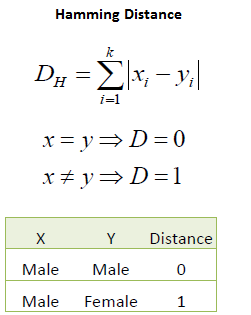
### Thuật toán KNN

Một cách thực hiện đơn giản của hồi quy KNN là tính trung bình của mục tiêu số của K lân cận gần nhất. Một cách tiếp cận khác sử dụng trung bình khoảng cách nghịch đảo của K lân cận gần nhất. Hồi quy KNN sử dụng các hàm khoảng cách giống như phân loại KNN.



Hình .. Công thức tính khoảng cách trong KNN

Ba biện pháp khoảng cách trên chỉ có giá trị cho các biến liên tục. Trong trường hợp các biến phân loại, bạn phải sử dụng khoảng cách Hamming, đây là thước đo số lượng phiên bản trong đó các ký hiệu tương ứng khác nhau ở hai chuỗi có độ dài bằng nhau.



Hình .. Công thức Hamming

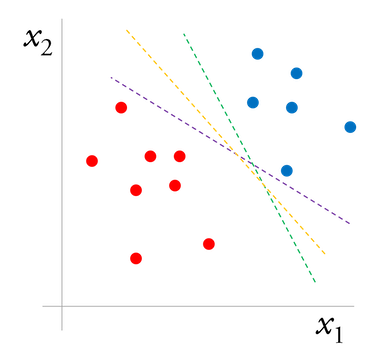
Chọn giá trị tối ưu cho K được thực hiện tốt nhất bằng cách kiểm tra dữ liệu trước tiên. Nói chung, giá trị K lớn chính xác hơn vì nó làm giảm nhiễu tổng thể; tuy nhiên, sự thỏa hiệp là các ranh giới riêng biệt trong không gian tính năng bị mờ. Xác thực chéo là một cách khác để xác định lại giá trị K tốt bằng cách sử dụng bộ dữ liệu độc lập để xác thực giá trị K của bạn. K tối ưu cho hầu hết các bộ dữ liệu là 10 trở lên. Điều đó tạo ra kết quả tốt hơn nhiều so với 1-NN.

### Thuật toán SVR

Thuật toán SVR - Support Vector Regression là một mở rộng của thuật toán SVM – Support Vector Machine sử dụng trong các bài toán hổi quy (regression).

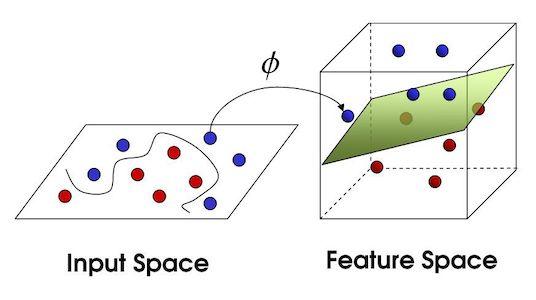
Ý tưởng của SVM là tìm một siêu phẳng (hyper lane) để phân tách các điểm dữ liệu. Siêu phẳng này sẽ chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một loại giữ liệu.

Hình dung ta có bộ data gồm các điểm xanh và đỏ đặt trên cùng một mặt phẳng.Ta có thể tìm được đường thẳng để phân chia riêng biệt các bộ điểm xanh và đỏ như hình bên dưới.



Hình .. SVM chia bộ dữ liệu trong không gian 2 chiều

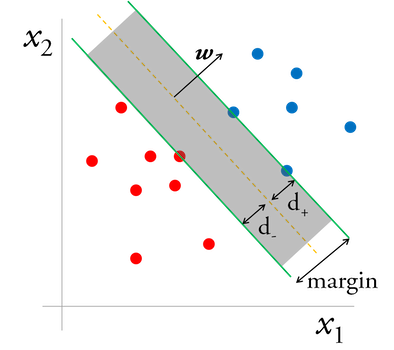
Với những bộ data phức tạp hơn ta cần dùng thuật toán để ánh xạ bộ data đó vào không gian nhiều chiều hơn (n chiều), từ đó tìm ra siêu mặt phẳng (hyperplane) để phân chia.Ví dụ trong hình bên dưới là việc ánh xạ tập data từ không gian 2 chiều sang không gian 3 chiều.



Hình .. SVM chia bộ dữ liệu trong không gian 3 chiều

Quay lại bài toán với không gian 2 chiều ta có thể thấy đường tối ưu là đường tạo cho ta có cảm giác 2 lớp dữ liệu nằm cách xa nhau và cách xa đường đó nhất. Để tính toán được tối ưu này trong SVM có sử dụng thuật ngữ Margin.

**Margin** là khoảng cách giữa siêu phẳng (trong trường hợp không gian 2 chiều là đường thẳng) đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp.



Hình .. Khoảng cách margin

SVM cố gắng tối ưu thuật toán bằng các tìm cách maximize giá trị margin này, từ đó tìm ra siêu phẳng đẹp nhất để phân 2 lớp dữ liệu.

Bài toán của chúng ta trở thành tìm ra 2 đường biên của 2 lớp dữ liệu (ở hình bên trên là 2 đường xanh lá cây) sao cho khoảng cách giữa 2 đường này là lớn nhất. Đường biên của lớp xanh sẽ đi qua một (hoặc một vài) điểm xanh.  
Đường biên của lớp đỏ sẽ đi qua một (hoặc một vài) điểm đỏ.  
Các điểm xanh, đỏ nằm trên 2 đường biên được gọi là các support vector, vì chúng có nhiệm vụ support để tìm ra siêu phẳng. Đó cũng là lý do của tên gọi thuật toán Support Vector Machine.

**Cách tính Margin:**

Trong bài toán không gian 2 chiều, ta giả sử đường thẳng phân chia cần tìm có phương trình là:

Giả sử 2 đường thẳng đi qua các support vector của 2 lớp dữ liệu lần lượt là:

Với không gian 2 chiều Margin giữa 2 đường thẳng được tính bằng công thức:

Tổng quát lên không gian nhiều chiều, cần tìm phương trình siêu phẳng có phương trình:

Khi đó margin được tính bằng công thức:

Bài toán tìm Margin cực đại được giải bằng cách giải bài toán đối ngẫu Lagrange (Lagrange dual problem).

Để tránh **overfitting**, nhiều khi để muốn có margin cao, ta chấp nhận việc một vài data có thể không được chia chính xác (ví dụ như 1 bóng xanh bị lọt sang vùng của bóng đỏ). Data này được gọi là nhiễu. Margin trong trường hợp này gọi là **Soft Margin**. **Hard Margin** ám chỉ việc tìm được Margin mà không nhiễu (tất cả các data đều thỏa mãn sự phân chia).

Với các bài toán thực tế, việc tìm được **Hard Margin** nhiều khi là bất khả thi, vì thế việc chấp nhận sai lệch ở một mức độ chấp nhận được là vô cùng cần thiết.

Trong cài đặt SVM, người ta giới thiệu tham số với quy ước:

= Không cho phép sai lệch, đồng nghĩa với **Hard Margin**.

lớn : Cho phép sai lệch nhỏ, thu được Margin nhỏ.

nhỏ: Cho phép sai lệch lớn, thu được Margin lớn.

Tuỳ bài toán cụ thể mà ta cần điều chỉnh tham số này để thu được kết quả tốt nhất.

### Thuật toán Decision Tree

Cây quyết định (Decision Tree) là một phương pháp phổ biến trong việc khai phá dữ liệu, một mô hình thuộc nhóm thuật toán Học có giám sát (Supervised Learning). Tại mỗi nút, ta sẽ đối chiếu các điều kiện thực tế để quyết định rẽ nhánh nào. Nút lá là quyết định cuối cùng.

Decision Tree được áp dụng vào cả 2 bài toán: Phân loại (Classification) và Hồi quy (Regression).

* Cây hồi quy (Regression tree) ước lượng các hàm giá có giá trị là số thực thay vì được sử dụng cho các nhiệm vụ phân loại. (ví dụ: ước tính giá một ngôi nhà hoặc khoảng thời gian một bệnh nhân nằm viện)
* Cây phân loại (Classification tree), phân loại như: giới tính (nam hay nữ), kết quả của một trận đấu (thắng hay thua).

**Ưu điểm của việc phân tích cây quyết định**

* Học những đặc trưng từ data đầu vào, đầu ra là kết quả với hình dạng cây quyết định. Nghĩa là dễ dàng nhìn thấy đặc trưng của data đầu vào.
* Các loại phương pháp khác cần công việc tiền xử lý rất nhiều, còn với cây quyết định thì hầu như là không cần công đoạn tiền xử lý.
* Với các loại phân tích như Neural Network được coi như kiểu model hộp đen (nói đơn giản là bạn không hiểu trong hộp có gì, xử lý ra sao) thì Decision Tree giống như model hộp trắng.
* Hỗ trợ đánh giá độ chính xác của các model được tạo ra. Ở đây các model chính là các cây quyết định được tạo trong quá trình xử lý theo phương pháp này.

**Thuật toán cây quyết định**

Để phân tích cây quyết định thì có khá nhiều phương pháp, sự khác nhau như sau:

* ID3  
  ID3 (Iterative Dichotomiser 3) được phát triển vào nào 1986 bởi Ross Quinlan. Sử dụng lượng thông tin ứng với biến số phân loại sau đó dùng kỹ thuật tham lam ( = lựa chọn tối ưu địa phương ở mỗi bước đi với hy vọng tìm được tối ưu toàn cục ). Ví dụ như thuật toán tìm đường đi ngắn nhất của Dijkstra.
* C4.5  
  Được phát triển từ ID3. C4.5 là thuật toán phân lớp dữ liệu dựa trên cây quyết định hiệu quả và phổ biến trong những ứng dụng khai phá cơ sở dữ liệu có kích thước nhỏ. So với ID3, C4.5 không cần biến số phân loại lượng đặc trưng. Output theo dạng if-then, không hiển thị những phần cành không cần thiết.
* C5.0  
  Là bản cải tiến của C4.5. Giúp cải thiện vấn đề hiệu năng và sử dụng ít bộ nhớ hơn.
* CART  
  CART (Classification and Regression Trees) khá giống với C4.5. Được phát triển bởi Breiman năm 1984. Tạo cây phân tích dựa trên biến phân loại, giải thích, mục đích và hồi quy. Và scikit-learn có chứa phiên bản tối ưu hoá của CART.

**Các bước xây dựng cây Quyết định**

***Bước 1: Xây dựng cây***

* Ta xây dựng cây quyết định từ gốc đến ngọn
* Chọn 1 thuộc tính làm gốc và đưa các giá trị tương ứng của thuộc tính được chọn làm gốc vào các nhánh.
* Từ các nhánh trên, chọn các thuộc tính tương ứng là m nút trong. Sau đó tiếp tục lặp lại với cây con.

Thuật toán kết thúc:

* Mọi mẫu (bộ dữ liệu) đều thuộc cùng một lớp.
* Không còn mẫu để phân chia.
* Không còn thuộc tính nào để chọn phân chia

***Bước 2. Thu gọn cây.:*** Xác định và loại bỏ các nhánh cây gây nhiễu hoặc các giá trị tách rời khỏi nhóm

**Giải quyết bài toán hồi quy (Regression)**

- Thuật toán ID3 có thể được sử dụng để xây dựng cây quyết định cho hồi quy bằng cách thay thế Information Gain bằng Standard Deviation Reduction ( Giảm độ lệch chuẩn).

- Giống phương pháp phân tích hồi quy( Regression analysis) bao gồm: Linear Regression, Logistic Regression,…phân tích quan hệ, mức độ liên hệ, tính bền vững trong mối quan hệ tuyến tính giữa các biến đầu vào và biến mục tiêu( dựa vào dữ liệu lịch sử), lấy đó làm cơ sở để dự báo giá trị của biến mục tiêu cho đối tượng dữ liệu.

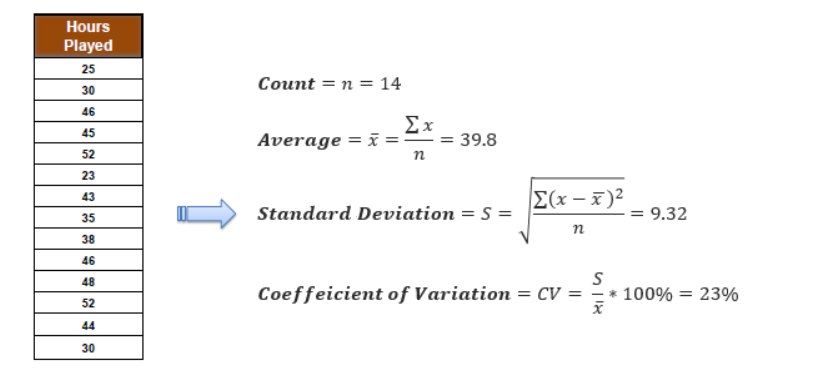
- Đối với Regression không biết chắc chắn giá trị nào của biến mục tiêu mà đối tượng dữ liệu có thể nhận, mà phải quan tâm đến mối quan hệ giữa toàn bộ các biến, vì nó sẽ ảnh hưởng đến giá trị sau này.

- Phương sai là trung bình cộng của bình phương các độ lệch giữa các giá trị của từng quan sát và số trung bình cộng(mean) của dãy số.

- Độ lệch chuẩn(Standard Deviation):

Cây quyết định được xây dựng từ trên xuống từ nút gốc và liên quan đến việc phân vùng dữ liệu thành các tập hợp con có chứa các thể hiện có giá trị tương tự (đồng nhất), do đó sử dụng độ lệch chuẩn để tính độ đồng nhất của mẫu số. Nếu mẫu số hoàn toàn đồng nhất thì độ lệch chuẩn của nó bằng không. Độ lệch chuẩn chính là căn bậc 2 của phương sai.

+ Độ lệch chuẩn cho 1 thuộc tính:



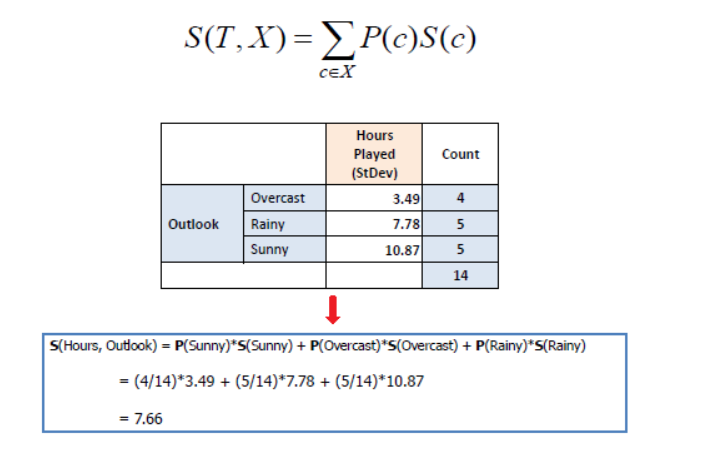
Hình .. Độ lệch chuẩn cho 1 thuộc tính

Với Độ lệch chuẩn (S) dành cho xây dựng cây (phân nhánh).

Hệ số sai lệch (CV) được sử dụng để quyết định khi nào ngừng phân nhánh, có thể sử dụng Count (n) là tốt.

Trung bình (Avg) là giá trị trong các nút lá.

+ Độ lệch chuẩn cho 2 thuộc tính( mục tiêu, yếu tố dự đoán):



Hình .. Độ lệch chuẩn cho 2 thuộc tính

**Giảm độ lệch chuẩn**

Việc giảm độ lệch chuẩn dựa trên việc giảm độ lệch chuẩn sau khi tập dữ liệu được phân chia trên một thuộc tính. Xây dựng cây quyết định là việc tìm tất cả thuộc tính trả về mức giảm độ lệch chuẩn cao nhất (nghĩa là các nhánh đồng nhất nhất).

