

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

------------------------------------

**BÁO CÁO ĐỒ ÁN 3**

**ĐỀ TÀI:**

**DỰ ĐOÁN GIÁ NHÀ/ ĐẤT**

**SỬ DỤNG THUẬT TOÁN LINEAR REGRESSION**

**GVHD:** Ph.D Nguyễn Thiên Bảo

**SVTH:**

17110226 – Lê Thanh Thảo

17110147 – Huỳnh Xuân Huy

TP. HỒ CHÍ MINH – 01/2021

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

------------------------------------

**BÁO CÁO ĐỒ ÁN 3**

**ĐỀ TÀI:**

**DỰ ĐOÁN GIÁ NHÀ/ ĐẤT**

**SỬ DỤNG THUẬT TOÁN LINEAR REGRESSION**

**GVHD:** Ph.D Nguyễn Thiên Bảo

**SVTH:**

17110226 – Lê Thanh Thảo

17110147 – Huỳnh Xuân Huy

TP. HỒ CHÍ MINH – 01/2021

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

----\*\*\*----

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên: Lê Thanh Thảo MSSV: 17110226

Huỳnh Xuân Huy MSSV: 17110147

Ngành: Công nghệ thông tin

Tên đề tài: Dự đoán giá nhà/đất sử dụng thuật toán Linear Regression

Họ và tên Giáo viên hướng dẫn: Ph.D Nguyễn Thiên Bảo

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

.....................................................................................................................................................................................................................................................................................................................................................................................................................................................................

2. Ưu điểm:

...................................................................................................................................................................................................................................................................................................................................................................................................................................................

3. Khuyết điểm:

.............................................................................................................................................................................................................................................................................................................. ....................................................................................................................................

5. Đánh giá loại:

.................................................................................................................................................

.................................................................................................................................................

6.Điểm:……………….(Bằngchữ: .......................................................................................)

Tp*. Hồ Chí Minh, ngày tháng năm 2021*

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc61088484)

[DANH MỤC HÌNH ẢNH ii](#_Toc61088485)

[DANH MỤC BẢNG iii](#_Toc61088486)

[PHẦN MỞ ĐẦU 1](#_Toc61088487)

[1.1 Lý do chọn đề tài 1](#_Toc61088488)

[1.2 Mục tiêu nghiên cứu 1](#_Toc61088489)

[1.3 Phạm vi nghiên cứu 1](#_Toc61088490)

[1.4 Đối tượng nghiên cứu 1](#_Toc61088491)

[1.5 Phương pháp nghiên cứu 1](#_Toc61088492)

[PHẦN NỘI DUNG 2](#_Toc61088493)

[2.1 Phương pháp thẩm định giá nhà [1] 2](#_Toc61088494)

[2.2 Thuật toán Linear Regression [2] 3](#_Toc61088495)

[2.2.1 Giới thiệu 3](#_Toc61088496)

[2.2.2 Ưu điểm 6](#_Toc61088497)

[2.2.3 Nhược điểm 6](#_Toc61088498)

[2.2.4 Ứng dụng của Linear Regression trong thực tế 7](#_Toc61088499)

[2.3 Dataset Housing Price [3] 7](#_Toc61088500)

[2.3.1 Giới thiệu 7](#_Toc61088501)

[2.3.2 Cấu trúc [4] 7](#_Toc61088502)

[2.4 Cài đặt thuật toán 10](#_Toc61088503)

[2.4.1 Các thư viện được sử dụng 10](#_Toc61088504)

[2.4.2 Cài đặt và kiểm thử 16](#_Toc61088505)

[KẾT LUẬN 25](#_Toc61088506)

[TÀI LIỆU THAM KHẢO 26](#_Toc61088507)

# LỜI CẢM ƠN

*TPHCM, tháng 01 năm 2021*

Trong thời gian thực hiện đồ án 3, nhóm xin chân thành cảm ơn thầy Nguyễn Thiên Bảo đã tận tình hỗ trợ cho chúng em về vật chất cũng như tinh thần, để chúng em có thể hoàn thành đồ án một cách tốt nhất

Với mỗi sinh viên chúng em nói chung, việc tích lũy kiến thức qua giáo trình, bài giảng trên lớp rất quan trọng và cần thiết, tuy nhiên, việc áp dụng các kiến thức vào thực tế chưa thực sự nhiều. Cho nên việc thực hiện đồ án 3 là một cơ hội để nhóm có thể áp dụng các kiến thức mà mình đã học vào giải quyết các vấn đề thực tiễn.

Vì kiến thức và thời gian thực hiện còn hạn chế, nên trong quá trình thực hiện không tránh xảy ra sai sót. Nhóm rất mong thầy đóng góp ý kiến để hoàn thiện đồ án tốt hơn.

Chúng em xin chân thành cảm ơn thầy!

# DANH MỤC HÌNH **ẢNH**

[Hình 1: Tổng quan về Scikit-learn 11](#_Toc61088220)

[Hình 2: Demo thuật toán phân cụm Clustering 13](#_Toc61088221)

[Hình 3: Quy trình làm việc trong Seaborn 15](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088222)

[Hình 4: Train-set ban đầu 16](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088223)

[Hình 5: Tóm tắt thống kê mô tả 17](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088224)

[Hình 6: Biểu đồ thể hiện tỷ lệ 17](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088225)

[Hình 7: Bản đồ nhiệt thể hiện tương quan với giá nhà 18](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088226)

[Hình 8: Ma trận tương quan giá 18](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088227)

[Hình 9: Phân tán tính năng quan trọng 20](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088228)

[Hình 10: Thể hiện % dữ liệu bị thiếu 21](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088229)

[Hình 11: Chuẩn hóa dữ liệu 22](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088230)

[Hình 12: Tỷ lệ giữa giá nhà và diện tích sống 22](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088231)

[Hình 13: Độ chính xác của Model 23](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088232)

[Hình 14: Độ mất mát của model 24](https://d.docs.live.net/2a6480cd7ef0d4d7/BÁO%20CÁO%20project%203.docx#_Toc61088233)

# DANH MỤC BẢNG

[Bảng 1: Thuộc tính của dataset 7](#_Toc61088234)

# PHẦN MỞ ĐẦU

## Lý do chọn đề tài

* Hiện nay, thị trường Bất động sản ngày càng phát triển, với nhiều loại hình đầu tư và xây dựng khác nhau. Ngày càng xuất hiện các chủ đầu tư với các dự án lớn và quy mô rộng khắp Việt Nam. Điều này dẫn tới giá trị của các dự án Bất động sản ngày càng có sự biến động nhiều theo thời gian, khiến cho các nhà đầu tư cần có các phương pháp và công cụ để dự đoán giá trị của các dự án mà họ quan tâm, từ đó đưa ra kế hoạch đầu tư hợp lý và lâu dài.
* Nhận thấy thực tế đó, nhóm quyết định lựa chọn đề tài dự đoán giá nhà dựa trên thuật toán của Machine Learning.