### Thuật toán Random Forest

Random Forest — rừng ngẫu nhiên: đây là phương pháp xây dựng một tập hợp rất nhiều cây quyết định và sử dụng phương pháp voting để đưa ra quyết định về biến target cần được dự báo. Một ví dụ về Random Forest như sau: giả sử bạn muốn đi tham quan du lịch Anh và có sự cân nhắc cho việc tham quan thành phố nào như: Manchester, Liverpool hay Birmingham. Để trả lời câu hỏi này chúng ta sẽ cần tham khảo rất nhiều ý kiến từ bạn bè, blog du lịch, tour lữ hành … Mỗi một ý kiến tương ứng với một Decision Tree trả lời các câu hỏi như: thành phố này đẹp không, có được tham quan các sân vận động khi đến thăm không, số tiền bỏ ra là bao nhiêu, thời gian để tham quan thành phố là bao lâu… Sau đó bạn sẽ có một rừng các câu trả lời để quyết định xem mình sẽ đi tham quan thành phố nào. Random Forest hoạt động bằng cách đánh giá các Decision Tree sử dụng cách thức voting để đưa ra kết quả cuối cùng.

Về mặt toán học thuật toán có thể được giải thích như sau: Random Forest là một tập hợp của hàng trăm Decision Tree, trong đó mỗi Decision Tree được tạo nên ngẫu nhiên từ việc tái chọn mẫu (chọn random 1 phần của data để xây dựng) và random các biến từ toàn bộ các biến trong trong data. Với một cơ chế như vậy, Random Forest cho ta một kết quả chính xác rất cao nhưng đánh đổi bằng việc ta không thể hiểu cơ chế hoạt động của thuật toán này do cấu trúc quá phức tạp của mô hình này, do vậy thuật toán này là một trong những phương thức Black Box, tức ta sẽ bỏ tay vào bên trong và rút ra được kết quả chứ không thể giải thích được cơ chế hoạt động của mô hình. Đó là sự đánh đổi giữa khả năng giải thích và khả năng dự báo như mình đã nêu ở bài đầu tiên.

Random Forest là một phương pháp Supervised Learning do vậy có thể xử lý được các bài toán về Classification (phân loại) và Regression (dự báo về các giá trị)

## Tổng kết chương

Như vậy trong chương này chúng ta có được cái nhìn tổng thể về học máy, một số thuật toán điển hình cho bài toán hồi quy. Từ đó có thể phát triển hướng nghiên cứu tìm ra phương pháp để xây dựng bài toán định giá nhà.

# Chương 2

# TỔNG QUAN VỀ HỌC MÁY CHO ĐỊNH GIÁ NHÀ

## Nghiên cứu bài báo định giá

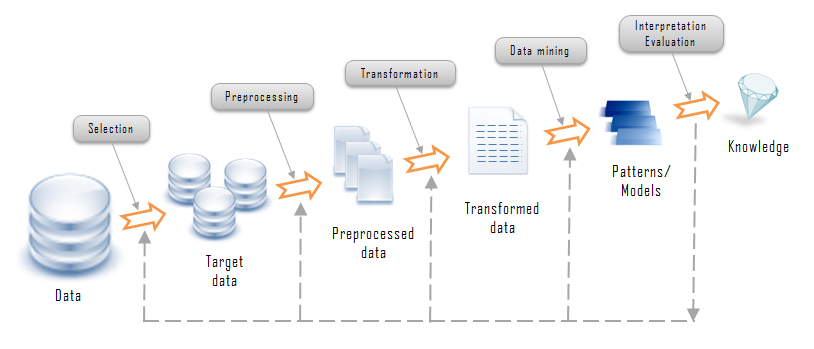
Thị trường bất động sản luôn là thị trường quan trọng nhất của của nền kinh tế. Bởi nó có giá trị lớn, phản ảnh sự phát triển của khu vực. Chính vì sự quan trọng của nó nên trên thế giới đã có một số bài báo khoa học được công bố viết về đề tài nghiên cứu học máy áp dụng cho bài toán định giá nhà. Một trong số đó phải kể đên như bài báo “*Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project”* [2]của Dean De Cock mô tả quá trình phân công, các bước thực hiện đề tài dự báo giá nhà ở Ames, Iowa Boston của tác giả cho một nhóm sinh viên làm bài tập cuối kì. Hay bài báo *"Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data”* [3]của hai đồng tác giả Byeonghwa Park, Jae Kwon Bae viết về việc xây dựng mô hình học máy cho định giá nhà ở quận Fairfax thuộc tiểu bang Virginia, Hoa Kỳ. Đặc điểm chung của hai bài báo này là nguồn dữ liệu được lấy từ một đơn vị của chính phủ, từ văn phòng giám định giá nhà. Tức là những dữ liệu này là dữ liệu thực tế, chính xác và mức độ tin cậy cao.

Còn đối với ở nước ta, hiện này cũng đã có một số đơn vị thực hiện thu thập dữ liệu dữ liệu bằng cách sử dụng con người để đi khảo sát và thu thập thông tin. Việc này cực kì tốn kém và được nhiên dữ liệu cũng sẽ không được mở ra cộng đồng. Đó chính là một trong số các lý do đề tài được ra đời để tìm kiếm một nguồn dữ liệu có thể sử dụng được cho bài toán định giá nhà.

## Qui trình ứng dụng học máy cho định giá nhà

Để xây dựng được mô hình học máy nói chung hay cho định giá nhà nói riêng cần được thực hiện qua các bước của quả trình khai phá dữ liệu.

### Quy trình khai phá dữ liệu



Hình .. Quy trình khai phá dữ liệu

Trong quy trình khai phá dữ liệu gồm có các giai đoạn:

* Bước 1: Xác định lĩnh vực cần tìm hiểu: các tri thức có liên quan trước đó, những mục tiêu cuối cùng cần đạt được…
* Bước 2: Tạo tập dữ liệu mục tiêu (data selection): chọn lựa một tập dữ liệu, hoặc tập trung vào một tập con các biến hoặc các mẫu dữ liệu. Trên tập dữ liệu được chọn, quá trình KPDL sẽ được tiến hành.
* Bước 3: Làm sạch dữ liệu (data preprocessing): Các phép toán cơ bản dùng để xử lý dữ liệu: lọc nhiễu, loại bỏ các giá trị bất thường (outliers), chọn lựa các mô hình dùng để xử lý đối với các dữ liệu không đầy đủ…
* Bước 4: Rút gọn dữ liệu: lựa chọn các đặc tính hữu ích mô tả dữ liệu phụ thuộc vào mục tiêu của bài toán. Sử dụng các phép toán rút gọn chiều dữ liệu và các phương thức chuyển đổi DL để rút gọn DL.
* Bước 5: Lựa chọn bài toán KPDL: quyết định nhiệm vụ chính của giai đoạn KPDL là gì?, phân lớp, phân cụm, hồi quy..
* Bước 6: Lựa chọn phương thức KPDL: lựa chọn phương thức sử dụng tìm kiếm mẫu, bao gồm cả việc quyết định mô hình và tham số của phương thức.
* Bước 7: Trích rút tri thức(data mining): tìm kiếm tri thức quan tâm từ sự thể hiện của dạng tìm kiếm: luật phân lớp, cây quyết định, hồi quy, phân cụm…
* Bước 8: Biểu diễn và đánh giá tri thức (knowledge representation and evaluation): Đánh giá mẫu thu được và biểu diễn tri thức, hiển thị hóa, chuyển đổi, bỏ đi các mẫu dư thừa,…
* Bước 9: Sử dụng tri thức được khai phá

### Quy trình xây dựng mô hình định giá

Áp dụng theo các bước của quả trình khái phá dữ liệu ta có các chi tiết các bước thực hiện xây dựng mô hình định giá.

#### Data Selection

Tại bước này chúng ta tìm kiếm thu thập lựa chọn những bộ dữ liệu về giá nhà với đầy đủ các thông tin mô tả đặc điểm của giá nhà. Những bộ dữ liệu này phải là những bộ dữ liệu uy tín, chính xác, thể hiện được giá cả thị trường nhà đất. Nó được phân loại theo khu vực, theo loại nhà là nhà riêng để ở, nhà mặt phố có giá trị kinh doanh hay nhà chung cư, … . Ngoài ra trong bước này chúng ta cần phải khám phá dữ liệu thu thập được để thấy được các đặc điểm của bộ dữ liệu từ đó đánh giá bộ dữ liệu đã cho có khả năng xây dựng được mô hình định giá nhà hay không.

#### Data Preprocessing

Sau khi lựa chọn được bộ dữ liệu và khám phá nó để thấy được những đặc điểm của bộ dữ liệu chúng ta cần thực hiện các phương pháp để làm sạch dữ liệu, lọc bỏ dữ liệu nhiễu và impute missing value.

Lọc dữ liệu nhiễu (outlier) dữ liệu có những giá trị bất thường, (do giá trị đo của các trường hợp rất đặc biệt, do lỗi của công cụ đo lường,…). Lỗi này là lỗi khó phát hiện nhất, gây ra các sai sót nghiêm trọng làm sai lệch, giảm hiệu quả thực hiện các thuật toán khai phá.

Xử lý dữ liệu khuyết thiếu (impute missing value): là công việc chúng ta phải tìm một giá trị phù hợp để điền vào vị trí bị khuyết thiếu trong bộ dữ liệu. Có một số phương pháp để xử lý dữ liệu khuyết thiếu như:

* Một giá trị hằng có ý nghĩa trong miền xác định của dữ liệu, ví dụ như 0.
* Một giá trị của một đặc trưng từ một mẫu dữ liệu ngẫu nhiên trong tập dữ liệu.
* Các giá trị thống kê cơ bản như giá trị trung bình, giá trị trung vị hay giá trị mốt (mode) của cột.
* Một giá trị được ước lượng từ một mô hình dự đoán khác.Mỗi phương pháp sẽ phù hợp với từng bộ dữ liệu cụ thể.

#### Tranform Data

Biến đổi dữ liệu là việc chuyển toàn bộ tập giá trị của một thuộc tính sang một tập các giá trị thay thế sao ho mỗi giá trị cũ tương ứng với một giá trị mới. Trong bước này chúng ta lựa chọn và thực hiện các công việc tùy thuộc vào mục đích và đặc điểm của bộ dữ liêu để đưa dữ liệu về dạng chuẩn:

* Điều chỉnh lại tỉ lệ dữ liệu trên các đặc tính (Rescale)
* Chuẩn hóa dữ liệu (Standardization)
* Bình thường hóa dữ liệu (normalization)
* Số hóa dữ liệu (Digitalization)

Điều chỉnh tỷ lệ (rescale) Dữ liệu gồm nhiều đặc tính (cột), và mỗi đặc tính thì lại có các đơn vị và độ lớn nhỏ khác nhau. Điều này tác động tới tính hiệu quả của nhiều thuật toán, ví dụ thời gian thực hiện, quá trình hội tụ, hay thậm chí ảnh hưởng cả tới độ chính xác của thuật toán. Chính vì vậy, người ta thường tiến hành điều chỉnh dữ liệu để các đặc tính cùng có chung một tỉ lệ (data scaling) . Và thường để các đặc tính có giá trị trong khoảng [0, 1]. Kết quả sẽ giúp cho nhiều thuật toán quan trọng trong Máy học sử dụng kĩ thuật Gradient Descent hội tụ nhanh. Việc điều chỉnh tỉ lệ thường dùng công thức sau đây (giả sử chúng ta đang làm trên một cột dữ liệu số cụ thể, gọi là F):

Trong đó: là gì trị ban đầu trong cột F

là giá trị sau khi được điều chỉnh

là gì trị nhỏ nhất trong cột F

là giá trị lớn nhất trong cột F

Phương pháp trên được gọi là MinMaxScaler sẽ đưa bộ dữ liệu của chúng ta về khoảng giá trị [0,1].

Chuẩn hóa dữ liệu (Standardize Data): Nhiều thuật toán trong Máy học giả định rằng dữ liệu đầu vào có phân phối Gauss. Chính vì vậy, khi chuẩn hóa dữ liệu về dạng chuẩn phân phối Gauss với giá trị trung bình bằng 0 và phương sai bằng 1. Nhờ việc chuẩn hóa, các thuật toán như linear regression, logistic regression được cải thiện. Công thức được sử dụng trong phương pháp này là :

Trong đó: giá trị trung bình

Độ lệch chuẩn :

Bình thường hóa dữ liệu (Normalize Data): Bình thường hóa dữ liệu là sự điều chỉnh tỉ lệ dữ liệu sao cho mỗi thể hiện (trên hàng) đều cho độ dài là 1. Kĩ thuật này rất cần thiết cho dữ liệu thưa (gồm nhiều số 0) trên mỗi cột đặc tính. Điều này đặc biệt ảnh hưởng tới các thuật toán lấy trọng số của các giá trị nhập vào như neuron networks, hay các thuật toán dùng độ đo khoảng cách (như k-Nearest Neighbors)

Số hóa dữ liệu (Digitalization): Rất nhiều thuật toán học máy hiện nay chỉ chấp nhận dữ liệu dạng số. Những những liệu dạng category chúng ta đều phải chuyển sang dạng số. Ở đây chúng ta có 3 phương pháp đễ mã hóa dữ liệu, chuyển dữ liệu category về dạng sô. Ví dụ một đặc tính có N giá trị khác nhau

* Mã hoá một trong N (One of N encoding): sử dụng một tập gồm N các cột nguyên để chuẩn hoá, low (0, 0, 1), mid (0, 1, 0), high (1,0,0)
* Mã hoá nhị phân (Binary encoding): chuẩn hoá thành M cột nhị phân, với M l= [log2N]. low (0,0), mid (1,0), high (1,1)
* Mã hoá số nguyên (Numeric encoding): chuẩn hoá thành một cột có giá trị nguyên, mỗi giá trị của cột tương ứng với thứ tự của giá trị gốc có trong tập giá trị ban đầu: low (1), mid (2), high (3)

Bên cạnh đó, trong bước này chúng ta còn thực hiện rút gọn dữ liệu, lựa chọn các đặc trưng phù hợp để đưa vào mô hình hay còn gọi là feature selection

#### Lựa chọn thuật toán, xây dựng mô hình

Định giá nhà là một bài toán hồi quy. Chính vì thế các thuật toán được lựa chọn là các thuật toán học máy phục vụ cho bài toán phổ biển và thông dụng nhất hiện nay. Nghĩ đến bài toán hồi quy thì thuật toán đầu tiên được nghĩ đến đó chính là Linear Regression. Đây là thuật toán cơ bản nhất của bài toán hồi quy. Ngoài ra hiện nay có một số thuật toán khác có tính hiệu quả cao là KNN, Decision Tree, SVM, Random Forest.

#### Đánh giá chất lượng mô hình mô hình

Ta đánh giá chất lượng mô hình dựa vào các độ đo là sai số tuyệt đối trung bình MAE (mean absolute error) và phần trăm sai số tuyệt đối trung bình MAPE (mean absolute percentage error). Đây là hai độ đo được tính toán trên kết quả ở cả tập training và tập test ở tất cả các mô hình được xây dựng để lựa chọn đánh giá mô hình.

Một mô hình tốt là một mô hình có kết quả các độ đo MAE và MAPE càng nhỏ càng tốt

Ngoài ra có một khái niện nữa để đánh giá chất lượng mô hình là overfitting. Overfitting là hiện tượng mô hình tìm được quá khớp với dữ liệu training. Việc quá khớp này có thể dẫn đến việc dự đoán nhầm nhiễu, và chất lượng mô hình không còn tốt trên dữ liệu test nữa. Dữ liệu test được giả sử là không được biết trước, và không được sử dụng để xây dựng các mô hình Machine Learning. Về cơ bản, overfitting xảy ra khi mô hình quá phức tạp để mô phỏng training data. Điều này đặc biệt xảy ra khi lượng dữ liệu training quá nhỏ trong khi độ phức tạp của mô hình quá cao và không bị hiện tượng overfitting tức sự chênh lệch giữa các độ đô trên tập training và tập testing càng nhỏ càng tốt. Một mô hình tốt sẽ không xảy ra hiện tượng overfitting hoặc sự chênh lệch này được giảm xuống thấp nhất có thể.

Từ viện đánh giá chất lượng mô hình chúng ta sẽ quyết định cải tiến mô hình bằng cách thay đổi, bổ sung phương pháp xử lý dữ liệu hay chuẩn hóa dữ liệu hay tiến hành tunning tham số của thuật toán để tăng độ chính xác và giảm hiện tượng overfitting của mô hình.