## Mục tiêu nghiên cứu

* Hiểu được một số phương pháp và các yêu tố liên quan đến thẩm định giá Bất động sản
* Hiểu được thuật toán Linear Regression và cách ứng dụng vào việc giải quyết bài toán dự đoán giá nhà.
* Đánh giá thuật toán Linear Regression thông qua việc đánh giá kết quả thu được khi áp dụng thuật toán vào bài toán.

## Phạm vi nghiên cứu

* Phương pháp thẩm định và các yếu tố liên quan đến giá nhà
* Thuật toán Linear Regression
* Ứng dụng thuật toán vào giải quyết bài toán
* Cách đánh giá kết quả và thuật toán

## Đối tượng nghiên cứu

* Các yếu tố thẩm định, dự đoán giá nhà đất
* Thuật toán Linear Regression

## Phương pháp nghiên cứu

* Đặt vấn đề thực tế
* Tìm hướng giải quyết
* Ứng dụng thuật toán Machine Learning

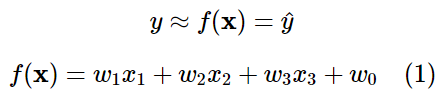
# PHẦN NỘI DUNG

## Phương pháp thẩm định giá nhà [1]

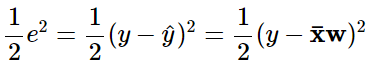
* Hiện nay có 5 phương pháp phổ biến được áp dụng
  + Phương pháp so sánh trực tiếp:
    - Phương pháp này dựa trên cơ sở giá đã được bán trên thị trường của các bất động sản có nhiều đặc điểm tương đối giống so với nhà đất đang xem xét. Để ước tính giá trị thực của tài sản, đây là phương pháp sử dụng rộng rãi và phổ biến nhất hiện nay ở Việt Nam cũng như nhiều nước trên thế giới.
  + Phương pháp thu nhập
    - Phương pháp này dựa trên cơ sở thu nhập ròng trung bình hằng năm trong tương lai từ một bất động sản, tương ứng với tỷ lệ (%) thu hồi vốn nhất định (còn gọi là tỷ lệ vốn hóa) để tính ra giá trị của tài sản.
    - Mặt hạn chế của phương pháp này là các tham số để tính toán giá trị bất động sản đòi hỏi độ chính xác cao, trong khi việc xác định chúng lại phải tiến hành trong điều kiện dự kiến trước. Vì vậy độ chính xác của kết quả tính toán thường bị hạn chế. Trong trường hợp kết quả kiểm tra có sự chênh lệch lớn thì phải áp dụng các phương pháp khác để tính toán bổ sung.
  + Phương pháp giá thành
    - Phương pháp này chủ yếu được áp dụng để định giá những bất động sản không có hoặc rất ít khi xảy ra việc mua bán trên thị trường (nhà thờ, trường học, bệnh viện, công sở…).
    - Dựa trên nguyên tắc thay thế, phương pháp giá thành cho phép giả định rằng, giá trị của một tài sản hiện có có thể đo bằng chi phí làm ra một tài sản tương tự có vai trò như là một vật thay thế, nghĩa là giá trị của khu đất thay thế cộng với chi phí xây dựng hiện hành.
  + Phương pháp lợi nhuận
    - Phương pháp này được sử dụng để xác định giá của các tài sản đặc biệt như rạp chiếu phim, khách sạn và những tài sản khác mà giá trị của nó chủ yếu phụ thuộc vào khả năng sinh lời từ tài sản đó.
    - Mặt hạn chế của phương pháp này là chỉ áp dụng để xác định giá trị cho những bất động sản mà hoạt động của nó có tạo ra lợi nhuận
  + Phương pháp thặng dư
    - Phương pháp này thường được áp dụng để tính toán giá trị của những bất động sản không phải theo hiện trạng sử dụng mà căn cứ vào mục đích sẽ được sử dụng chúng trong tương lai. Chẳng hạn như căn cứ theo quy hoạch sẽ được cấp có thẩm quyền phê duyệt.
    - Thực chất phương pháp thặng dư là một dạng của phương pháp giá thành. Chúng thực hiện theo nguyên tắc: giá trị đất đai được xác định trên cơ sở giá trị còn lại sau khi lấy giá trị công trình bất động sản trừ đi tổng số chi phí và lợi nhuận.

## Thuật toán Linear Regression [2]

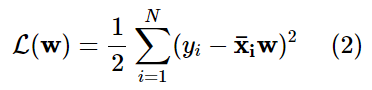
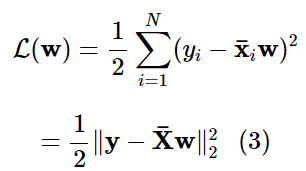
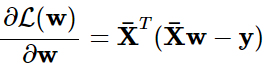
### Giới thiệu

* Ví dụ một căn nhà rộng x1 m2, có x2 phòng ngủ và cách trung tâm thành phố x3 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 y=f(x)sẽ có dạng như thế nào. Ở đây x=[x1,x2,x3] 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).
* **Lưu ý về ký hiệu toán học:** trong các bài viết của tôi, các số vô hướng được biểu diễn bởi các chữ cái viết ở dạng không in đậm, có thể viết hoa, ví dụ x1,N,y,k. Các vector được biểu diễn bằng các chữ cái thường in đậm, ví dụ y,x1. Các ma trận được biểu diễn bởi các chữ viết hoa in đậm, ví dụ X,Y,W
* Một cách đơn giản nhất, chúng ta có thể thấy rằng: i) diện tích nhà càng lớn thì giá nhà càng cao; ii) số lượng phòng ngủ càng lớn thì giá nhà càng cao; iii) 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, w1,w2,w3,w0 là các hằng số, w0 còn được gọi là bias. Mối quan hệ y≈f(x) 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 { w1,w2,w3,w0} chính vì vậy được gọi là bài toán Linear Regression.
* **Chú ý 1:** ***y*** là giá trị thực của outcome (dựa trên số liệu thống kê chúng ta có trong tập training data), trong khi y^ là giá trị mà mô hình Linear Regression dự đoán được. Nhìn chung, y và y^ là hai giá trị khác nhau do có sai số mô hình, tuy nhiên, chúng ta mong muốn rằng sự khác nhau này rất nhỏ.
* **Chú ý 2:** Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Trong không gian nhiều hơn 3 chiều, khái niệm mặt phẳng không còn phù hợp nữa, thay vào đó, một khái niệm khác ra đời được gọi là siêu mặt phẳng (hyperplane). Các hàm số tuyến tính là các hàm đơn giản nhất, vì chúng thuận tiện trong việc hình dung và tính toán. Chúng ta sẽ được thấy trong các bài viết sau, tuyến tính rất quan trọng và hữu ích trong các bài toán Machine Learning. Kinh nghiệm cá nhân tôi cho thấy, trước khi hiểu được các thuật toán phi tuyến (non-linear, không phẳng), chúng ta cần nắm vững các kỹ thuật cho các mô hình tuyến tính.