## Giới thiệu về cơ sở dữ liệu của bài toán định giá nhà

Cơ sở dữ liệu của bài toán định giá nhà là một tập các thuộc tính được lựa chọn để có thể đưa vào mô hình học máy phải là những những thuộc tính thuộc kiểu dữ liệu number hoặc category, có thông tin rõ ràng, có cấu trúc, miêu tả chính xác đặc điểm căn nhà.

Chúng ta cùng tìm hiểu một số bộ dữ liệu được sử dụng trong các bài báo nghiên cứu khoa học trên thế giới. Đầu tiên là bộ dữ liệu nhà ở Ames, Iowa Boston được sử dụng trong bài báo “*Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project”* [2]*.* Bộ dữ liệu có một số đặc điểm như sau:

- Bộ dữ liệu được lấy từ Văn phòng giám định thành phố Ames gồm có 113 biến với độ lớn là 3970 căn nhà từ năm 2006- 2010.

- Lược bớt dữ liệu thay đổi quyền sở hữu nhiều lần trong 4 năm, loại bỏ những căn nhà có diện tích quá lơn, giữ lại những giao dịch gần đây nhất. Cuối cùng dữ liệu còn lại 2930 với 82 biến.

- Bộ dữ liệu gồm có 82 đặc trưng trong đó:

+ 2 biến PID, order

+ 23 biến định danh

+ 23 biến thứ tự

+ 20 biến liên tục

+ 14 biến không liên tục

Trong đó Biến định danh là các thông tin: Loại nhà, khu vực, loại đường, độ phẳng, kiểu mái hệ thống điện, sưởi, điều hòa, … Biến thứ tự đánh giá về chất lượng nhà, vật liệu, mức độ hoàn thiện, chất lượng tầng hầm, nhà để xe, hang rào,… Biến rời rạc :năm xây dựng, số lượng phòng ngủ, phòng tắm, bếp, năm sửa chữa, kích thước nhà để xe. Biến liên tục: tổng diện tích, diện tích từng tầng, diện tích sinh hoạt chung, diện tích bể bơi,…

Nhận xét áp dụng và tình hình Việt Nam :

+ Có thể bỏ những đặc trưng không cần thiết liên quan đến lò sửa

+ Cần xem xét các đặc trưng liên quan đến tầng hầm, nhà để xe, bể bơi, vật liệu xây dựng tổng thể. Tùy thuộc vào từng khu vực định giá để cân nhắc.

## Tổng quan về các mô hình cho định giá nhà

Mô hình định giá nhà tại Ames, Iowa Boston được thực hiện có các đặc điểm sau:

* Xử lý dữ liệu :
  + Sử dụng giá trị trung bình để impute missing value
  + Sử dụng phương pháp Standardization để đưa dữ liệu về khoảng [-1,1]
  + Sử dụng phương pháp onehot encoding để chuyển dữ liệu dạng category về dạng số
* Các thuật toán được sử dụng là: Ridge regression, Lasso, KNN, Decision Tree, SVR. Trong đó Ridge regression là thuật toán tốt nhất. Decision Tree thực hiện traning rất tốt nhưng khi đưa bộ test vào thì lại không hiệu quả. Thuật toán lasso regression thực hiện kém hiệu quả nhất.
* Sau đó thực hiện tunning tham số để tìm ra các tham số phù hợp nhất với bộ dữ liệu.

# Chương 3

# HỌC MÁY CHO ĐỊNH GIÁ BÁN NHÀ RIÊNG TẠI HÀ NỘI

## Thu thập dữ liệu

### Nguồn gốc dữ liệu

Một trong số các mục đích của đề tài là xây dựng một bộ dữ liệu về bất động sản bao gồm giá, các đặc trưng cũng như mô tả về ngôi nhà phục vụ cho mục đích nghiên cứu và đồng thời xây dựng được một mô hình định giá nhà từ bộ dữ liệu này. Trong thực tế hiện nay, để có được một bộ dữ liệu đầy đủ, chính xác nhất về giá trị của căn nhà thực sự là khó khăn và chưa có một đơn vị nào có thể thu thập đầy đủ, chính xác giá trị thực sự của căn nhà. Bởi, giá trị căn nhà dựa trên thỏa thuận của người mua và người bán nên sự trung thực khi khai báo trong quá trình làm thủ tục pháp lý là không cao. Ngoài ra, để có thể định giá được chính xác nhất giá trị căn nhà cần phải có các chuyên gia nhiều năm kinh nghiệm trong lĩnh vực bất động sản ở từng khu vực, từng loại hình thì mới có thể định giá được. Thế nên việc thu thập dữ liệu về bất động sản từ việc khảo sát, định giá trực tiếp là không khả thi và rất tốn kém.

Thật may mắn, chúng ta còn có một nguồn dữ liệu khác đó chính là những tin rao bán bất động sản trên các website môi giới bất động sản. Đây là những tin do người bán, môi giới bất động sản đăng tin mô tả căn nhà, các thông tin cơ bản của căn nhà cũng như giá trị căn nhà mà người đăng mong muốn. Giá trị này có thể không đúng với giá trị thực của căn nhà nhưng để dễ dàng tìm được người mua nhất thì giá trị này cũng rất gần so với giá trị thực của căn nhà. Ngoài ra giá trị này vẫn phản ánh được xu hướng giá cả và vẫn theo mặt bằng chung của thị trường nên nguồn dữ liệu này hoàn toàn có thể sử dụng được.

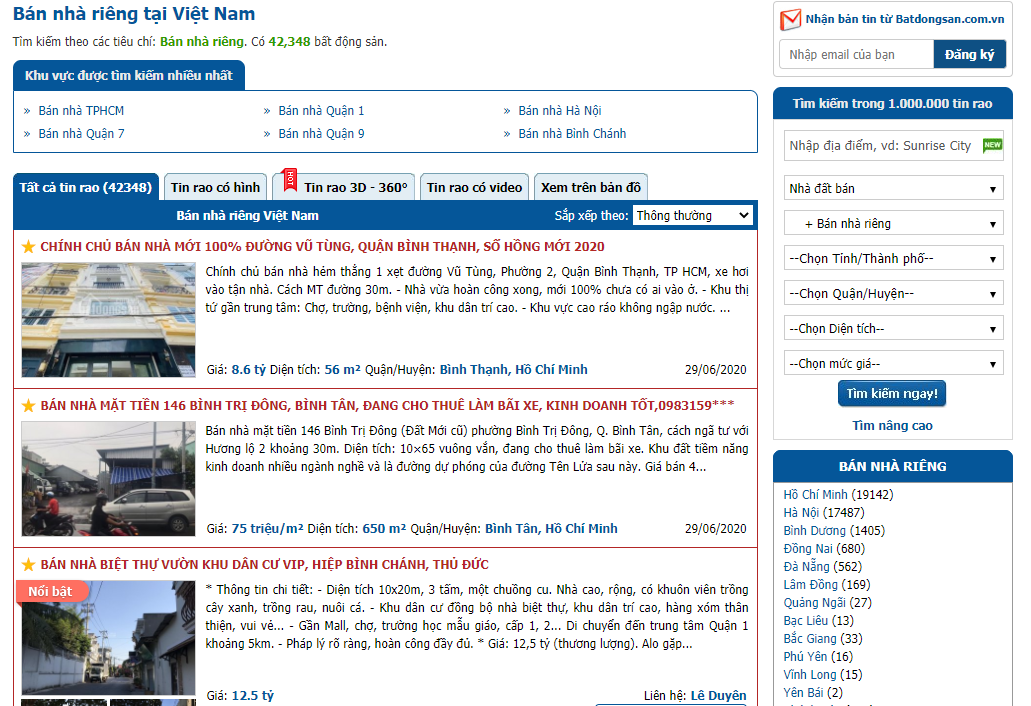
Dữ liệu được sử dụng trong đề tài được thu thập từ một trang môi giới bất động sản với đông đảo người dùng ở Việt Nam được chọn lọc là những tin bán nhà riêng, nhà trong ngõ hẻm tại các quận, huyện trên địa bàn Hà Nội trong vòng 5 tháng từ tháng 1/2020 đến tháng 5/2020 được 64173 bản ghi. Các thông tin đặc trưng của căn nhà thu thập được là tối đa các thông tin cung cấp trên trang web.

### Cách thức thu thập dữ liệu

Trang web được lựa chọn để thu thập dữ liệu là [*http://batdongsan.com*](http://batdongsan.com). Đây một trang môi giới bất động sản với đông đảo người dùng ở Việt Nam. Những tin được chọn lọc là những tin bán nhà riêng, nhà trong ngõ hẻm tại các quận, huyện trên địa bàn Hà Nội trong vòng 5 tháng từ tháng 1/2020 đến tháng 5/2020 được 64173 bản ghi. Các thông tin đặc trưng của căn nhà thu thập được là tối đa các thông tin cung cấp trên trang web. Sau đây là chi tiết cách thức thực hiện thu thập dữ liệu.

#### Khảo sát trang web

Trang web [*http://batdongsan.com*](http://batdongsan.com) là một trang web môi giới bất động sản nơi người cần bán, cần cho thuê hay sang nhượng bất động sản đăng tin lên và người mua, người có nhu cầu thuê bất động sản lên đó để tìm bất động sản. Nó có các chức năng cơ bản như tìm kiếm tin đăng, lọc tin đăng theo nhu cầu, loại hình, tỉnh/ thành phố, quận huyện,… và hiển thị chúng dưới dạng một danh sách các tin đăng có thể được sắp xếp theo giá, theo giá, theo diện tích như hình :



Hình .. Giao diện trang web

Các tin đăng của trang web được chia theo từng trang. Đường dẫn tới các trang được thay đổi theo số đánh tương ứng của các trang từ 1 đến trang cuối cùng ví dụ [*https://batdongsan.com.vn/ban-nha-rieng-ha-noi/p2*](https://batdongsan.com.vn/ban-nha-rieng-ha-noi/p20)là trang thứ 2.

Sau khi hiển thị danh sách tin đăng, ta truy cập vào chi tiết tin đăng bằng cách click vào tiêu đề tin đăng tương ứng sẽ sang một giao diện mới hiển thị chi tiết tin đăng với các thông tin đặc điểm của căn nhà.

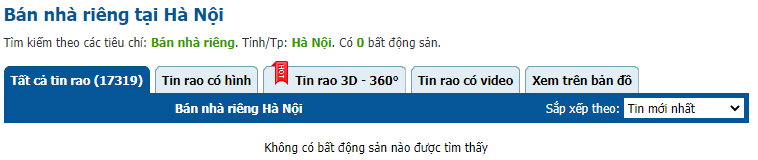
#### Quy trình các bước thu thập dữ liệu

Sau khi khảo sát trang web ta xây dựng các bước thực hiện thu thập dữ liệu:

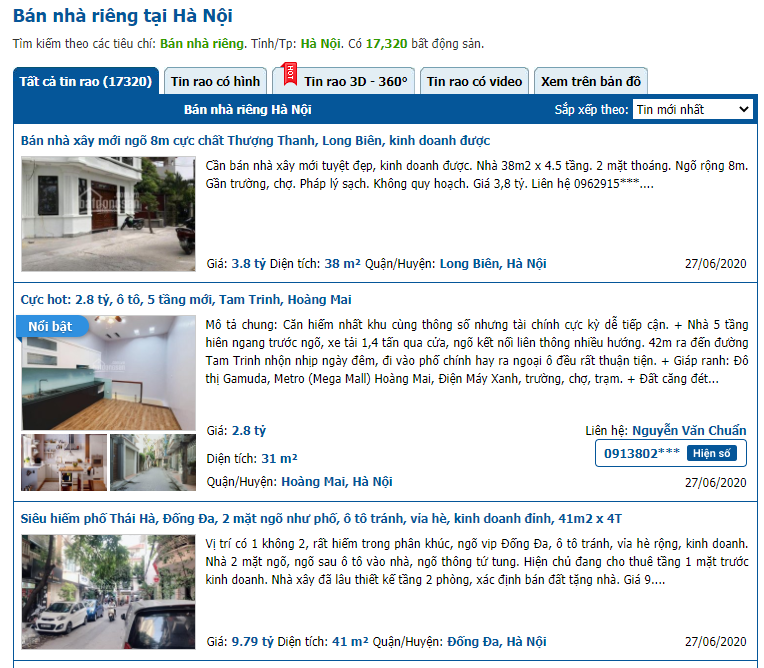
Bước 1: Thu thập các đường dẫn tin đăng URL.

Đây là các đường dẫn dẫn đến trang chi tiết tin đăng. Để thu thập được các đường dẫn tin đăng cần sử dụng tính năng filter của trang web để lọc nhưng tin bán nhà riêng tại Hà Nội. Dữ liệu được lọc chỉ là những tin trong khoảng hơn 1 tháng gần nhất. Chính vì thế để thu thập được nhiều tin nhất thì cần phải tiến hành thu thập định kỳ theo tháng, sử dụng chức năng sắp xếp theo tin mới nhất và chỉ lấy những tin trong tháng đó như Hình 3.3.

Trang web có sử dụng chức năng phân trang với đường dẫn trang 1: [*https://batdongsan.com.vn/ban-nha-rieng-ha-noi/*](https://batdongsan.com.vn/ban-nha-rieng-ha-noi/p200), đường dẫn trang 2: [*https://batdongsan.com.vn/ban-nha-rieng-ha-noi/p2*](https://batdongsan.com.vn/ban-nha-rieng-ha-noi/p20). để thu thập được các trang trang ta chỉ cần thay đổi số trang phía cuối đường dẫn đến khi trang hiển thị như hình dưới hoặc đến chặn dưới thời điểm lọc thì dừng lại.



Hình .. Trang cuối không có dữ liệu

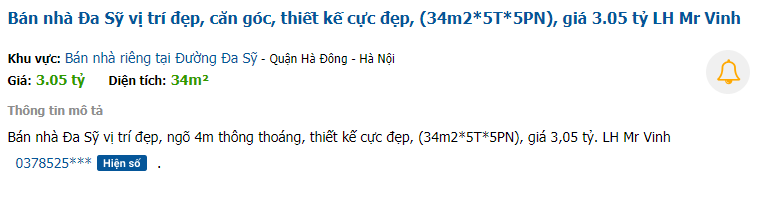


Hình .. Dữ liệu được lọc trên website

Tại mỗi trang ta thu thập các đường dẫn URL được đính kèm trong title tin đăng.

Bước 2: Thu thập thông tin của bản tin

Sau khi thu thập được các đường dẫn đến chi tiết tin đăng ta tiến hành thu thập chi tiết các tin đăng.



Hình .. Chi tiết tin đăng

Ở Hình 3.4 những thông tin chúng ta có thể thu thập về là tiêu đề tin đăng, khu vực bán quận/huyện – tỉnh/thành phố, giá, diện tích, và thông tin mô tả căn nhà.



Hình .. Đặc điểm bất động sản

Với Hình 3.5 những thông tin chúng ta có thể thu thập về là loại tin rao, địa chỉ, mặt tiền, hướng nhà, hướng ban công, số tầng, số phòng ngủ, số toilet, nội thất.

#### Xây dựng cơ sở dữ liệu

Sau quá trình khảo sát trang web, xây dựng các bước thu thập dữ liệu ta có thể năm được những thông tin nào cần lưu trữ trong cơ sở dữ liệu. Chính vì thế ta chuyển sang bước tiếp theo là bước xây dựng cơ sở dữ liệu.

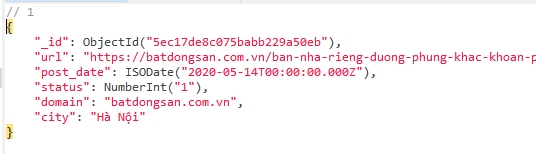
Dữ liệu của chúng ta cần lưu trữ là thông tin URL và thông tin căn nhà. Dữ liệu URL được cập nhật theo từng tháng và cần phải lưu trữ độc lập với thông tin căn nhà tránh tình trạng xử lý dữ liệu bị mất mát thông tin và để tiện cho việc thu thập lại thông tin bài đăng khi xảy ra sự cố. Với đặc điểm hai thông tin cần lưu trữ như vậy ta chọn hệ quản trị cơ sở dữ liệu mongoDB bởi tính linh hoạt trong lưu trữ vì được lưu dưới dạng json gồm các key-value và hiệu năng truy xuất cao của nó. Thông tin URL và thông tin căn nhà sẽ được lưu vào hai collection, mỗi document trong collection URL là thông tin của một URL và mỗi document trong collection bài đăng là thông tin của một căn nhà.