#### Phân tích toán học

* + Dạng của Linear Regression
* Trong phương trình (1) phía trên, nếu chúng ta đặt [w0,w1,w2,w3]**T**= là vector (cột) hệ số cần phải tối ưu và x¯=[1,x1,x2,x3] (đọc là x bar trong tiếng Anh) là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:
  + Sai số dự đoán
    - Chúng ta mong muốn rằng sự sai khác e giữa giá trị thực y và giá trị dự đoán y^ (đọc là y hat trong tiếng Anh) là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt:

trong đó hệ số 1/2 (lại) là để thuận tiện cho việc tính toán (khi tính đạo hàm thì số 1/2 sẽ bị triệt tiêu). Chúng ta cần e2 vì e=y−y^ có thể là một số âm, việc nói e nhỏ nhất sẽ không đúng vì khi e=−∞ là rất nhỏ nhưng sự sai lệch là rất lớn.

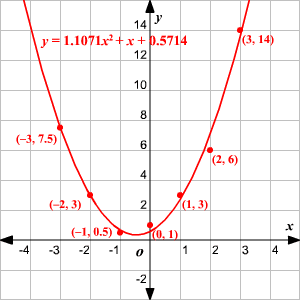
* Hàm mất mát
* Điều tương tự xảy ra với tất cả các cặp (input, outcome)  (xi,yi),i=1,2,…,N, với N là số lượng dữ liệu quan sát được. Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm **w** để hàm số sau đạt giá trị nhỏ nhất:
* Hàm số L(w) được gọi là **hàm mất mát** (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số ww sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của ww làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point), ký hiệu:
* Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát (2). Đặt y=[y1;y2;…;yN] là một vector cột chứa tất cả các output của training data; X¯=[x¯1;x¯2;…;x¯N] là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát L(w) được viết dưới dạng ma trận đơn giản hơn:
* Với ∥z∥2 là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác ∥z∥2/2 là tổng của bình phương mỗi phần tử của vector z. Tới đây, ta đã có một dạng đơn giản của hàm mất mát được viết như phương trình (3).
  + Nghiệm cho bài toán Linear Regression:
* Đạo hàm theo ww của hàm mất mát là
* Phương trình đạo hàm bằng 0 tương đương với:



* + - Nếu ma trận vuông khả nghịch (non-singular hay invertible) thì phương trình (4) có nghiệm duy nhất: w=A−1b.
* Với khái niệm giả nghịch đảo, điểm tối ưu của bài toán Linear Regression có dạng:

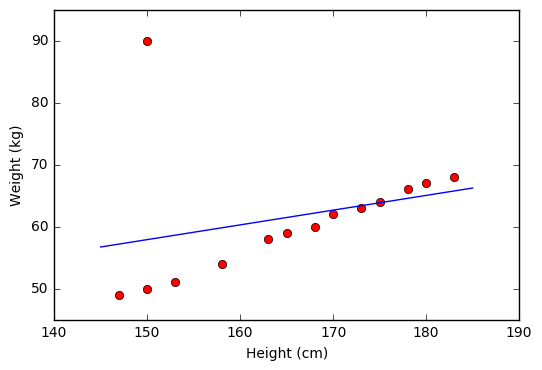
### Ưu điểm

* Hàm số y≈f(x)=wT là một hàm tuyến tính theo cả w*w* và x. Trên thực tế, Linear Regression có thể áp dụng cho các mô hình chỉ cần tuyến tính theo w. Ví dụ:y≈w1x1+w2x2+w3x2+w4sin(x2)+w5x1x2+w0*+w4sin⁡(x2)+w5x1x2+w0* là một hàm tuyến tính theo w và vì vậy cũng có thể được giải bằng Linear Regression. Với mỗi dữ liệu đầu vào x=[x1;x2], chúng ta tính toán dữ liệu mới x=[x1,x2,x21,sin(x2),x1x2] rồi áp dụng Linear Regression với dữ liệu mới này.
* Xem thêm ví dụ về [Quadratic Regression](http://www.varsitytutors.com/hotmath/hotmath_help/topics/quadratic-regression) (Hồi Quy Bậc Hai).

</a>

### Nhược điểm

* + Hạn chế đầu tiên của Linear Regression là nó rất nhạy cảm với nhiễu (sensitive to noise). Trong ví dụ về mối quan hệ giữa chiều cao và cân nặng bên trên, nếu có chỉ một cặp dữ liệu *nhiễu* (150 cm, 90kg) thì kết quả sẽ sai khác đi rất nhiều. Xem hình dưới đây:



* Vì vậy, trước khi thực hiện Linear Regression, các nhiễu (*outlier*) cần phải được loại bỏ. Bước này được gọi là tiền xử lý (pre-processing).
* Hạn chế thứ hai của Linear Regression là nó không biễu diễn được các mô hình phức tạp. Mặc dù trong phần trên, chúng ta thấy rằng phương pháp này có thể được áp dụng nếu quan hệ giữa *outcome* và *input* không nhất thiết phải là tuyến tính, nhưng mối quan hệ này vẫn đơn giản nhiều so với các mô hình thực tế.

### Ứng dụng của Linear Regression trong thực tế

* Tính giá nhà đất
* Tính trung bình số cân nặng chiều cao của mọi người

## Dataset Housing Price [3]

### Giới thiệu

Dataset Housing Price là một dataset đa dạng các thuộc tính thuộc về đối tượng nhà/ căn hộ. Với sự đa dạng về các thông tin, sẽ cho ra được các kết quả khác nhau và giúp người dùng có những đánh giá đa chiều về đối tượng mà mình đang quan sát.