Các thông tin được lưu trữ trong collection URL là:

Bảng .. Các thông tin lưu trữ trong collection URL

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Ý nghĩa | Ví dụ |
| 1 | \_id | Id của document | 5ec17de8c075babb229a50eb |
| 2 | url | Đường dẫn tin đăng | <https://batdongsan.com.vn/ban-nha-rieng-duong-phung-khac-khoan-phuong-pham-dinh-ho/chinh-chu-can-mat-pho-khoan-pr25438308> |
| 3 | Post\_date | Ngày đăng | 2020-05-14 00:00:00.000 |
| 4 | status | Trạng thái để đánh dấu những tin đã được crawl hay chưa. | 1 |
| 5 | domain | Tên miền trang web | batdongsan.com.vn |
| 6 | city | Tỉnh/thành phố | Hà Nội |

Ví dụ một document được lưu trong collection URL:



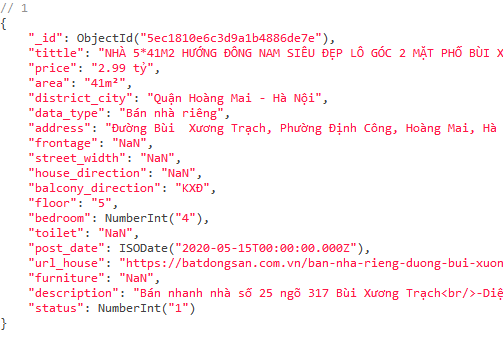
Hình .. Ví dụ một document được lưu trong collection URL

Các thông tin được lưu trữ trong collection rawData – dữ liệu thô chưa qua xử lý là:

Bảng .. Các thông tin lưu trữ trong collection rawData

|  |  |  |
| --- | --- | --- |
| STT | Tên thuộc tính | Ý nghĩa |
| 1 | \_id | Id document |
| 2 | title | Tiêu đề tin đăng |
| 3 | price | Giá trị căn nhà |
| 4 | area | Diện tích căn nhà |
| 5 | district\_city | Tên quận/ huyện và tỉnh/ thành phố |
| 6 | data\_type | Loại hình tin đăng |
| 7 | address | Địa chỉ căn nhà |
| 8 | frontage | Độ rộng mặt tiền |
| 9 | street\_width | Độ rộng đường vào |
| 10 | house\_direction | Hướng nhà |
| 11 | balcony\_direction | Hướng ban công |
| 12 | floor | Tổng số tầng. |
| 13 | bedroom | Số phòng ngủ |
| 14 | toilet | Số toilet |
| 15 | post\_date | Ngày đăng tin |
| 16 | furniture | Mô tả nội thất |
| 17 | description | Mô tả căn nhà |
| 18 | url\_house | Đường dẫn tin đăng |
| 19 | status | Đánh dấu trạng thái đã dữ liệu đã được xử lý chưa |

Ví dụ một document được lưu trong collection rawData:



Hình .. Ví dụ một document được lưu trong collection rawData

### Mô tả dữ liệu thô

Dữ liệu thô là dữ liệu nguyên bản được thu thập về từ website chưa thông qua xử lý, trích rút bổ sung thông tin.

Tổng số bản ghi thu thập được là 64173 bản ghi.

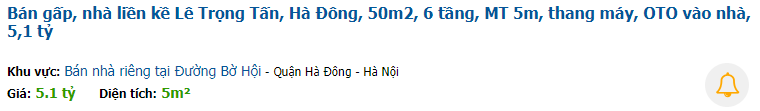
Các thuộc tính thu thập được được thống kê và mô tả trong bảng sau:

Bảng .. Liệt kê các thuộc tính thu thập được

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Ý nghĩa | Ví dụ |
| 1 | title | Tiêu đề tin đăng | CHÍNH CHỦ CẦN BÁN NHÀ RIÊNG… |
| 2 | price | Giá trị căn nhà | 2.99 tỷ |
| 3 | area | Diện tích căn nhà | 41m². |
| 4 | district\_city | Tên quận/ huyện và tỉnh/ thành phố | Quận Hoàng Mai - Hà Nội |
| 5 | data\_type | Loại hình tin đăng | Bán nhà riêng |
| 6 | address | Địa chỉ căn nhà | Đường Bùi Xương Trạch, Phường Định Công, Hoàng Mai, Hà Nội |
| 7 | frontage | Độ rộng mặt tiền | 4,40 |
| 8 | street\_width | Độ rộng đường vào | 6 |
| 9 | house\_direction | Hướng nhà | Đông-Nam |
| 10 | balcony\_direction | Hướng ban công | Đông-Nam |
| 11 | floor | Tổng số tầng. | 5 |
| 12 | bedroom | Số phòng ngủ | 4 |
| 13 | toilet | Số toilet | 3 |
| 14 | post\_date | Ngày đăng tin | 2020-05-14 |
| 15 | furniture | Mô tả nội thất | Đầy đủ điều hòa, thang máy, nóng lạnh, quạt trần, tivi,… |
| 16 | description | Mô tả căn nhà | Có căn nhà 4,5 tầng mới xây, đang hoàn thiện tại tổ 13 phường Bồ Đề… |

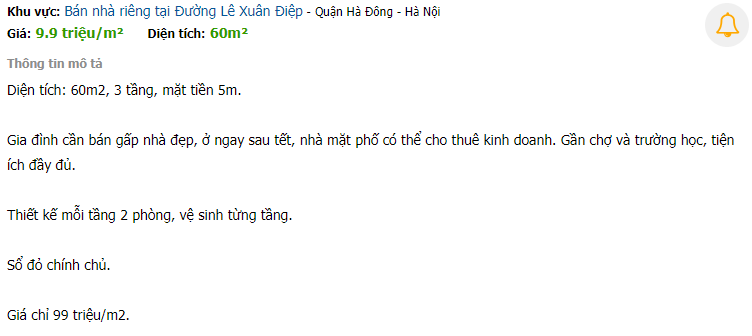
Như đã trình ở trên, dữ liệu được thu thập là những tin rao bán nhà do người bán, nhà môi giới đăng nên tính chính xác phụ thuộc rất lớn vào người đăng. Vì lý do đó các dữ liệu thu thập có rất nhiều điểm bị ghi sai do lỗi cả vô ý và cố ý của người đăng. Một vài lỗi sai điển hình như viết nhầm giá, diện tích căn nhà ví dụ như nhà có giá 2.5 tỷ thì viết nhầm là 25 tỷ, nhà 30 m² được viết thành 3 m² như các hình sau:

* Ghi thông tin vô lý:



Hình .. Người dùng ghi thông tin vô lý

* Ghi sai thông tin so với mô tả:



Hình .. Người dùng ghi sai thông tin giá so với mô tả

## Tiền xử lý dữ liệu

Dữ liệu thu thập được là dữ liệu thô, chưa được thống nhất về loại dữ liệu, kiểu dữ liệu. Chính vì thế bước tiền xử lý dữ liệu cần được thực hiện để đưa dữ liệu được thống nhất theo các trường, chia, tách, biến đổi các thuộc tính thu thập được thành các thuộc tính thống nhất, có cấu trúc và ngoài ra tìm phương án bổ sung thêm thuộc tính cho dữ liệu. Các công việc tiền xử lý dữ được thực hiện qua các bước sau:

Bước 1: Đưa về chuẩn đơn vị

Thuộc tính giá - price và diện tích - area vẫn còn để tên đơn vị trong dữ liệu. Ngoài ra đơn vị của giá còn chưa thống nhất. Vẫn còn tồn tại có đơn vị của giá như: tỷ, triệu, triệu/m². Chính vì thế cần phải đưa về một đơn vị thống nhất. Đối với diện tích loại bỏ tên đơn vị ra khỏi dữ liệu, thống nhất đơn vị m² cho trường diện tích - area. Đối với thuộc tính giá sẽ được tính toán để tách thành hai thuộc tính là thuộc tính giá cả căn - price và giá tính theo m² - price\_m2 trong đó thống nhất đơn vị cho trường giá cả giá cả căn – price là đơn vị tỷ và giá tính theo m² - price\_m2 là đơn vị triệu/m². Dữ liệu của hai thuộc tính này được tính toán dữ vào thông tin giá và diện tích của dữ liệu thô thu thập từ trang web.

Bước 2: Tách thông tin quận/ huyện và tỉnh/ thành phố

Thông tin về quận/ huyện và tỉnh/ thành phố trong dữ liệu thu thập về được ghép chung vào một thuộc tính ngăn cách bởi dấu gạch nối. Ví dụ “Quận Hoàng Mai - Hà Nội”. Chính vì thế chúng ta cần phải tách hai trường thuộc tính này ra làm hai độc lập với nhau là city – tên tỉnh/ thành phố và district – tên quận/ huyện.

Bước 3: Tách tên đường, tên xã phường.

Quan sát thông tin trường address – địa chỉ ta nhận thấy rằng có rất nhiều thông tin địa chỉ của bài đăng được viết theo tên đường, tên phường/xã, tên quận/ huyện, tên tỉnh/thành phố như hình:



Hình .. Quy luật trong cách viết địa chỉ

Dựa vào quy luật này ta có thể tách thêm thông tin tên đường, tên phường/xã từ thuộc tính địa chỉ của bài đăng để bổ sung thêm 2 thuộc tính cho dữ liệu.

Bước 4: Chuyển địa chỉ nhà thành địa chỉ lat, long.

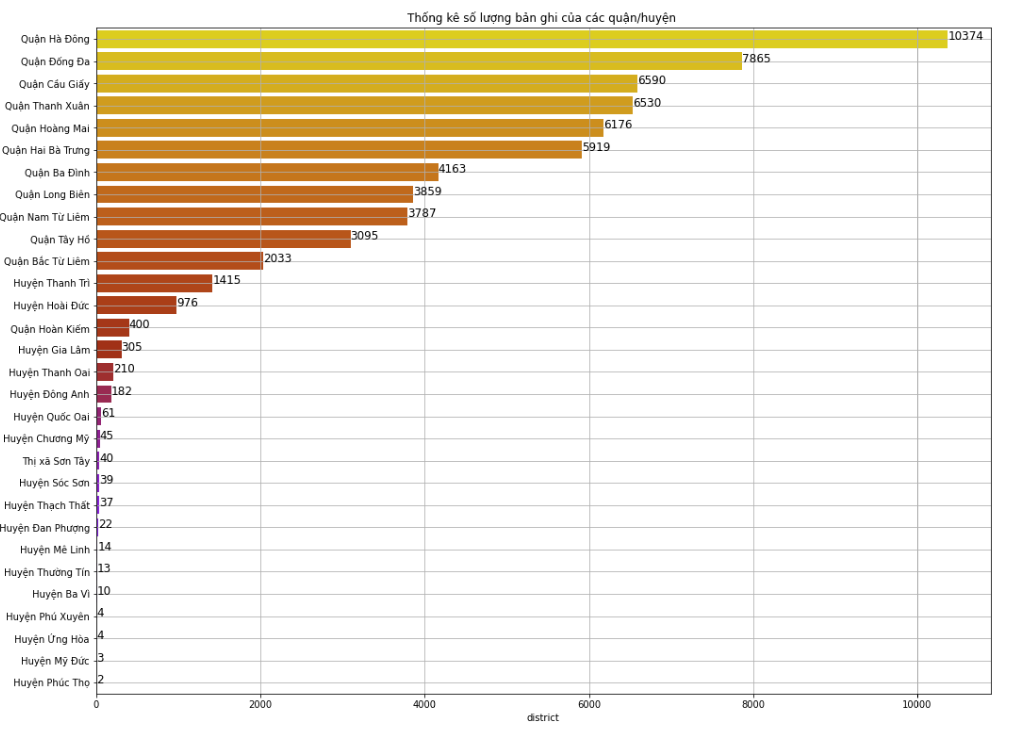
Địa chỉ - address là một trường nhằm xác định vị trí của tin đăng. Nội dung trường này rất khó để tách chi tiết đến từng số nhà, ngõ ngách mà chỉ dừng lại ở tên đường, phường/ xã như ở bước 3. Chính vì thế chúng ta cần phải chuyển đổi dữ liệu địa chỉ thành vị trí tọa độ lat, long sử dụng Google API. Với thông tin này chúng ta có thể thể hiện chính xác vị trí của từng tin đăng cũng như mối quan hệ khoảng cách giữa chúng.

Bước 5: Tách thêm thông tin từ mô tả căn nhà.

Trên thực tế để định giá căn nhà thì nhưng thông tin chúng ta có được vẫn còn là rất ít. Ngoài những thông tin đó ra chúng tôi mong muốn và tìm phương án thu thập được thêm những thông tin quan trọng khác như là thông tin pháp lý, đặc điểm nổi trội căn góc, nhà có mấy mặt tiền, mấy mặt thoáng, hình dạng ô đất ra sao,… . Sau quá trình khảo sát, phân tích mô tả của căn nhà chúng tôi chỉ trích xuất thêm thông tin pháp lý của căn nhà là sổ đỏ hay sổ hồng để bổ sung vào dữ liệu.

## Khám phá dữ liệu

Tổng số lượng các bản ghi thu thập được là 64173. Trong đó số lượng bản ghi thu thập được cụ thể theo từng quận/huyện được thống kế trong hình :



Hình .. Thống kê số lượng bản ghi theo từng quận/huyện

Hình trên cho thấy số lượng bản ghi thu thập được tập chung tại các quận nội thành Thành phố Hà Nội như Hà Đông, Đống Đa, Cầu Giấy,… .