### Cấu trúc [4]

#### Thuộc tính

Bảng 1: Thuộc tính của dataset

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Mô tả** |
| 1 | **SalePrice** | The property's sale price in dollars. This is the target variable that you're trying to predict. |
| 2 | **MSSubClass** | The building class |
| 3 | **MSZoning** | The general zoning classification |
| 4 | **LotFrontage** | Linear feet of street connected to property |
| 5 | **LotArea** | Lot size in square feet |
| 6 | **Street** | Type of road access |
| 7 | **Alley** | Type of alley access |
| 8 | **LotShape** | General shape of property |
| 9 | **LandContour** | Flatness of the property |
| 10 | **Utilities** | Type of utilities available |
| 11 | **LotConfig** | Lot configuration |
| 12 | **LandSlope** | Slope of property |
| 13 | **Neighborhood** | Physical locations within Ames city limits |
| 14 | **Condition1** | Proximity to main road or railroad |
| 15 | **Condition2** | Proximity to main road or railroad (if a second is present) |
| 16 | **BldgType** | Type of dwelling |
| 17 | **HouseStyle** | Style of dwelling |
| 18 | **OverallQual** | Overall material and finish quality |
| 19 | **OverallCond** | Overall condition rating |
| 20 | **YearBuilt** | Original construction date |
| 21 | **YearRemodAdd** | Remodel date |
| 22 | **RoofStyle** | Type of roof |
| 23 | **RoofMatl** | Roof material |
| 24 | Exterior1st | Exterior covering on house |
| 25 | Exterior2nd | Exterior covering on house (if more than one material) |
| 26 | MasVnrType | Masonry veneer type |
| 27 | MasVnrArea | Masonry veneer area in square feet |
| 28 | ExterQual | Exterior material quality |
| 29 | ExterCond | Present condition of the material on the exterior |
| 30 | Foundation | Type of foundation |
| 31 | BsmtQual | Height of the basement |
| 32 | BsmtCond | General condition of the basement |
| 33 | BsmtExposure | Walkout or garden level basement walls |
| 34 | BsmtFinType1 | Quality of basement finished area |
| 35 | BsmtFinSF1 | Type 1 finished square feet |
| 36 | BsmtFinType2 | Quality of second finished area (if present) |
| 37 | BsmtFinSF2 | Type 2 finished square feet |
| 38 | BsmtUnfSF | Unfinished square feet of basement area |
| 39 | TotalBsmtSF | Total square feet of basement area |
| 40 | Heating | Type of heating |
| 41 | HeatingQC | Heating quality and condition |
| 42 | CentralAir | Central air conditioning |
| 43 | Electrical | Electrical system |
| 44 | 1stFlrSF | First Floor square feet |
| 45 | 2ndFlrSF | Second floor square feet |
| 46 | LowQualFinSF | Low quality finished square feet (all floors) |
| 47 | GrLivArea | Above grade (ground) living area square feet |
| 48 | BsmtFullBath | Basement full bathrooms |
| 49 | BsmtHalfBath | Basement half bathrooms |
| 50 | FullBath | Full bathrooms above grade |
| 51 | HalfBath | Half baths above grade |
| 52 | Bedroom | Number of bedrooms above basement level |
| 53 | Kitchen | Number of kitchens |
| 54 | KitchenQual | Kitchen quality |
| 55 | TotRmsAbvGrd | Total rooms above grade (does not include bathrooms) |
| 56 | Functional | Home functionality rating |
| 57 | Fireplaces | : Number of fireplaces |
| 58 | FireplaceQu | Fireplace quality |
| 59 | GarageType | Garage location |
| 60 | GarageYrBlt | Year garage was built |
| 61 | GarageFinish | Interior finish of the garage |
| 62 | GarageCars | Size of garage in car capacity |
| 63 | GarageArea | Size of garage in square feet |
| 64 | GarageQual | Garage quality |
| 65 | GarageCond | Garage condition |
| 66 | PavedDrive | Paved driveway |
| 67 | WoodDeckSF | Wood deck area in square feet |
| 68 | OpenPorchSF | Open porch area in square feet |
| 69 | EnclosedPorch | Enclosed porch area in square feet |
| 70 | 3SsnPorch | Three season porch area in square feet |
| 71 | ScreenPorch | Screen porch area in square feet |
| 72 | PoolArea | Pool area in square feet |
| 73 | PoolQC | Pool quality |
| 74 | Fence | Fence quality |
| 75 | MiscFeature | Miscellaneous feature not covered in other categories |
| 76 | MiscVal | $Value of miscellaneous feature |
| 77 | MoSold | Month Sold |
| 78 | YrSold | Year Sold |
| 79 | SaleType | Type of sale |
| 80 | SaleCondition | Condition of sale |

## Cài đặt thuật toán

### Các thư viện được sử dụng

#### **Numpy**

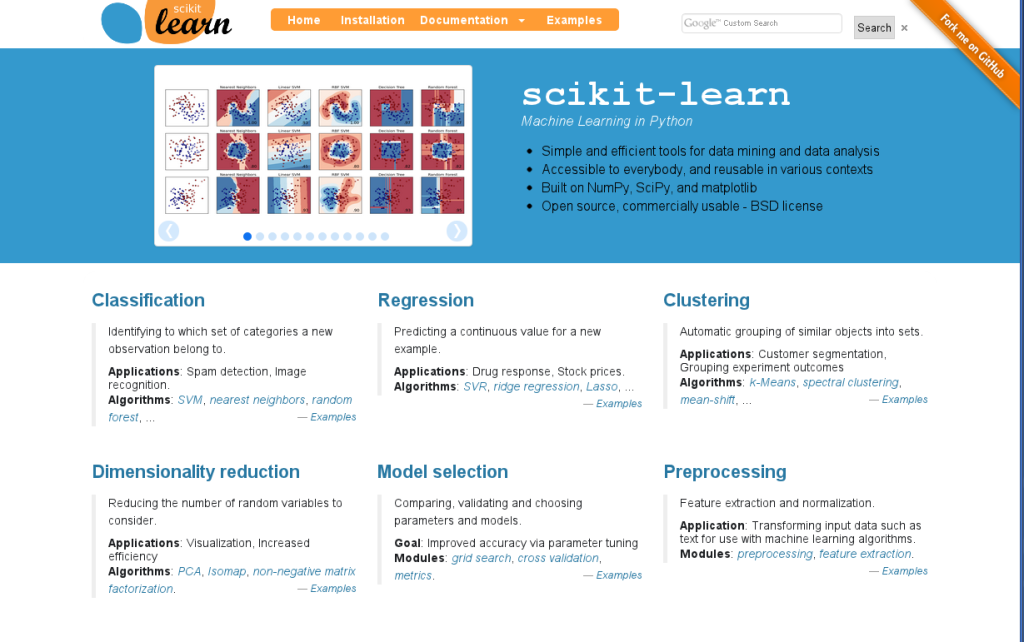
* Numpy là gì: là một thư viện toán học phổ biến và mạnh mẽ nhất của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi sử dụng “core Python” đơn thuần
* Cài đặt thư viện Numpy: pip install numpy

#### **Pandas**

* Pandas là một thư viện mã nguồn mở được xây dựng dựa trên Numpy, sử dụng thao tác và phân tích dữ liệu, được thiết kế để cho phép bạn làm việc với dữ liệu được gắn nhãn hoặc quan hệ theo cách trực quan hơn
* Có thể xử lý tập dữ liệu khác nhau về định dạng: chuỗi thời gian, bảng không đồng nhất, ma trận dữ liệu
* Khả năng import dữ liệu từ nhiều nguồn khác nhau như CSV, DB/SQL
* Có thể xử lý vô số phép toán cho tập dữ liệu: subsetting, slicing, filtering, merging, groupBy, re-ordering, and re-shaping,..
* Xử lý dữ liệu mất mát theo ý người dùng mong muốn: bỏ qua hoặc chuyển sang 0
* Xử lý, phân tích dữ liệu tốt như mô hình hoá và thống kê
* Tích hợp tốt với các thư viện khác của python
* Cung cấp hiệu suất tốt
* Để cài đặt pandas nếu bạn có Anaconda chỉ cần gõ conda install pandas hoặc sử dụng tools pip: pip install pandas.
* Sau khi cài đặt xong, trong Python, chúng ta cần khai báo import pandas để có thể bắt đầu sử dụng các hàm của pandas. Vì pandas là thư viện được sử dụng thường xuyên nên nó thường được khai báo gọn lại thành pd import pandas as pd pd có thể thay thế bằng các từ khác, tuy nhiên bạn nên đặt là pd vì các tài liệu hướng dẫn đều ngầm quy ước như thế
* Pandas có ba cấu trúc dữ liệu và nó được xây dựng dựa trên thư viện Numpy vậy nên chúng hoạt động rất nhanh và hiệu quả: **Series**, **DataFrame**, **Panel**. Trong đó **Panel** là mảng 3 chiều. **Panel** thì không được sử dụng rãi như như Series hay DataFrame và nó cũng không dễ hiển thị hay trừu tượng hoá như màn một chiều và hai chiều, nên dưới đây mình chỉ giới thiệu **Series** và **DataFrame**

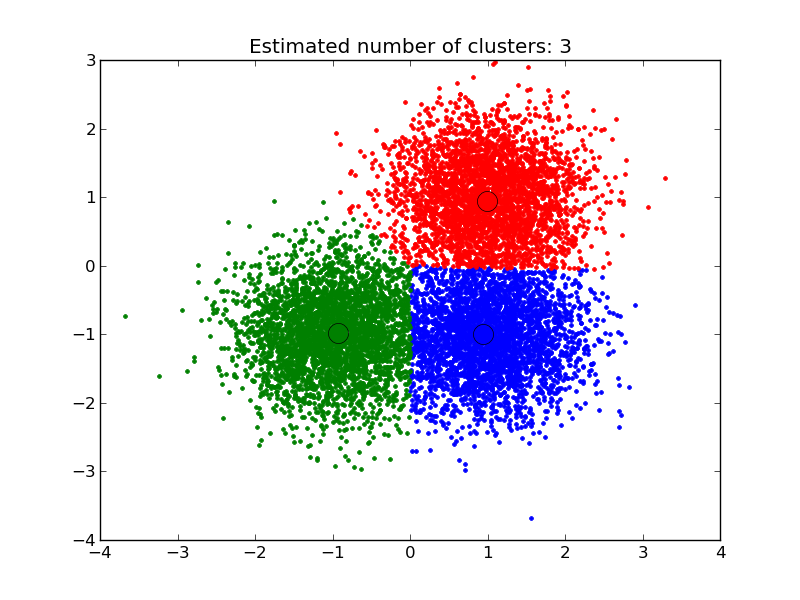
#### **Sklearn**

* Thư Viện Scikit-learn Trong Python Là Gì?
  + Nếu bạn đang sử dụng Python và đang muốn tìm một thư viện mạnh mẽ mà bạn có thể mang các thuật toán học máy (machine learning) vào trong một hệ thống thì không còn thư viện nào thích hợp hơn scikit-learn. Thư viện này tích hợp rất nhiều thuật toán hiện đại và cố điển giúp bạn vừa học vừa tiến hành đưa ra các giải pháp hữu ích cho bài toán của bạn một cách đơn giản.
* Thư viện này được hình thành như thế nào?
* Scikit-learn ban đầu được đề xuất bởi David Cournapeau trong một dự án mùa hè của Google vào năm 2007.
* Later Matthieu Brucher tham gia dự án trên và bắt đầu sử dụng nó làm một phần luận văn tiến sĩ của ông ấy. Vào năm 2010, INRIA bắt đầu tài trợ và phiên bản đầu tiên được xuất bản (v0.1 beta) vào cuối tháng 1 năm 2010.
* Dự án vẫn đang được nghiên cứu bởi một đội ngũ hơn 30 nhà nghiên cứu đến từ các công ty lớn INRIA, Google, Tinyclues và Python Software Foundation.



Hình 1: Tổng quan về Scikit-learn

* + - **Scikit-learn là gì?**
* Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.
* Thư viện được cấp phép bản quyền chuẩn FreeBSD và chạy được trên nhiều nền tảng Linux. Scikit-learn được sử dụng như một tài liệu để học tập.
* Để cài đặt scikit-learn trước tiên phải cài thư viện SciPy (Scientific Python). Những thành phần gồm:
* [**Numpy**](https://codelearn.io/sharing/tim-hieu-thu-vien-numpy-trong-python): Gói thư viện xử lý dãy số và ma trận nhiều chiều
* **SciPy**: Gói các hàm tính toán logic khoa học
* [**Matplotlib**](https://codelearn.io/sharing/ve-bieu-do-voi-thu-vien-matplotlib-p1): Biểu diễn dữ liệu dưới dạng đồ thị 2 chiều, 3 chiều
* **IPython**: Notebook dùng để tương tác trực quan với Python
* **SymPy**: Gói thư viện các kí tự toán học
* [**Pandas**](https://codelearn.io/sharing/xu-ly-du-lieu-voi-pandas-trong-python): Xử lý, phân tích dữ liệu dưới dạng bảng
* Những thư viện mở rộng của SciPy thường được đặt tên dạng SciKits. Như thư viện này là gói các lớp, hàm sử dụng trong thuật toán học máy thì được đặt tên là scikit-learn.
* Scikit-learn hỗ trợ mạnh mẽ trong việc xây dựng các sản phẩm. Nghĩa là thư viện này tập trung sâu trong việc xây dựng các yếu tố: dễ sử dụng, dễ code, dễ tham khảo, dễ làm việc, hiệu quả cao.
* Mặc dù được viết cho Python nhưng thực ra các thư viện nền tảng của scikit-learn lại được viết dưới các thư viện của C để tăng hiệu suất làm việc. Ví dụ như: Numpy(Tính toán ma trận), LAPACK, LibSVM và Cython.
* Nhóm thuật toán
* Thư viện tập trung vào việc mô hình hóa dữ liệu. Nó không tập trung vào việc truyền tải dữ liệu, biến đổi hay tổng hợp dữ liệu. Những công việc này dành cho thư viện Numpy và Pandas.