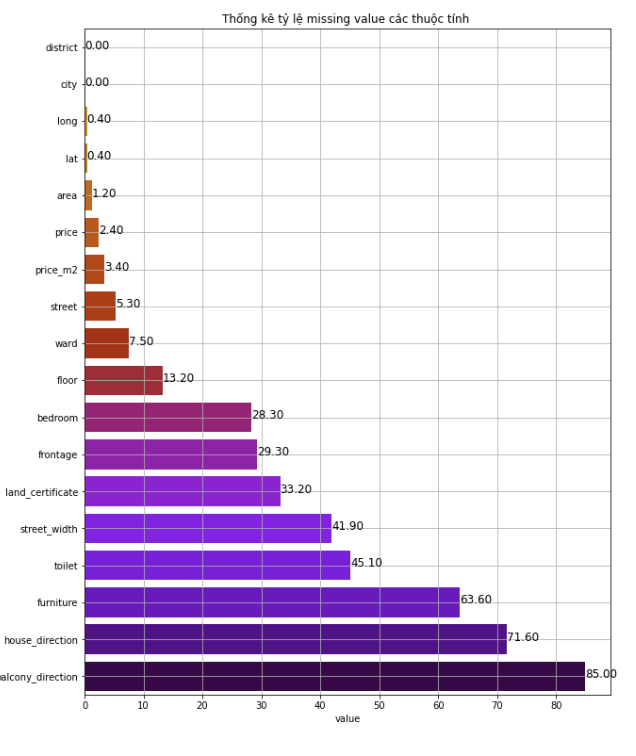
Sau bước tiền xử lý dữ liệu những thuộc tính có thể được đưa vào mô hình là:

Bảng .. Danh sách các thuộc tính có thể đưa vào mô hình

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ý nghĩa | Ví dụ |
| 1 | Title | Text | Tiêu đề tin đăng |  |
| 2 | price | Number | Giá trị căn nhà. Đơn vị tỷ | 2.99 |
| 3 | area | Number | Diện tích căn nhà. Đơn vị m². | 41 |
| 3 | price\_m2 | Number | Giá theo mét vuông. Đơn vị triệu/ m². | 32 |
| 4 | district\_city | Text | Tên quận/ huyện và tỉnh/ thành phố | Quận Hoàng Mai - Hà Nội |
| 5 | district | Category | Tên quận/ huyện | Quận Hoàng Mai |
| 6 | city | Category | tỉnh/ thành phố | Hà Nội |
| 7 | ward | Category | Tên xã/ phường | Phường Cổ Nhuế |
| 8 | street | Category | Tên đường | Đường Hoàng Quốc Việt |
| 9 | data\_type | Text | Loại hình tin đăng | Bán nhà riêng |
| 10 | address | Text | Địa chỉ căn nhà |  |
| 11 | lat | Number | Thông tin lat của địa chỉ |  |
| 12 | long | Number | Thông tin long của địa chỉ |  |
| 13 | frontage | Number | Độ rộng mặt tiền |  |
| 14 | street\_width | Number | Độ rộng đường vào |  |
| 15 | house\_direction | Category | Hướng nhà |  |
| 16 | balcony\_direction | Category | Hướng ban công |  |
| 17 | floor | Number | Tổng số tầng. |  |
| 18 | bedroom | Number | Số phòng ngủ |  |
| 19 | toilet | Number | Số toilet |  |
| 20 | post\_date | Date | Ngày đăng tin |  |
| 21 | furniture | Text | Mô tả nội thất |  |
| 22 | description | Text | Mô tả căn nhà |  |
| 23 | land\_certificate | Number | Thông tin pháp lý  1: Sổ đỏ | 1 |

Dựa vào thông tin đặc điểm các thuộc tính thu được ta có thể thấy được những thuộc tính được lựa chọn để có thể đưa vào mô hình học máy phải là những thông tin rõ ràng, miêu tả chính xác đặc điểm căn nhà. Đó chính là các thuộc tính thuộc kiểu dữ liệu number hoặc category. Vì thế, ta lựa chọn được thuộc tính có thể đưa vào mô hình học máy là : *price, area, frontage, street\_width, house\_direction, balcony\_direction, floor, bedroom, toilet, furniture, price\_m2, city, district, lat, long, street, ward, land\_certificate.*

Thống kê tỷ lệ missing value của các thuộc tính trên:



Hình .. Thống kê tỷ lệ missing value của các thuộc tính

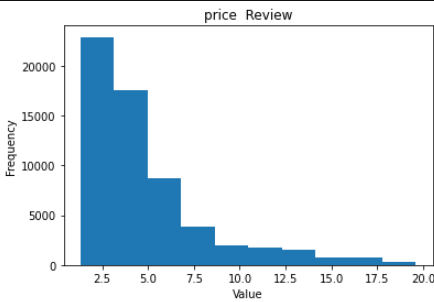
Bảng thống kê giá trị của một số thuộc tính:

Bảng .. Bảng thống kê giá trị một số thuộc tính

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Giá trị  nhỏ nhất | Giá trị  trung vị | Giá trị  trung vị | Giá trị  lớn nhất |
| 1 | Price (tỷ) | 0.00129 | 3.65 | 5.56 | 4700 |
| 2 | Area( | 0.32 | 42 | 52.71 | 14568 |
| 3 | Price\_m2 (tr/) |  | 88.75 | 107.1 | 115000 |
| 4 | Bedroom | 1 | 4 | 4.25 | 150 |
| 5 | Floor | 1 | 5 | 4.52 | 55 |
| 6 | Toilet | 1 | 4 | 3.98 | 117 |
| 7 | Frontage (m) | 1 | 4 | 4.83 | 922 |
| 8 | Street\_width (m) | 0.5 | 3.5 | 4.9 | 300 |

Dựa vào bảng thống kê trên ta có thể thấy được sự chênh lệch giữa giá trị lớn nhất, giá trị nhỏ nhất với giá trị trung bình, trung vị. Kiểm tra thông tin của các bản ghi có giá trị lớn nhất, nhỏ nhất này thấy rằng đây là những lỗi người đăng ghi sai thông tin hoặc đăng những tin không đúng với thể loại nhà riêng nhà ngõ hiểm ví dụ như rao bán cả khách sạn với hàng chục tầng và hàng chục phòng, những căn ở mặt tiền đường lớn có khả năng kinh doanh thương mại lớn.

Phân tích cụ thể giá trị của thuộc tính *Price* thấy rằng có 97 bản ghi có giá lớn hơn 50 tỷ chiếm 0.15% tổng số lượng bản ghi, và . Có 15 bản ghi có giá nhỏ hơn 500 triệu chiếm 0.02%. Điều này thấy rằng số lượng bản ghi có giá quá lớn hoặc quá nhỏ chiếm tỷ lệ rất nhỏ. Ta có biểu đồ thống kê giá trị thuộc tính *Price* với các giá trị trong khoảng phân vị 2% - 98% ta có các biểu đồ sau:

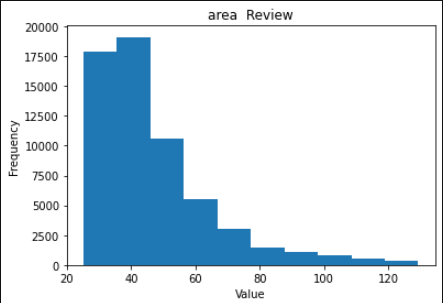


Hình .. Biểu đồ thống kê giá trị thuộc tính Price

trong khoảng phân vị 2%-98%

Nhìn vào hình trên ta có thể thấy được rằng giá trị các căn nhà tập trung trong khoảng 1-5 tỷ đồng.

Với thuộc tính *Area* ta có 112 bản ghi có giá lớn hơn 500 chiếm 0.17% tổng số lượng bản ghi, và . Có 342 bản ghi có giá nhỏ hơn 20 chiếm 0.53%. Ta có biểu đồ thống kê giá trị thuộc tính *Area* với các giá trị trong khoảng phân vị 2%-98% ta có các biểu đồ sau:



Hình .. Biểu đồ thống kê giá trị thuộc tính Price

trong khoảng phân vị 2%-98%

Nhìn vào hình trên ta có thể thấy được rằng diện các căn nhà tập trung trong khoảng 25-60 .

## Xây dựng mô hình học máy cho định giá nhà

### Lọc dữ liệu nhiễu

Dữ liệu nhiều là những bản ghi sai thông tin do người dùng đăng sai, những bản ghi nằm ngoài phạm vị bài toán. Sau bước khám phá dữ liệu chúng ta có thể thấy rằng những bản ghi nhiễu trong bộ dữ liệu thu thập được rơi vào những bản ghi có giá trị của các thuộc tính như *price, area, frontage, street\_width, floor, bedroom, toilet, price\_m2* quá lớn hoặc quá nhỏ.

Như đã trình bày ở trên, phạm vi bài toán là định giá nhà riêng trong ngõ hẻm tại Hà Nội. Chính vì thế chúng ta chỉ giữ lại những bản ghi có đặc điểm của căn nhà như giá và diện tích phù hợp với thực tế. Để thực hiện được việc này chúng ta cần lọc bỏ những căn có giá và diện tích quá lớn hoặc quá nhỏ.

Đối với việc lọc bỏ dữ liệu nhiễu có giá trị quá lớn hoặc quá nhỏ trên các mô hình xử lý dữ liệu thực tế trên thế giới người ta hay sử dụng phương pháp lọc theo phân vị chỉ giữ lại những bản ghi có giá trị trong khoảng phân vị 2% - 98%. Trong quá trình khai phá dữ liệu chúng ta đã tiến hành thống kê dữ liệu trong khoảng phân vị từ 2% - 98% thấy được rằng những bản ghi có giá và diện tích quá lớn hoặc quá nhỏ đã được loại bỏ. Ngoài ra dựa vào Hình 3.12 ta có thể thấy số lượng khuyết thiếu dữ liệu ở các thuộc tính giá, diện tích và cả giá theo mét vuông rất nhỏ chỉ chiếm 1-3% số lượng tổng số lượng bản ghi, chính vì thế trong ta hoàn toàn có thể loại bỏ những dữ liệu khuyết thiếu ở các thuộc tính này và tiến hành lọc bỏ những bản ghi ngoài khoảng phân vị 2-98%. Ta có bảng thống kê giá trị lớn nhất, giá trị nhỏ nhất của một số thuộc tính sau khi tiến hành lọc bỏ :

Bảng .. Thống kê giá trị lớn nhất, nhỏ nhất của các thuộc tính sau khi lọc theo phân vị ở thuộc tính giá và diện tích

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Giá trị nhỏ nhất | Giá trị lớn nhất |
| 1 | Price | 1.4 | 18 |
| 2 | Area | 25 | 120 |
| 3 | Price\_m2 | 13.33 | 593.33 |
| 4 | Floor | 1 | 55 |
| 5 | Bedroom | 1 | 37 |
| 6 | Toilet | 1 | 117 |
| 7 | Frontage | 1 | 612 |
| 8 | Street\_width | 0.5 | 300 |

Từ bảng thống kê trên ta có thể thấy được giá trị của các thuộc tính về giá và diện tích không còn giá trị quá lớn hoặc quá nhỏ tuy nhiên một số thuộc tính khác vẫn còn xuất hiện những giá trị quá lớn hoặc quá nhỏ. Kiểm tra lại thông tin bản ghi có các giá trị quá lớn hoặc quá nhỏ thì ta có một số nhận xét như sau:

Thứ nhất là bản ghi có giá trị lớn ở thuộc tính số toilet là 117 nhưng lại chỉ có 17 phòng ngủ như thế con số này hoàn toàn do người dùng ghi sai và trong bộ dữ liệu cũng chỉ có một bản ghi này có số toilet lớn hơn bất thường. Bản ghi này cần phải loại bỏ.

Thứ hai, đối với các bản ghi số phòng ngủ, số tầng quá lớn. Sau khi kiểm tra lại thông tin những căn nhà này thì thấy rằng những con số này có bản ghi hợp lý có bản ghi bị sai. Tức là những bản ghi hợp lý là những căn nhà lớn được xây nhiều phòng theo kiểu chung cư mini và được rao bán. Còn những bản ghi không hợp lý là những bản ghi do người dùng ghi sai. Điều này gây khó khăn trong quá trình lọc bỏ những bản ghi ghi sai thông tin.

Nhận xét thứ ba là các bản ghi có mặt tiền (frontage) và độ rộng mặt đường (street\_width) có giá trị lớn còn lại rất ít, đó là những thông tin người đăng ghi sai ví dụ như bản ghi có độ rộng mặt tiền lớn nhất là 612 m nhưng trong mô tả con số đó chỉ là 6,12 m. Như vậy bổ sung thêm hai thuộc tính này cũng với hai thuộc tính là giá và diện tích để tiến hành lọc bỏ, chỉ giữ lại những bản ghi trong khoảng phân vị từ 2% đến 98%. Việc lọc bỏ này được tính toán đồng thời cùng một lúc cả bốn thuộc tính trên dữ liệu ban đầu. Tức là chỉ giữ lại những căn có giá, diện tích, mặt tiền và độ rộng mặt đường trong khoảng phân vị 2-98%, vẫn giữ lại những bản ghi bị thiếu thông tin ở hai thuộc tính mặt tiền và độ rộng đường do tỷ lệ missing ở hai thuộc tính này còn lớn, loại bỏ những bản ghi bị thiếu thông tin giá và diện tích vì trên thực tế thì đây là hai thông tin quan trọng nhất của căn nhà và đối với bộ dữ liệu tỷ lệ khuyết thiếu ở hai thuộc tính này là rất ít nên việc loại bỏ sẽ đem lại lợi ích cho mô hình.

Cuối cùng ta được bộ dữ liệu có những thống kê sau:

Bảng .. Thống kê giá trị lớn nhất, giá trị nhỏ nhất của các thuộc tính

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Giá trị nhỏ nhất | Giá trị lớn nhất |
| 1 | Price | 1.4 | 18 |
| 2 | Area | 25 | 120 |
| 3 | Price\_m2 | 13.33 | 593.33 |
| 4 | Floor | 1 | 55 |
| 5 | Bedroom | 1 | 37 |
| 6 | Toilet | 1 | 117 |
| 7 | Frontage | 1 | 9 |
| 8 | Street\_width | 0.5 | 15 |

### Impute missing value

Các phương pháp được lựa chọn để impute missing value là sử dụng giá trị nhỏ nhất, giá trị trung bình và giá trị trung vị. Đây là những phương pháp cơ bản nhất, đơn giản nhất thường được sử dụng. Mỗi phương pháp này sẽ phù hợp với từng bộ dữ liệu Để lựa chọn được phương pháp tốt nhất đối với bộ dữ liệu định giá nhà này chúng ta lần lượt sử dụng các phương pháp này tiến hành xây dựng các mô hình định giá sử dụng cùng một bộ dữ liệu đã được lọc nhiễu, cùng một phương pháp bình thường hóa dữ liệu và mã hóa dữ liệu trên cùng một tập các feature với một vài thuật toán nhất định. Sau đó, chúng ta tiến hành đánh giá chất lượng các mô hình để tìm được mô hình có kết quả tốt nhất và lựa chọn được phương pháp impute missing value phù hợp nhất với mô hình của mình.

### Feature selection

Các đặc trưng của căn nhà được lựa chọn cho mô hình định giá phải là những đặc điểm nổi bật nhất của căn nhà có giá trị đầy đủ, chính xác nhất. Một trong số những tiêu chí đầu tiên để lựa chọn các đặc trưng là dữ liệu là tỷ lệ khuyết thiếu dữ liệu của thuộc tính đó không được vượt quá 50% bởi nếu vượt quá 50% thì giá trị của thuộc tính này không còn ý nghĩa, cho dù sử dụng phương pháp impute missing tối ưu nào đi chăng nữa thì đó vẫn không phải là giá trị thực tế của dữ liệu. Ngoài ra những thuộc tính được lựa chọn trong mô hình phải mang một giá trị rõ ràng, cụ thể và đó là những thuộc tính có cấu trúc. Từ hình 3.12 thống kê tỷ lệ missing value ở các thuộc tính ta cùng với tiêu chí trên ta có thể lựa chọn được các thuộc tính đưa vào mô hình là: *price, area, frontage, street\_width, floor, bedroom, toilet, price\_m2, lat, long, land\_certificate.* Các thuộc tính chỉ địa danh đã được thay thế bằng địa chỉ lat, long và với hi vọng lat, long có thể thể hiện được sự khác nhau về mặt bằng giá cả nhà đặt tại các khu vực

### Chuẩn hóa dữ liệu

Các thuộc được lựa chọn để đưa vào mô hình đều ở dạng số, có thuộc tính *land\_certificate* là biến dạng category nhưng đã được đưa về giá trị 0 và 1.

Đối với việc chuẩn hóa dữ liệu hai phương pháp thường xuyên được sử dụng đó chính là Rescaling data tức là đưa dữ liệu về khoảng [0, 1] và phương pháp z-score standardizes đưa dữ liệu về [-1,1]. Mỗi phương pháp sẽ phù hợp với từng bộ dữ liệu và tưng thuật toán khác nhau. Chính vì thế để lựa chọn được phương pháp phù hợp chúng ta xây dựng hai mô hình sử dụng hai phương pháp khác nhau để lựa chọn thuật toán phù hợp.

### Lựa chọn thuật toán

Các thuật toán được lựa chọn để tiến hành thử nghiệm và lựa chọn ra thuật toán cho kết quả mô hình tốt nhất là Linear Regression, KNN, SVR, Decision Tree và Random Forest.

### Xây dựng các mô hình học máy

#### Phương pháp xây dựng, đánh giá mô hình

Phương pháp xây dựng đánh giá mô hình được sử dụng là phương pháp cross validation. Chúng ta tiến hành chia bộ dữ liệu sau khi được xử lý, lọc nhiều thành hai phần là training set và testing set theo tỷ lên 70:30 một cách ngẫu nhiên. Trong đó tập dữ liệu training set được sử dụng để xây dựng mô hình và testing set để kiểm tra kết quả mô hình. Việc chia dữ liệu thành hai phần để xây dựng và đánh giá chất lượng nhằm kiểm tra hiện tượng mô hình xây dựng được quá khớp so với dữ liệu training nhưng với dữ liệu test thì kết quả sai rất nhiều. Hiện tượng này được gọi là Overfitting.