Hình 2: Demo thuật toán phân cụm Clustering

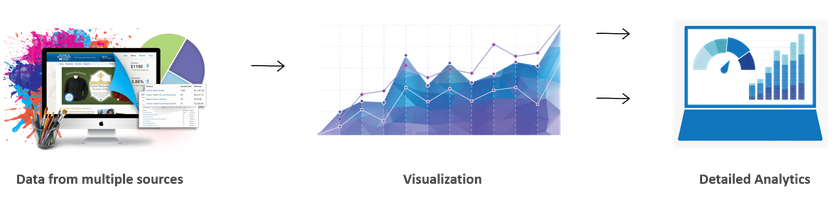
* Hình trên lấy từ một bản demo của [thuật toán phân cụm](http://scikit-learn.org/stable/auto_examples/cluster/plot_mean_shift.html)
* Sau đây là một số nhóm thuật toán được xây dựng bởi thư viện scikit-learn:
* **Clustering**: Nhóm thuật toán Phân cụm dữ liệu không gán nhãn. Ví dụ thuật toán KMeans
* **Cross Validation**: Kiểm thử chéo, đánh giá độ hiệu quả của thuật toán học giám sát sử dụng dữ liệu kiểm thử (validation data) trong quá trình huấn luyện mô hình.
* **Datasets**: Gồm nhóm các Bộ dữ liệu được tích hợp sẵn trong thư viện. Hầu như các bộ dữ liệu đều đã được chuẩn hóa và mang lại hiêu suất cao trong quá trình huấn luyện như iris, digit, ...
* **Dimensionality Reduction**: Mục đích của thuật toán này là để Giảm số lượng thuộc tính quan trọng của dữ liệu bằng các phương pháp như tổng hợp, biểu diễn dữ liệu và lựa chọn đặc trưng. Ví dụ thuật toán PCA (Principal component analysis).
* **Ensemble methods**: Các Phương pháp tập hợp sử dụng nhiều thuật toán học tập để có được hiệu suất dự đoán tốt hơn so với bất kỳ thuật toán học cấu thành nào.
* **Feature extraction**: Trích xuất đặc trưng. Mục đích là để định nghĩa các thuộc tình với dữ liệu hình ảnh và dữ liệu ngôn ngữ.
* **Feature selection**: Trích chọn đặc trưng. Lựa chọn các đặc trưng có ý nghĩa trong việc huấn luyện mô hình học giám sát.
* **Parameter Tuning**: Tinh chỉnh tham số. Các thuật toán phục vụ việc lựa chọn tham số phù hợp để tối ưu hóa mô hình.
* **Manifold Learning**: Các thuật toán học tổng hợp và Phân tích dữ liệu đa chiều phức tạp.
* **Supervised Models**: Học giám sát. Mảng lớn các thuật toán học máy hiện nay. Ví dụ như linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines và decision trees.

#### **Matplotlib**

* Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở.
* Để cài đặt Matplotlib nếu bạn có Anaconda chỉ cần gõ conda install matplotlib hoặc sử dụng tools pip pip install matplotlib
* Khái niệm chung: Một Matplotlib figure có thể được phân loại thành nhiều phần như dưới đây:
* **Figure:** Như một cái cửa sổ chứa tất cả những gì bạn sẽ vẽ trên đó.
* **Axes:** Thành phần chính của một figure là các axes (những khung nhỏ hơn để vẽ hình lên đó). Một figure có thể chứa một hoặc nhiều axes. Nói cách khác, figure chỉ là khung chứa, chính các axes mới thật sự là nơi các hình vẽ được vẽ lên.
* **Axis:** Chúng là dòng số giống như các đối tượng và đảm nhiệm việc tạo các giới hạn biểu đồ.
* **Artist:** Mọi thứ mà bạn có thể nhìn thấy trên figure là một artist như Text objects, Line2D objects, collection objects. Hầu hết các Artists được gắn với Axes.
* Bắt đầu với Pyplot:
* Pyplot là một module của Matplotlib cung cấp các hàm đơn giản để thêm các thành phần plot như lines, images, text, v.v. vào các axes trong figure.
* Tạo một biểu đồ đơn giản: import matplotlib.pyplot as pltimport numpy as np

#### **Seaborn**

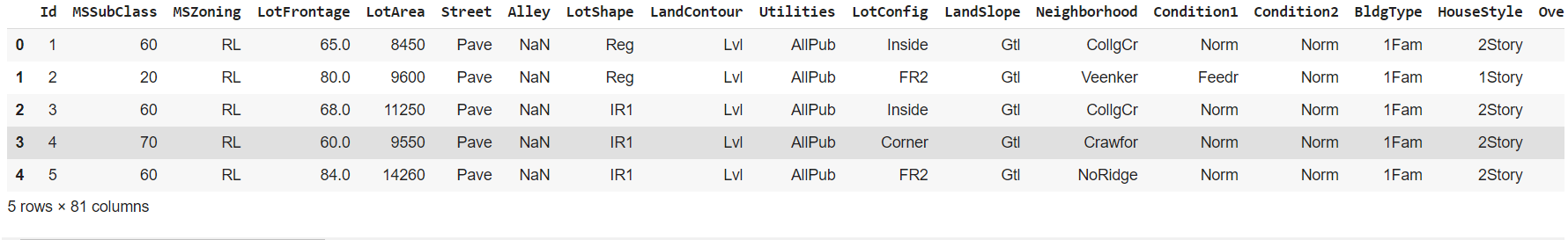
* Phân tích dữ liệu đã phát triển ổn định kể từ đầu thế kỷ 21. Các nhà phát triển và doanh nghiệp cũng như đã biết giá trị của dữ liệu và những lợi ích hiệu quả mà nó có thể mang lại nếu nó được phân tích và sử dụng tốt. Hầu như tất cả các doanh nghiệp ngày nay đều sử dụng công cụ trực quan hóa dữ liệu theo cách này hay cách khác.
* Những công cụ phổ biến nhất trong số những công cụ này là Seaborn và Matplotlib. Đây là hai trong số các thư viện trực quan hóa dữ liệu phổ biến mà người ta có thể sử dụng khi làm việc với một ngôn ngữ lập trình như Python.
* **Giới thiệu về Seaborn**
* Seaborn là một trong những thư viện Python được đánh giá cao nhất thế giới được xây dựng nhằm mục đích tạo ra các hình ảnh trực quan đẹp mắt. Nó có thể được coi là một phần mở rộng của một thư viện khác có tên là Matplotlib vì nó được xây dựng trên đó.
* Trực quan hóa dữ liệu được thực hiện dễ dàng trong Seaborn và đây là cách quy trình



Hình 3: Quy trình làm việc trong Seaborn

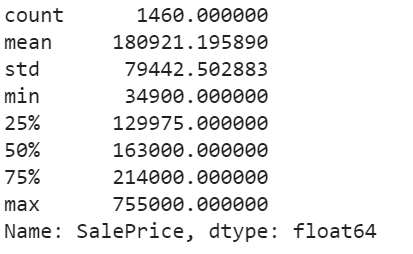
* Dữ liệu từ nhiều nguồn khác nhau: Dữ liệu cần thiết để thực hiện trực quan hóa và phân tích có thể đi vào kiến trúc từ nhiều nguồn khác nhau, chẳng hạn như đơn vị lưu trữ cục bộ, máy chủ, cấu trúc đám mây, v.v.
* Trực quan hóa dữ liệu: Đây là nơi dữ liệu được chuyển đổi từ trạng thái số của nó thành một đối tác trực quan đẹp mắt về mặt thẩm mỹ. Seaborn đóng vai trò chính ở đây.
* Phân tích dữ liệu : Kết quả của trực quan hóa dữ liệu là xem xét dữ liệu theo cách bạn chưa làm trước đây. Phân tích chỉ giúp làm điều này để tiết lộ thông tin chi tiết và xu hướng mà không thể được phát hiện bằng cách khác.
* Quy trình làm việc này rất quan trọng vì đây là chuỗi sự kiện giúp thúc đẩy nhiều loại hình kinh doanh và các yêu cầu của họ đạt được mục tiêu của họ.
* Thư viện Seaborn chắc chắn là phổ biến, nhưng tôi chắc chắn rằng bạn đang thắc mắc về việc tại sao mọi người lại sử dụng Python cho việc này. Tiếp theo trong Hướng dẫn Seaborn dành cho người mới bắt đầu này.
* Đặc điểm của Seaborn
* Trước khi chúng ta nói về các tính năng của Seaborn, điều quan trọng cần biết là: Vì Seaborn là một phần mở rộng của Matplotlib, nó có nghĩa là để bổ sung cho phần sau chứ không phải thay thế hoàn toàn.
* Với điều này đã nói, đây là các tính năng hàng đầu của Seaborn:
* Rất nhiều chủ đề hoạt động với đồ họa kiểu Matplotlib
* Khả năng trực quan hóa cả dữ liệu đơn biến và đa biến
* Hỗ trợ trực quan hóa các loại dữ liệu mô hình hồi quy
* Dễ dàng vẽ biểu đồ dữ liệu thống kê cho phân tích chuỗi thời gian
* Hiệu suất liền mạch với Pandas, NumPy và các thư viện Python khác