Bên cạnh đó, chúng ta sử dụng các độ đo để đánh giá chất lượng mô hình là sai số tuyệt đối trung bình MAE (mean absolute error) và phần trăm sai số tuyệt đối trung bình MAPE (mean absolute percentage error). Đây là hai độ đo được tính toán trên kết quả ở cả tập training và tập test ở tất cả các mô hình được xây dựng để lựa chọn đánh giá mô hình.

Một mô hình tốt là một mô hình có kết quả các độ đo MAE và MAPE càng nhỏ càng tốt và không bị hiện tượng overfitting tức sự chênh lệch giữa các độ đô trên tập training và tập testing càng nhỏ càng tốt.

#### Xây dựng các mô hình học máy tìm phương pháp impute missing value phù hợp

Mục tiêu của việc xây dựng các mô hình ở phần này là tìm ra phương pháp impute missing value phù hợp nhất với bộ dữ liệu giá nhà riêng tại Hà Nội. Để thực hiện được việc này ta tiến hành xây dựng các mô hình theo các bước

* Bước 1: Sử dụng bộ dữ liệu training và testing được tách ngẫu nhiên với tỷ lệ 70:30 trên bộ dữ liệu đã lọc nhiễu.
* Bước 2: Sử dụng trên cùng một tập các thuộc tính là *area, frontage, street\_width, floor, bedroom, toilet, lat, long, land\_certificate.*
* Bước 3: Impute missing, sử dụng một trong các phương pháp mode, median và mean.

Bước 4: Sử dụng cùng một phương pháp min/max scale để chuẩn hóa dữ liệu đối với các thuộc tính kiểu number *area, frontage, street\_width, floor, bedroom, toilet, lat, long.*

* Bước 5: Sử dụng phương pháp one hot encoding để mã hóa dữ liệu dạng category là  *land\_certificate.* Thuộc tính này đã được mã hóa.
* Bước 6: Sử dụng 5 thuật toán Linear regression, KNN, Decision tree, SVR, Random Forest để xây dựng mô hình định giá với biến predict là *Price\_m2*. Các thuật toán này sử dụng thư viện scikit-learn và cùng với tham số mặc định của thư viện để tiến hành xây dựng mô hình.
* Bước 7: Tính toán các độ đo trên tập training và testing ứng với từng thuật toán.

Như vậy với các bước như trên, ta tiến hành xây dựng 3 mô hình có sử dụng lần lượt 3 phương pháp impute missing value là sử dụng mean, median, mode từ đó đánh giá lựa chọn phương pháp.

Mô hình 1 sử dụng giá trị yếu vị của thuộc tính (mode) để điền vào giá trị khuyết thiếu. Sau khi tiến hành xây dựng mô hình lần lượt theo các bước trên ta có bảng kết quả sau:

Bảng .. Mô hình sử dụng mode để impute missing value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 26.708358 | 27.028387 | 29.574627 | 29.936583 |
| 2 | KNN | 24.115918 | 19.675426 | 26.490448 | 21.529798 |
| 3 | Decision tree | 18.303688 | 0.613596 | 18.866146 | 0.626899 |
| 4 | SVR | 27.486459 | 27.958970 | 28.135816 | 28.641530 |
| 5 | Random Forest | 14.510183 | 5.821956 | 15.259570 | 6.154740 |

Dựa vào bảng trên ta có thể thấy được thuật toán Random Forest đang có kết quả tốt nhất với sai số 15,26% mặc dù overfitting đang khá cao.

Mô hình 2 sử dụng giá trị trung bình của thuộc tính (mean) để điền vào giá trị khuyết thiếu. Sau khi tiến hành xây dựng mô hình lần lượt theo các bước trên ta có bảng kết quả sau:

Bảng .. Mô hình sử dụng mean để impute missing value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 26.385374 | 26.741687 | 29.114979 | 29.536321 |
| 2 | KNN | 23.812946 | 19.488784 | 26.248476 | 21.462667 |
| 3 | Decision tree | 18.556929 | 0.324837 | 19.120743 | 0.328948 |
| 4 | SVR | 26.723891 | 27.135271 | 27.696687 | 28.133592 |
| 5 | Random Forest | 14.373670 | 5.602794 | 15.133512 | 5.913055 |

Mô hình 3 sử dụng giá trị trung vị của thuộc tính (median) để điền vào giá trị khuyết thiếu. Sau khi tiến hành xây dựng mô hình lần lượt theo các bước trên ta có bảng kết quả sau:

Bảng .. Mô hình sử dụng median để impute missing value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 26.71 | 27.03 | 29.58 | 29.94 |
| 2 | KNN | 24.03 | 19.57 | 26.54 | 21.53 |
| 3 | Decision tree | 18.30 | 0.61 | 18.89 | 0.63 |
| 4 | SVR | 27.01 | 27.42 | 27.89 | 28.28 |
| 5 | Random Forest | 14.50 | 5.81 | 15.25 | 6.13 |

Sau khi có được kết quả của cả 3 mô hình sử dụng phương pháp impute missing value khác nhau ta có biểu đồ so sánh tỷ lệ sai số của các mô hình :

Hình .. Hình so sánh tỷ lệ sai số các mô hình

Dựa vào hình trên ta có thể thấy được tỷ lệ sai số của các mô hình có không có sự chênh lệch với nhau là mấy. Tuy nhiên về mặt số liệu thì phương pháp impute missing value bằng giá trị trung bình vẫn mang lại kết quả tốt hơn đôi chút so với hai mô hình còn lại. Như vậy ta quyết định chọn phương pháp sử dụng giá trị trung bình để impute missing value.

#### Xây dựng các mô hình học máy tìm phương pháp chuẩn hóa dữ liệu

Có hai phương pháp chúng ta đang muốn tiến hành thử nghiệm để tìm ra phương pháp chuấn hóa phù hợp nhất với dữ liệu và với thuật toán là MinMaxScale và Z-score standardizes. Phương pháp sử dụng MinMaxScale đã được xây dựng ở bước trên. Chúng ta sẽ xây dựng mô hình Z-score standardizes với phương pháp impute missing value là giá trị trung bình thu được kết quả như sau:

Bảng .. Kết quả mô hình sử dụng Z-score standardizes chuẩn hóa dữ liệu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 26.38537 | 26.74169 | 29.11498 | 29.53632 |
| 2 | KNN | 19.45883 | 16.04357 | 20.52099 | 16.95177 |
| 3 | Decision tree | 18.54889 | 0.324837 | 19.13955 | 0.328948 |
| 4 | SVR | 20.9062 | 21.34839 | 20.50469 | 20.9945 |
| 5 | Random Forest | 14.32548 | 5.611946 | 15.07279 | 5.924988 |

Với kết quả này ta có thể thấy rõ hai thuật toán KNN và SVR có được kết quả tốt hơn khá nhiều trong khi đó các thuật toán khác thì tỷ lệ sai số gần như không thay đổi. Vậy phương pháp sử dung Z-score standardizes được lựa chọn để chuẩn hóa dữ liệu.

#### Xây dựng các mô hình học máy lựa chọn các đặc trưng đưa vào mô hình

Một số thuộc tính trên thực tế ta nghĩ nó ảnh hướng rất nhiều đến giá trị của căn nhà. Nhưng đối với mô hình học máy nó thực sự có hiệu quả hay không. Chúng ta sẽ lần lượt xây dựng các mô hình mà lược bỏ một số thuộc tính để xác định tầm ảnh hưởng của thuộc tính với mô hình định giá.

Yếu tố đầu tiên mà chúng ta muốn thử nghiệm đó chính là diện tích của căn nhà. Ta có kết quả của mô hình sau khi lược bỏ thuộc tính diện tích căn nhà:

Bảng .. Kết quả mô hình khi lược bỏ thuộc tính area

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 26.51 | 26.85 | 29.13 | 29.51 |
| 2 | KNN | 20.37 | 16.61 | 21.38 | 17.32 |
| 3 | Decision tree | 19.68 | 2.25 | 20.26 | 2.27 |
| 4 | SVR | 21.22 | 21.63 | 20.67 | 21.07 |
| 5 | Random Forest | 16.12 | 7.23 | 16.87 | 7.56 |

Sai số của mô hình đã tăng lên khoảng 2%.

Tiếp theo là thông tin pháp lý của căn nhà tức nhà có sổ đỏ hay không. Trên thực tế, giá trị của một căn nhà có sổ đỏ và một căn nhà chỉ có sổ hồng có thể chênh lệch đến hàng tỷ đồng trên cùng một khu vực. Vậy ta thử lược bỏ thuộc tính này ra khỏi mô hình xem chất lượng mô hình có giảm xuống không. Ta có kết quả của mô hình sau khi lược bỏ thuộc tính *land\_certificate* như sau:

Bảng .. Kết quả mô hình khi lược bỏ thuộc tính làn\_certificate

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 26.40 | 26.75 | 29.10 | 29.52 |
| 2 | KNN | 18.99 | 15.59 | 20.00 | 16.43 |
| 3 | Decision tree | 18.34 | 0.46 | 18.92 | 0.46 |
| 4 | SVR | 20.84 | 21.29 | 20.42 | 20.93 |
| 5 | Random Forest | 14.39 | 5.66 | 15.13 | 5.96 |

Và một thông tin nữa chúng ta muốn thử nghiệm là độ rộng đường. Giá trị của căn nhà ô tô có thể vào được sẽ rất khác so với trong ngõ ngách nhỏ. Đó là kì vọng thực tế còn kết quả mô hình sau khi lược bỏ thuộc tính này như sau:

Bảng .. Kết quả mô hình khi lược bỏ thuộc tính street\_width

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Thuật toán | MAE testing | MAE training | MAPE testing | MAPE training |
| 1 | Linear regression | 28.28 | 28.43 | 31.32 | 31.60 |
| 2 | KNN | 20.48 | 16.78 | 21.56 | 17.72 |
| 3 | Decision tree | 18.67 | 0.46 | 19.24 | 0.49 |
| 4 | SVR | 22.40 | 22.78 | 21.81 | 22.25 |
| 5 | Random Forest | 14.80 | 5.79 | 15.59 | 6.14 |

Ở hai thuộc tính là *land\_certificate* và *street\_width* sai số của mô hình cũng tăng lên nhưng khá là nhỏ.

Từ những thử nghiệm này chúng ta có thể thấy được ảnh hưởng của mỗi thuộc tính là không nhiều nhưng chúng đều giúp cải thiện chất lượng mô hình. Ngoài ra số lượng thuộc tính mà chúng ta có được cũng không lơn chính vì thế chúng ta sẽ sử dụng toàn bộ những thuộc tính có thể để xây dựng mô hình,

#### Xây dựng mô hình học máy cho định giá nhà riêng Hà Nội

Như các mô hình ở trên chúng ta đang xây dựng một mô hình cho toàn thành phố Hà Nội. Trên thực tế thì giá trị căn nhà ở từng khu vực khác nhau rất nhiều đặc biệt là giữa khu vực ngoại thành với khu vực trung tâm, hay đặc biệt có những căn ở khu vực Hoàn Kiếm, hay khu vực phố cổ giá trị của căn nhà cũng lớn hơn rất nhiều so với các khu vực khác. Chính vì thế việc xây dựng một mô hình cho cả Hà Nội như thế sẽ làm giảm rất nhiều chất lượng của mô hình. Chúng ta sẽ xây dựng mỗi quận một mô hình. Điều này vướng phải một vấn đề là không phải quận/ huyện nào cũng có số lượng dữ liệu đủ lớn. Theo như bảng thống kê số lượng dữ liệu theo quận/ huyện ở Hình 3.11 thì dữ liệu phần lớn tập trung ở các quận/ huyện ở trung tâm, nội thành Hà Nội. Điều này cũng dễ hiểu vì nhu cầu cũng như thị trường nhà đất ở các quận nội thành cũng lớn hơn nhiều so với ở các huyện ngoại thành. Mặc dù chúng ta đã có hai thuộc tính lat, long để thể hiện vị trí địa lý của các bản ghi nhưng có vẻ chúng không thực sự hiệu quả. Chúng ta sẽ tiến hành xây dựng mô hình cho các quận/ huyện có số lượng bản ghi sau khi đã lọc bỏ dữ liệu nhiễu phải lớn hơn 100 bản ghi, vì khi xây dựng mô hình trên bộ dữ liệu nhỏ thì mô hình chúng ta xây dựng được hoàn toàn không có ý nghĩa vì độ phủ của dữ liệu không tốt, không trải dài hết được các trường hợp của mô hình.

Thực hiện lọc nhiễu với cách thức thực hiện như thực hiện trên bộ dữ liệu toàn Hà Nội tức là :

* Chỉ giữa lại những bản ghi có đầy đủ giá, diện tích.
* giữ lại những căn có giá, diện tích, mặt tiền và độ rộng mặt đường trong khoảng phân vị 2-98%

Sau khi thực hiện lọc bỏ dữ liệu nhiễu trên bộ dữ liệu từng quận huyện ta có bảng thống kê số lượng dữ liệu còn lại lớn hơn 100 bản ghi ở các quận/ huyện như sau:

Bảng .. Thống kê dữ liệu sau khi lọc nhiễu theo từng quận/ huyện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên quận** | **Số bản ghi** | **Số lượng bản ghi sau khi lọc nhiễu** | **Diện tích**  **Min-Max**  **(m2)** | **Giá**  **Min-Max**  **(tỷ)** |
| 1 | Quận Hà Đông | 10374 | 8825 | 30.0-80.0 | 1.27-7.6 |
| 2 | Quận Đống Đa | 7865 | 7005 | 20.0-120.0 | 1.8-20.5 |
| 3 | Quận Cầu Giấy | 6590 | 5768 | 30.0-130.0 | 2.65-24.0 |
| 4 | Quận Thanh Xuân | 6530 | 5699 | 27.0-130.0 | 2.1-16.0 |
| 5 | Quận Hoàng Mai | 6176 | 5457 | 25.0-89.0 | 1.42-8.5 |
| 6 | Quận Hai Bà Trưng | 5919 | 5253 | 22.0-110.0 | 1.75-12.8 |
| 7 | Quận Ba Đình | 4163 | 3636 | 20.0-120.0 | 1.98-24.0 |
| 8 | Quận Long Biên | 3859 | 3360 | 30.0-120.0 | 1.65-12.5 |
| 9 | Quận Nam Từ Liêm | 3787 | 3286 | 30.0-106.0 | 1.78-14.0 |
| 10 | Quận Tây Hồ | 3095 | 2703 | 30.0-210.0 | 2.3-36.0 |
| 11 | Quận Bắc Từ Liêm | 2033 | 1737 | 30.0-108.0 | 1.65-11.5 |
| 12 | Huyện Thanh Trì | 1415 | 1241 | 30.0-102.0 | 1.25-8.3 |
| 13 | Huyện Hoài Đức | 976 | 845 | 30.0-66.5 | 1.0-3.5 |
| 14 | Quận Hoàn Kiếm | 400 | 328 | 15.0-114.0 | 1.6-38.0 |
| 15 | Huyện Gia Lâm | 305 | 256 | 30.0-155.0 | 1.15-8.55 |
| 16 | Huyện Đông Anh | 182 | 135 | 30.0-290.0 | 0.73-15.0 |

Dựa vào bảng trên ta có thể thấy chỉ còn 16 quận/ huyện số lượng bản ghi còn lại sau khi lọc nhiễu là lớn hơn 100 bản ghi. Các quận/ huyện có số lượng bản ghi còn lại quá ít là: huyện Thường Tín, huyện Mê Linh, huyện Ba Vì, huyện Đan Phượng, huyện Chương Mỹ, huyện Phúc Thọ, huyện Sóc Sơn, huyện Ứng Hòa, Huyện Quốc Oai, huyện Phú Xuyên, huyện Sơn Tây. Tất cả các huyện này đều là các huyện ở ngoại thành Hà Nội.

Mô hình được xây dựng cho từng quận sẽ được thực hiện lần lượt qua các bước sau:

* Bước 1: Chia bộ dữ liệu ban đầu theo từng quận/ huyện. Với mỗi bộ dữ liệu quận/ huyện ta tiến hành tách thành hai tập training và testing một cách ngẫu nhiên với tỷ lệ 70:30 trên bộ dữ liệu đã lọc nhiễu.
* Bước 2: Sử dụng các thuộc tính là đặc điểm của căn nhà: area, frontage, street\_width, floor, bedroom, toilet, lat, long, land\_certificate để xây dựng mô hình cho từng quận.
* Bước 3: Impute missing, sử dụng một trong các phương pháp impute missing value bằng giá trị trung bình như đã lựa chọn từng các mục trên.
* Bước 4: Sử dụng cùng một phương pháp z-score standardizes để chuẩn hóa dữ liệu đối với các thuộc tính kiểu number area, frontage, street\_width, floor, bedroom, toilet, lat, long.
* Bước 5: Đối với thuộc tính Land\_certificate giá trị chỉ có 2 giá trị là 0 hoặc 1 nên không cần chuẩn hóa.
* Bước 6: Sử dụng 5 thuật toán Linear regression, KNN, Decision tree, SVR, Random Forest để xây dựng mô hình định giá với biến predict là Price\_m2. Các thuật toán này sử dụng thư viện scikit-learn và cùng với tham số mặc định của thư viện để tiến hành xây dựng mô hình.
* Bước 7: Tính toán các độ đo trên tập training và testing ứng với từng thuật toán.

Với các bước trên ta tiến hành xây dựng mô hình định giá nhà cho từng quận và thu được kết quả tại một số quận/ huyện như sau:

* Đối với quận có nhiều bản ghi nhất là quận Hà Đông ta có bảng thống kê các độ đo của mô hình tương ứng với từng thuật toán thử nghiệm như sau:

Bảng .. Kết quả mô hình quận Hà Đông

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuật toán | MAE test (triệu) | MAE train (triệu) | MAEP test (%) | MAEP train (%) |
| RandomForest Regressor | 7.4 | 2.92 | 10.46 | 4.23 |
| Decision Tree | 9.05 | 0.36 | 12.43 | 0.53 |
| KNN | 9.86 | 8.19 | 14.15 | 11.99 |
| SVM | 10.54 | 10.92 | 15.11 | 15.73 |
| Linear | 12.21 | 12.43 | 18.4 | 18.82 |

Phần trăm sai số trung bình của thuật toán Random Forest đang cho kết quả tốt nhất và tốt hơn đáng kể so với tỷ lệ sai số trung bình của mô hình toàn Hà Nội.

* Thống kê kết quả mô hình tại quận Nam Từ Liêm:

Bảng .. Kết quả mô hình quận Nam Từ Liêm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuật toán | MAE test (triệu) | MAE train (triệu) | MAEP test (%) | MAEP train (%) |
| RandomForest Regressor | 10.11 | 3.72 | 12.55 | 4.44 |
| KNN | 11.43 | 9.22 | 13.98 | 10.91 |
| Decision Tree | 12.79 | 0.38 | 15.73 | 0.44 |
| SVM | 13.31 | 13.33 | 16.15 | 15.48 |
| Linear | 14.26 | 13.73 | 18.12 | 16.53 |

Thuật toán Random Forest có kết quả tốt nhất đối với mô hình tại quận Nam Từ Liêm.

* Thống kê kết quả mô hình tại quận Hoài Đức :

Bảng .. Kết quả mô hình quận Hoài Đức

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuật toán | MAE test (triệu) | MAE train (triệu) | MAEP test (%) | MAEP train (%) |
| RandomForest Regressor | 3.74 | 1.54 | 8.53 | 3.34 |
| Decision Tree | 4.89 | 0.14 | 11.24 | 0.31 |
| KNN | 6.02 | 4.65 | 13.85 | 9.78 |
| SVM | 6.51 | 6.4 | 14.54 | 13.25 |
| Linear | 6.88 | 6.71 | 15.86 | 14.51 |

Đây là mô hình có kết quả tốt nhất khi tỷ lệ sai số trung bình chỉ còn 8.53%.

* Kết quả mô hình tại quận Đống Đa

Bảng .. Kết quả mô hình quận Đống Đa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuật toán | MAE test (triệu) | MAE train (triệu) | MAEP test (%) | MAEP train (%) |
| RandomForest Regressor | 22.84 | 8.38 | 19.59 | 6.96 |
| KNN | 27.19 | 20.74 | 22.89 | 17.21 |
| Decision Tree | 31 | 29.95 | 23.93 | 23.13 |
| SVM | 29.33 | 0.8 | 25.17 | 0.57 |
| Linear | 33.95 | 33.65 | 27.67 | 27.35 |

Tỷ lệ sai số trung bình ở mô hình sử dụng thuật toán Random Forest có kết quả tốt nhất nhưng lại lơn hơn tỷ lệ sai số trung bình của mô hình toàn Hà Nội sử dụng giá trị trung bình để impute missing value (gần 15%) mà chúng ta đã xây dựng ở trên. Như vậy có phải là chất lượng mô hình bị giảm xuống ? Chúng ta thống kê thêm kết quả mô hình sử dụng thuật toán Random Forest tại các quận/ huyện :

Bảng .. Kết quả các mô hình sử dụng thuật toán Random Forest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Quận/huyện | MAE test (triệu) | MAE train (triệu) | MAEP test (%) | MAEP train (%) |
| 1 | Huyện Hoài Đức | 3.74 | 1.54 | 8.53 | 3.34 |
| 2 | Quận Hà Đông | 7.4 | 2.92 | 10.46 | 4.23 |
| 3 | Quận Bắc Từ Liêm | 10.55 | 3.98 | 12.16 | 4.88 |
| 4 | Quận Nam Từ Liêm | 10.11 | 3.72 | 12.55 | 4.44 |
| 5 | Quận Hoàng Mai | 9.68 | 3.71 | 12.61 | 4.79 |
| 6 | Quận Long Biên | 10 | 3.47 | 13.46 | 4.85 |
| 7 | Huyện Thanh Trì | 7.74 | 3.32 | 13.77 | 5.61 |
| 8 | Quận Hai Bà Trưng | 12.91 | 4.81 | 14.09 | 5.2 |
| 8 | Quận Thanh Xuân | 14.7 | 5.56 | 15.16 | 5.6 |
| 9 | Quận Cầu Giấy | 20.8 | 8.32 | 16.97 | 6.62 |
| 10 | Quận Tây Hồ | 19.92 | 7.58 | 17.61 | 6.83 |
| 12 | Huyện Gia Lâm | 7.97 | 3.1 | 17.85 | 8.17 |
| 13 | Quận Ba Đình | 23.2 | 8.34 | 19.45 | 6.78 |
| 14 | Quận Đống Đa | 22.84 | 8.38 | 19.59 | 6.96 |
| 15 | Quận Hoàn Kiếm | 61.61 | 20.32 | 34.87 | 15.03 |

Thực chất kết quả mô hình các quận/ huyện có chất lượng tốt không giảm xuống mà đang tăng lên giữa việc sử dụng một mô hình để định giá cho toàn Hà Nội và sử dụng mô hình cho từng quận/ huyện Hà Nội. Theo thống kê so sánh ở hình dưới thì tỷ lệ sai số trung bình trên tập test khi sử dụng mô hình toàn Hà Nội để định giá cho bộ dữ liệu test của quận Hoài Đức là 10.69% trong khi sử dụng mô hình riêng của quận Hoài Đức thì tỷ lệ sai số là 8.53%, tương tự đối với quận Nam Từ Liêm đánh giá trên cùng một tập test thì tỷ lệ sai số của mô hình toàn Hà Nội và mô hình của Nam Từ Liêm lần lượt là 12.55% và 13.41%. Tuy thế đối với mô hình quận Cầu Giấy thì tỷ lệ sai số của mô hình Hà Nội và mô hình Cầu Giấy đánh giá trên tập test của quận đó thì kết quả gần bằng nhau. Điều này chứng tỏ việc chia và xây dựng mô hình cho từng quận sẽ giúp cải thiện chất lượng của của quận có kết quả tốt. Đây đa phần là những quận nội thành, có số lượng bản ghi lớn, người dùng lớn, nhu cầu về thị trường cũng lớn.

Bảng .. Bảng so sánh kết quả giữa mô hình toàn Hà Nội và mô hình từng quận/ huyện

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Quận/huyện | MAEP test (%)  Với mô hình từng quận | MAEP test (%)  Với mô hình toàn Hà Nội |
| 1 | Huyện Hoài Đức | 8.53 | 10.69 |
| 2 | Quận Hà Đông | 10.46 | 11.71 |
| 3 | Quận Bắc Từ Liêm | 12.16 | 14.29 |
| 4 | Quận Nam Từ Liêm | 12.55 | 13.41 |
| 5 | Quận Hoàng Mai | 12.61 | 13.44 |
| 6 | Quận Long Biên | 13.46 | 15.40 |
| 7 | Huyện Thanh Trì | 13.77 | 13.95 |
| 8 | Quận Hai Bà Trưng | 14.09 | 14.51 |
| 9 | Quận Thanh Xuân | 15.16 | 15.24 |
| 10 | Quận Cầu Giấy | 16.97 | 16.60 |
| 11 | Quận Tây Hồ | 17.61 | 16.84 |
| 12 | Huyện Gia Lâm | 17.85 | 28.71 |
| 13 | Quận Ba Đình | 19.45 | 17.43 |
| 14 | Quận Đống Đa | 19.59 | 18.20 |
| 15 | Quận Hoàn Kiếm | 34.87 | 28.40 |

Cho dù chất lượng đã được cải thiện tuy nhiên một điểm cần lưu ý ở kết quả của các mô hình trên sự chênh lệch giữa kết quả trên tập train và trên tập test còn khá là cao, các mô hình đang bị hiện tượng overfitting khá cáo. Chính vì thế chúng ta cần phải cải tiển turning các tham số của các mô hình và tiếp tục cải thiện chất lượng của mô hình.

#### Điều chỉnh tham số, cải thiện chất lượng mô hình

Trong việc xây dựng mô hình định giá ở bước trên ngoài việc khẳng định được việc xây dựng mô hình cho từng quận sẽ tốt hơn việc sử dụng mô hình của cả Hà Nội để định giá thì ta còn thấy được rằng trong số 5 thuật toán được đưa vảo thử nghiệm thì hầu hết các mô hình từng quận thuật toán Random Forest có được kết quả tốt nhất. Tuy nhiên nó vẫn còn tồn tại mặt hạn chế là bị overfitting khá lớn, chất lượng của mô hình không được cao. Do chúng ta đang sử dụng những tham số mặc định của thư viện. Chính vì thế chúng ta cần phải tiến hành bước điều chỉnh tham số của mô hình, tìm ra những tham số của thuật toán phù hợp nhất với bộ dữ liệu, với mô hình nhằm cải thiện chất lượng mô hình, hạn chế tối đa hiện tượng overfitting.

Phương pháp để thực hiện điều chỉnh tham số mô hình hay còn được gọi là tunning parameter là phương pháp Grid Search. Phương pháp này lần lượt thử các nhóm tham số đưa vào sao cho thử được hết tất cả các trường hợp và tìm ra nhóm tham số. Trong thuật toán Random forest hai tham số quan trọng cần phải được hiệu chỉnh là số lượng cây con ( n\_estimators) và lượng mẫu tốt thiểu cần có ở một nút lá (min\_samples\_leaf). Sau khi thực hiện tunning parameter bằng phương pháp Grid Search ta thu được kết quả như sau:

Bảng .. Thống kê kết quả sau khi hiệu chỉnh tham số

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Quận/huyện | Best Parameter | MAE  test (triệu) | MAE  train (triệu) | MAEP test  (%) | MAEP train (%) |
| 1 | Huyện Hoài Đức | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 4.49 | 3.65 | 10.24 | 7.77 |
| 2 | Quận Hà Đông | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 7.78 | 5.98 | 11.04 | 8.64 |
| 3 | Quận Nam Từ Liêm | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 10.38 | 7.49 | 12.89 | 8.89 |
| 4 | Quận Bắc Từ Liêm | {'min\_samples\_leaf': 5, 'n\_estimators': 100} | 11.38 | 7.9 | 13.01 | 9.56 |
| 5 | Quận Hoàng Mai | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 10.39 | 7.68 | 13.41 | 9.84 |
| 6 | Quận Long Biên | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 10.64 | 7.12 | 14.17 | 9.82 |
| 7 | Quận Hai Bà Trưng | {'min\_samples\_leaf': 5, 'n\_estimators': 200} | 13.73 | 9.87 | 14.81 | 10.58 |
| 8 | Huyện Thanh Trì | {'min\_samples\_leaf': 5, 'n\_estimators': 200} | 8.32 | 6.75 | 15.15 | 11.48 |
| 9 | Quận Thanh Xuân | {'min\_samples\_leaf': 5, 'n\_estimators': 200} | 15.54 | 11.3 | 16.03 | 11.33 |
| 10 | Quận Cầu Giấy | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 21.79 | 16.78 | 17.7 | 13.31 |
| 11 | Huyện Gia Lâm | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 8.46 | 6.39 | 18.99 | 15.65 |
| 12 | Quận Tây Hồ | {'min\_samples\_leaf': 5, 'n\_estimators': 200} | 22.04 | 15.61 | 19.12 | 13.96 |
| 13 | Quận Đống Đa | {'min\_samples\_leaf': 5, 'n\_estimators': 300} | 24.14 | 17.03 | 20.36 | 14.01 |

Từ bảng thống kê kết quả trên ta có thể thấy được tỷ lệ sai số trên tập test của các mô hình đã bị tăng lên khoảng 1-2% tuy nhiên đổi lại sự chênh lệch kết quả giữa tập train và tập test đã được giảm xuống đáng kể. Điều này sẽ làm tăng mức độ ý nghĩa của mô hình.

Về cơ bản, những con số độ đo trung bình đã cho kết quả khá tốt. Thế nhưng chúng ta còn phải đánh giá chi tiết kết quả của các bản ghi trong một số mô hình để có thể đánh giá chính xác chất lượng của mô hình.

Đầu tiên, với kết quả trên tập test tại của mô hình quận Ba Đình ta kiểm tra những bản ghi có sai số quá lớn:

Bảng .. Kết quả định giá một số bản ghi tại quận Ba Đình

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Street | Area | Price | Price\_m2 | Price\_m2 so với trung vị theo street (%) | Tỷ lệ sai số |
| Phố Kim Mã Thượng | 80 | 2.59 | 32.375 | 22.63986 | 3.058025 |
| Phố Kim Mã Thượng | 70 | 16 | 228.5714 | 159.8402 | 0.389893 |
| Phố Kim Mã Thượng | 70 | 14.5 | 207.1429 | 144.8551 | 0.356116 |
| Phố Kim Mã Thượng | 20 | 4 | 200 | 139.8601 | 0.276621 |
| Phố Kim Mã Thượng | 30 | 4.75 | 158.3333 | 110.7226 | 0.247613 |
| Phố Kim Mã Thượng | 64 | 16 | 250 | 174.8252 | 0.243966 |
| Phố Kim Mã Thượng | 45 | 4.8 | 106.6667 | 74.59207 | 0.230792 |
| Phố Kim Mã Thượng | 50 | 5.65 | 113 | 79.02098 | 0.196868 |
| Phố Kim Mã Thượng | 50 | 5.65 | 113 | 79.02098 | 0.161205 |
| Phố Kim Mã Thượng | 70 | 16 | 228.5714 | 159.8402 | 0.139207 |
| Phố Kim Mã Thượng | 30 | 4.8 | 160 | 111.8881 | 0.136559 |
| Phố Kim Mã Thượng | 36.86 | 5.3 | 143.7873 | 100.5506 | 0.129673 |
| Phố Kim Mã Thượng | 29 | 4.8 | 165.5172 | 115.7463 | 0.11091 |
| Phố Kim Mã Thượng | 56 | 5.8 | 103.5714 | 72.42757 | 0.110215 |
| Phố Kim Mã Thượng | 40 | 4.5 | 112.5 | 78.67133 | 0.100861 |
| Phố Kim Mã Thượng | 70 | 16 | 228.5714 | 159.8402 | 0.071641 |
| Phố Kim Mã Thượng | 25 | 3.9 | 156 | 109.0909 | 0.06816 |
| Phố Kim Mã Thượng | 30 | 3.5 | 116.6667 | 81.58508 | 0.060084 |
| Phố Kim Mã Thượng | 50 | 10.5 | 210 | 146.8531 | 0.044499 |
| Phố Kim Mã Thượng | 46 | 5.5 | 119.5652 | 83.61204 | 0.04214 |
| Phố Kim Mã Thượng | 46 | 5.6 | 121.7391 | 85.13226 | 0.040055 |
| Phố Kim Mã Thượng | 30 | 3.3 | 110 | 76.92308 | 0.036289 |

Trong số các bản ghi trên ta có thể thấy rằng bản ghi có sai số lớn nhất có có giá trên mét vuông quá thấp so với các bản ghi khác trong cùng một khu vực là phố Kim Mã Thượng. Theo tính toán giá trị trung vị của cột price\_m2 tại phố Kim Mã Thượng là 143 triệu/. Như vậy giá theo mét vuông tại bản ghi có tỷ lệ sai số lớn nhất đang chỉ bằng 22.64% so với nhỏ giá trị trung vị.