### Cài đặt và kiểm thử

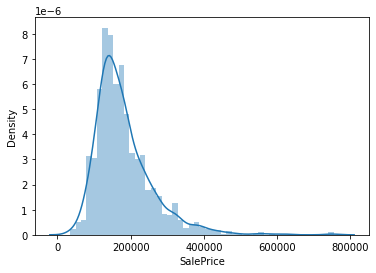
* Train-set ban đầu

Hình 4: Train-set ban đầu

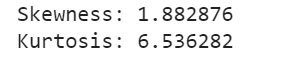
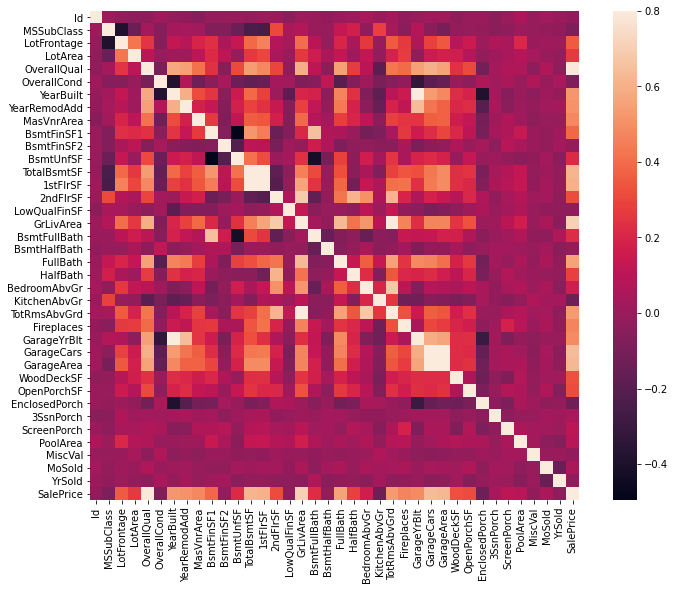
* Chuẩn bị data
  + Điều tra dữ liệu nào có quan hệ tuyến tính hoặc một số loại liên quan đến giá ưu đãi
  + Bỏ các tính năng không quan trọng hoặc các tính năng ít quan trọng hơn
  + Bỏ các tính năng có nhiều giá trị NaN
* Tóm tắt thống kê mô tả



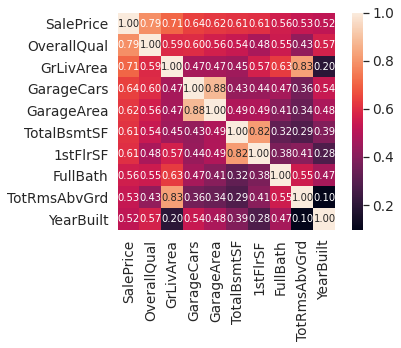
Hình 5: Tóm tắt thống kê mô tả

* Biểu đồ

Hình 6: Biểu đồ thể hiện tỷ lệ

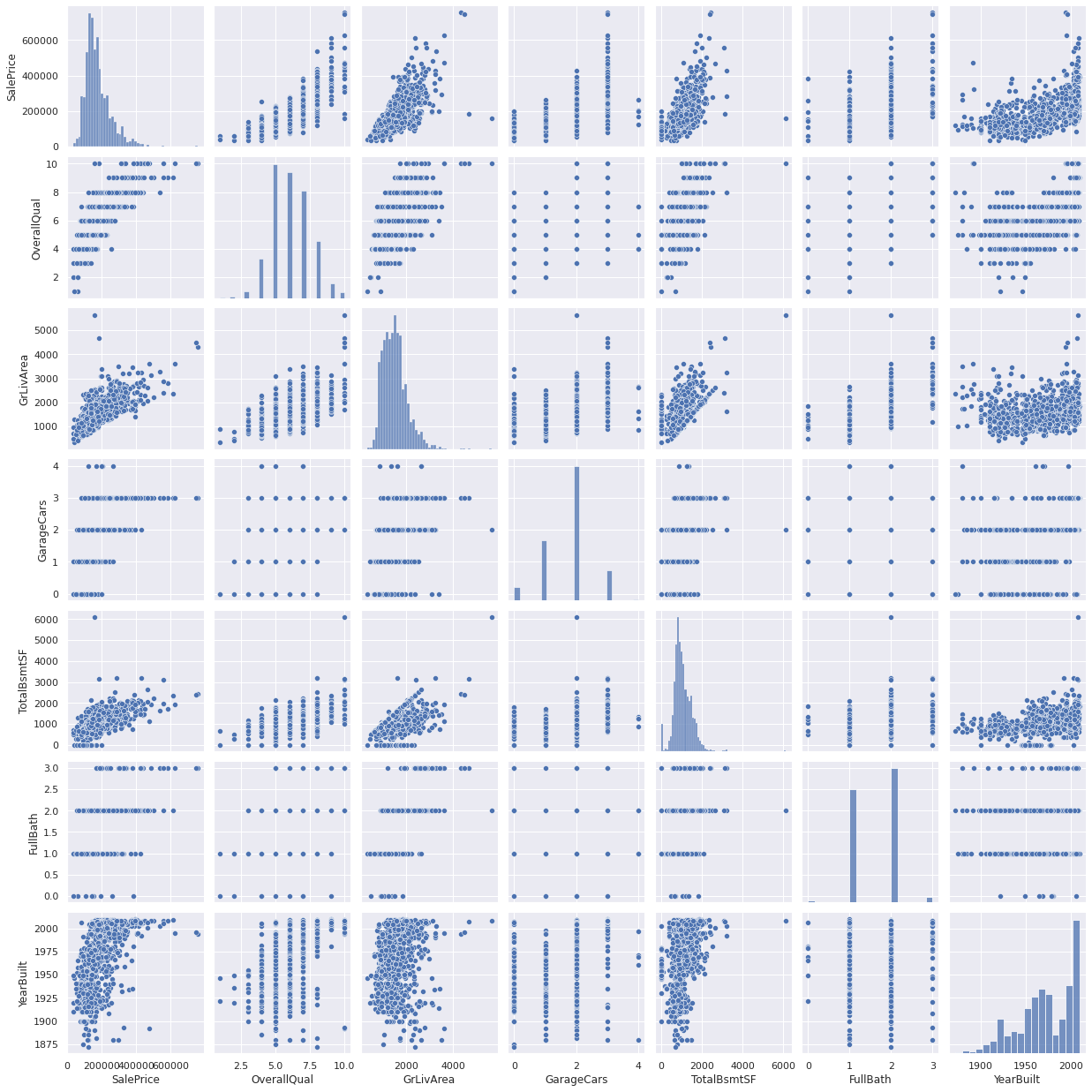
* Skewness có nghĩa là đỉnh của tảng băng không nằm ở giữa mà là hướng sang trái hoặc phải.
* Kurtosis mô tả nếu độ phân giải gaussian rất nhỏ và hẹp hoặc rất rộng
* Sử dụng bản đồ nhiệt để xem tính năng nào có mối tương quan mạnh nhất với giá nhà