Tiếp tục đánh giá những bản ghi có sai số quá lớn tại quận Đống Đa:

Bảng .. Kết quả định giá một số bản ghi tại quận Đống Đa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Street | Area | Price | Price\_m2 | Price\_m2 so với trung vị theo street (%) | Tỷ lệ sai số |
| Đường Đặng Tiến Đông | 55 | 1.95 | 35.45 | 31.94 | 3.4046 |
| Đường Đặng Tiến Đông | 30 | 2 | 66.67 | 60.06 | 0.6416 |
| Đường Đặng Tiến Đông | 45 | 4 | 88.89 | 80.08 | 0.4523 |
| Đường Đặng Tiến Đông | 46 | 4.5 | 97.83 | 88.13 | 0.3220 |
| Đường Đặng Tiến Đông | 33 | 4.5 | 136.36 | 122.85 | 0.1774 |
| Đường Đặng Tiến Đông | 90 | 16.5 | 183.33 | 165.17 | 0.1625 |
| Đường Đặng Tiến Đông | 85 | 15.5 | 182.35 | 164.28 | 0.1269 |
| Đường Đặng Tiến Đông | 46 | 4.3 | 93.48 | 84.21 | 0.1237 |
| Đường Đặng Tiến Đông | 94 | 12.5 | 132.98 | 119.80 | 0.0212 |
| Đường Đặng Tiến Đông | 58 | 8 | 137.93 | 124.26 | 0.0182 |
| Đường Đặng Tiến Đông | 45 | 4.5 | 100.00 | 90.09 | 0.0103 |

Cũng tương tự như bản ghi có sai số quá lớn tại quận Ba Đình, bản ghi sai số quá lớn có giá theo mét vuông tại bản ghi có tỷ lệ sai số lớn nhất đang chỉ bằng 31.94% so với nhỏ giá trị trung vị của đường Đường Đặng Tiến Đông.

Kiểm tra các kết quả các quận khác cũng cho kết quả tương tự những bản ghi có sai số quá lớn sẽ có giá trên mét vuông lơn hơn hoặc nhỏ hơn 50% trung vị của đường phố. Chính vì thể ta bổ sung phương pháp lọc những bản ghi này và xây dựng lại mô hình. Sau khi xây dựng lại mô hình ta có kết quả thống kê như sau:

Bảng .. Kết quả sau khi bổ sung lọc nhiễu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quận/huyện | MAE test (triệu) | MAE train (triệu) | MAEP test (%) | MAEP train (%) |
| Huyện Hoài Đức | 4.18 | 3.58 | 9.16 | 7.58 |
| Quận Bắc Từ Liêm | 7.79 | 6.76 | 10.18 | 8.68 |
| Quận Hà Đông | 7.26 | 5.3 | 10.44 | 7.58 |
| Quận Nam Từ Liêm | 8.38 | 6.24 | 10.48 | 7.72 |
| Quận Hai Bà Trưng | 9.75 | 7.35 | 11.56 | 8.63 |
| Quận Hoàng Mai | 9.43 | 7.03 | 11.98 | 9.14 |
| Quận Long Biên | 8.93 | 6.4 | 12.5 | 8.87 |
| Quận Thanh Xuân | 11.92 | 9.22 | 12.62 | 9.88 |
| Huyện Gia Lâm | 6.57 | 6.19 | 14.09 | 13.92 |
| Huyện Thanh Trì | 8.38 | 5.95 | 14.28 | 10.13 |
| Quận Đống Đa | 17.33 | 12.83 | 14.81 | 11.02 |
| Quận Cầu Giấy | 18.22 | 13.19 | 14.98 | 10.84 |
| Quận Tây Hồ | 17.49 | 12.43 | 15.55 | 12.05 |
| Quận Ba Đình | 18.32 | 13.35 | 15.77 | 11.34 |

Ta thấy kết quả tại các quận/ huyện huyện đã được cải thiện đặc biệt là tại các quận/ huyện có sai số ở các mô hình trước còn cao thì đã được cải thiện rõ rết, ví dụ như tại quận Ba Đình và Tây Hồ sai số đã giảm xuống 5%.

## Tổng kết và đánh giá kết quả

Như vậy trong chương này chúng ta đã xây dựng xong mô hình học máy cho định giá nhà riêng tại một số quận/ huyện của Hà Nội. Đó là những quận/ huyện huyện Hoài Đức, quận Bắc Từ Liêm, quận Hà Đông, quận Nam Từ Liêm, quận Hai Bà Trưng, quận Hoàng Mai, quận Long Biên, quận Thanh Xuân, huyện Gia Lâm, huyện Thanh Trì, quận Đống Đa, quận Cầu Giấy, quận Tây Hồ, quận Ba Đình. Đây chính là những quận/ huyện có mô hình định giá có thể sử dụng được bởi nó đảm bảo số lượng bản ghi đưa vào mô hình đủ lớn, chất lượng bản ghi trên tập test có sai số giao động khoảng 10 – 15% và hiện tượng overfit ở mức độ chấp nhận được.

# Chương 4

# XÂY DỰNG CHƯƠNG TRÌNH ĐỊNH GIÁ

## Phân tích yêu cầu

Chương trình định giá nhà sẽ chỉ có một chức năng chính đó chính là định giá nhà. Tức là người dùng sẽ đưa vào các thông số của căn nhà, chương trình sẽ sử dụng mô hình đã được huấn luyện để định giá căn nhà và trả về giá trị của căn nhà. Với một chức năng như thế, định giá nhà sẽ là một module trong một hệ thống. Chính vì thế để có thể sử dụng được module này thì việc tạo một API service để cung cấp dịch vụ định giá nhà là một giải pháp hàng đầu được lựa chọn vì nó dễ dàng tích hợp vào các hệ thống có sẵn ví dụ như website dịch vụ, sàn giao dịch bất động sản hay các ứng dụng trên mobile, desktop khác với đa dạng công nghệ, nền tảng, ngôn ngữ khác nhau.

Công nghệ được lựa chọn cho việc xây dựng API service là framework Flask. Bởi Flask là một framework cho phép chúng ta xây dựng một trang web hay một service đơn giản, nhanh chóng và đặc biệt là viết bằng ngôn ngữ python. Việc này thuận tiện cho chúng ta bơi các model học máy được sử dụng ở trên đều được xây dựng bằng ngôn ngữ Python.

## Xây dựng API định giá

Để xây dựng được API định giá nhà chúng ta cần xác định rõ đầu vào và đầu ra của API service.

Đầu vào của API định giá mà một mảng các dictionary gồm có các tham số là các thuộc tính của mô hình cũng với tham số để định danh xác định loại hình bất động sản định giá :

Bảng .. Các thuộc tính đầu vào của API định giá

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** | **Ví dụ** |
|  | id | String  (Optional) | Id của bản ghi |  |
| 1 | predict\_type (\*) | String  (Required) | Loại định giá | ResidentialHouseTrading |
| 2 | city (\*) | String  (Required) | Tỉnh /Thành phố |  |
| 3 | district(\*) | String  (Required) | Quận/ Huyện |  |
| 4 | street(\*) | String  (Required) | Đường, Phố |  |
| 5 | lat (\*) | Float  (Required) | lat |  |
| 6 | long (\*) | Float  (Required) | long |  |
| 7 | area (\*) | Float  (Required) | diện tích |  |
| 8 | street\_width | Float (Optional) | Độ rộng đường |  |
| 9 | frontage | Float  (Optional) | mặt tiền |  |
| 10 | floor | Integer  (Optional) | Số tầng |  |
| 11 | bedroom | Integer  (Optional) | Số phòng ngủ |  |
| 12 | toilet | Integer  (Optional) | Số toilet |  |

Đầu ra của API định giá là một mảng các dictionary gồm có các tham số thể hiện giá trị định giá của căn nhà cùng với id của bản ghi để xác định định giá cho bản ghi nào:

Bảng .. Danh sách các tham số output của API định giá

|  |  |  |
| --- | --- | --- |
| Thuộc tính | Kiểu giá trị | Ý nghĩa |
| id | String | Id của bản ghi cần định giá |
| price | String | Giá trị mô hình định giá, đơn vị VNĐ |
| text | String | Giá trị căn nhà được định giá ở dạng text ví dụ: "4.595 tỷ" |

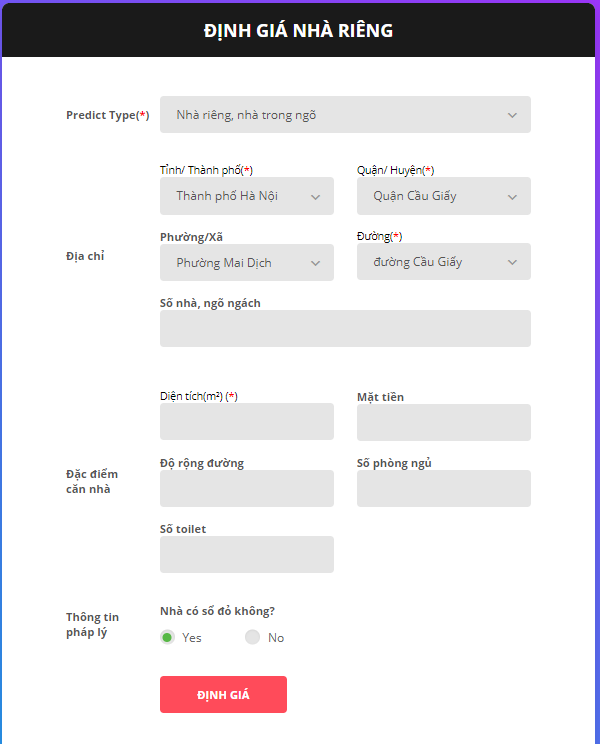
Đầu ra khi API bị lỗi hoặc không định giá được sẽ có dạng:

|  |
| --- |
| [  {  "price": "-1",  "text": "Dữ liệu đang cập nhật",  "id": "5ece32a72d5300000a006645"  }  ] |
|  |

Khi vượt ngoài phạm vi của mô hình định giá hay lỗi do người dùng truyền vào thì API luôn trả về kết quả là -1 cùng dòng thông tin “Dữ liệu đang được cập nhật” điều này sẽ trong suốt đối với người dùng. Điều này sẽ tránh được hiện tượng cố tình nhập sai, thử định giá quá mức.

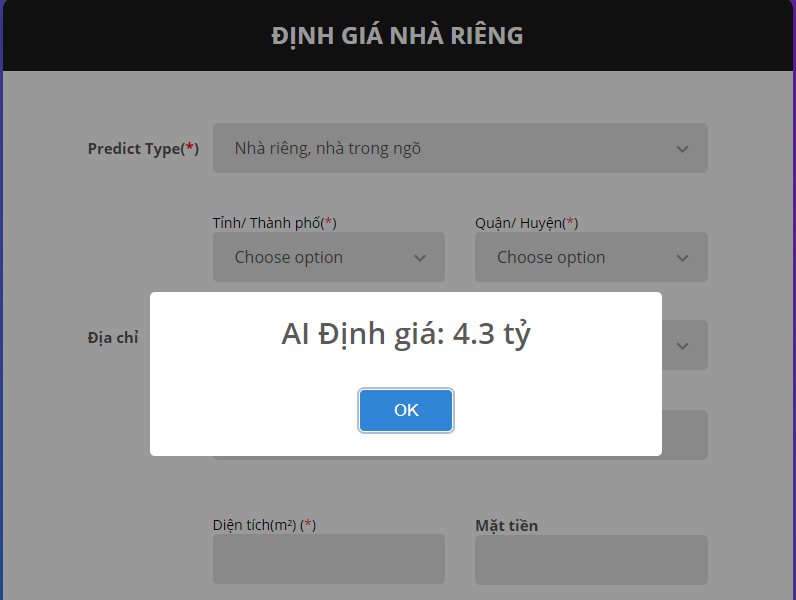
## Xây dựng chương trình minh họa

Chương trình minh họa là một form cho phép người dùng có thể nhập thông tin của căn nhà. Từ những thông tin này chương trình sẽ biến đổi để phù hợp với API định giá và gửi request đến API để định giá căn nhà. Sau khi nhận kết quả từ phía API sẽ được hiển thị cho người dùng.



Hình .. Giao diện chương trình demo API định giá

Kết quả nhận được :



Hình .. Hiển thị kết quả định giá

Ở đây chúng ta sẽ sử dụng từ “AI đinh giá” bởi đối với người dùng phổ thông, khái niện học máy định giá sẽ rất xa vời và khó hiểu đối với người dùng. Còn khi nhắc tới “AI” hay “trí tuệ nhân tạo” người ta sẽ hiểu ngay đây là kết quả của máy tính định lượng. Bởi những khái niện này phổ biến hơn, được truyền thông và đông đảo mọi người nhắc đến nhiều hơn khi quảng cáo sản phẩm.

# KẾT LUẬN

## Kết luận

Những kết quả nghiên cứu và những đóng góp cụ thể trong quá trình làm đồ án đã đạt được như sau:

* Đã xây dựng được bộ dữ liệu giá nhà riêng tại trên địa bàn Hà Nội.
* Đã xây dựng được toàn bộ quy trình xây dựng mô hình định giá nhà riêng tại một khu vực từ thu thập dữ liệu, xử lý, xây dựng mô hình đến cung cấp dịch vụ API định giá.
* Đã xây dựng được mô hình được mô hình định giá nhà riêng cho một số quận, huyện của Hà Nội đáp ứng được phạm vi đề tài đã đặt ra.
* Đã xây dựng được API định giá nhà riêng tại một số quận, huyện Hà Nội. Đây là những quận nội thành nên nhu cầu người sử dụng lớn, đáp ứng được đại đa số người dùng muốn định giá nhà riên tại Hà Nội.
* Đã xây dựng được chương trình demo API định giá cho phép người dùng nhập liệu thông tin căn nhà để định giá nhà.

Tuy nhiên đề tài vẫn còn tồn tại một số điểm hạn chế, khó khăn như:

* Độ chính xác của các mô hình chỉ ở mức trung bình 10 -15%
* Khó khăn trong việc xử lý bộ dữ liệu thực tế vì phụ thuộc vào tính chính xác của người dùng đăng tin.
* Thời gian có hạn nên mới định giá được ở một khu vực và một loại hình nhất định

## Hướng phát triển

Hướng phát triển của đề tài là:

* Tiếp tục mở rộng phạm vi đề tài về cả khu vực định giá lẫn loại hình bất động sản định giá.
* Tìm hiểu nghiên cứu các phương pháp khác để nâng cao chất lượng của mô hình định giá

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Tiêu chuẩn thẩm định giá Việt Nam số 02 Giá trị thị trường làm cơ sở cho thẩm định giá (Ký hiệu: TĐGVN 02), “Ban hành kèm theo Thông tư số 158/2014/TT-BTC ngày 27 tháng 10 năm 2014 của Bộ trưởng Bộ Tài chính”. |
| [2] | Dean De Cock, "Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project," *Journal of Statistics Education,* vol. 19, 2011. |
| [3] | Byeonghwa Park, Jae Kwon Bae, "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data," *Expert Systems with Applications,* vol. 42, pp. 2928-2934, 2015. |