Hình 7: Bản đồ nhiệt thể hiện tương quan với giá nhà

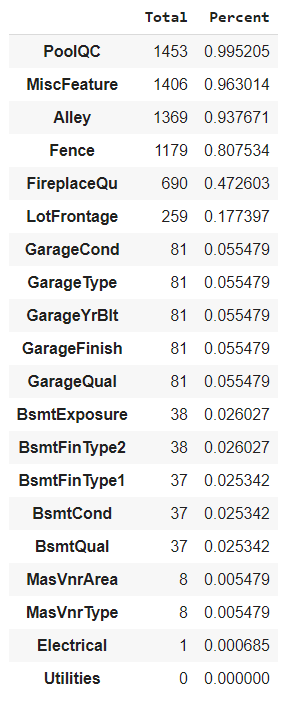
* Ở đây, chúng ta có thể phát hiện đa cộng tuyến, ví dụ như diện tích tầng hầm và diện tích tầng đầu tiên để chúng chứa nhiều hơn hoặc ít hơn cùng một loại dữ liệu. Tương tự đối với các biến số nhà để xe, ví dụ nếu bạn có một nhà để xe lớn, bạn cũng có nhiều ô tô hơn trong đó.
* Một số biến số cũng quan trọng đối với Giá bán với biến số lớn nhất là Tổng thể
* Ma trận tương quan giá

Hình 8: Ma trận tương quan giá

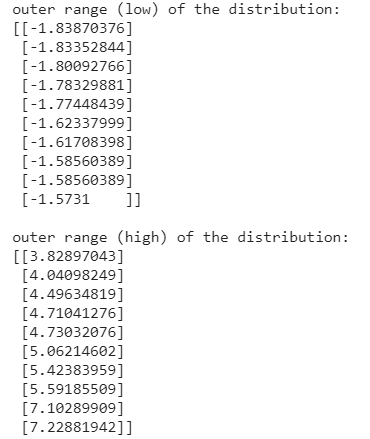
Dựa vào biểu đồ trên, ta có một số kết luận như sau:

* + Nhìn chungQual là quan trọng
  + GrLivArea cũng quan trọng
  + TotalBsmtSF rất quan trọng
  + GarageCars và GarageArea là hai tính năng quan trọng nhưng chúng tôi loại bỏ GarageArea vì nó ít nhiều có cùng thông tin với GarageCars
  + TotalBsmtSF và 1stFlrSF cũng ít nhiều giống nhau nên chúng ta bỏ 1StFlrSF
  + TotRmsAbvGrd và GrLivArea cũng có mối tương quan chặt chẽ với nhau để bỏ TotRmsAbvGrd
* Phân tán các tính năng quan trọng

Hình 9: Phân tán tính năng quan trọng

* Diện tích tầng hầm và tổng diện tích sinh hoạt dường như có những điểm giống nhau, lô đất của họ trông gần như giống hệt nhau, hãy bỏ diện tích tầng hầm.
* Cũng có thể xóa dữ liệu năm xây dựng vì dữ liệu này có thể khó sử dụng.
* Xem dữ liệu bị thiếu
* Hiển thị % một số dữ liệu bị thiếu trong một số cột

Hình 10: Thể hiện % dữ liệu bị thiếu

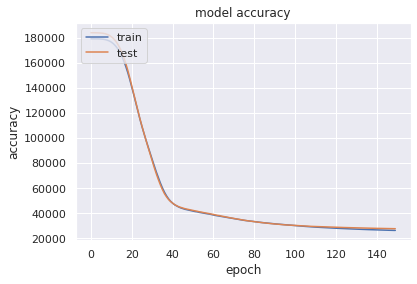
* + Tạo ra dữ liệu trung bình cho một số tính năng bị thiếu hụt dữ liệu
  + Bây giờ, chúng ta hãy loại bỏ các ngoại lệ, ví dụ như dữ liệu không khớp với những gì chúng ta mong đợi, chẳng hạn như một mức giá điên rồ cho một ngôi nhà
  + Để làm điều này, chúng tôi chuẩn hóa dữ liệu để giá trị trung bình là 0 và độ lệch chuẩn là 1.
*  Giá trị tương tự nhau luôn gần bằng 0

Hình 11: Chuẩn hóa dữ liệu

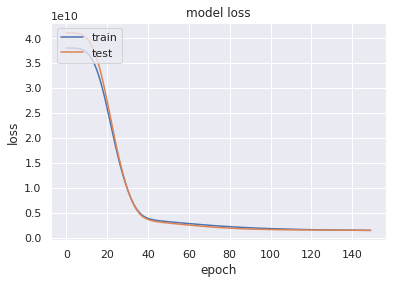
* Giá trị hơi lẻ nhận giá trị cao như 7 giá trị.
* Tỉ lệ giữa giá nhà và diện tích sống

Hình 12: Tỷ lệ giữa giá nhà và diện tích sống

* + - Hai giá trị có 'GrLivArea' lớn hơn có vẻ lạ và chúng không theo sau đám đông. Chúng ta có thể suy đoán tại sao điều này lại xảy ra. Có thể nói đến khu vực nông nghiệp và điều đó có thể giải thích cho giá thấp. Hai điểm này không phải là đại diện cho trường hợp điển hình. Kết luận: xác định chúng là ngoại lệ và xóa chúng.
    - Hai quan sát ở phần trên cùng của biểu dồ là 7 quan sát nên cẩn thận. Chúng trông giống như hai trường hợp đặc biệt, tuy nhiên chúng dường như đang theo xu hướng. Kết luận: giữ lại feature này
* Chuẩn bị data cho lần tiếp theo



Hình 13: Độ chính xác của Model



Hình 14: Độ mất mát của model

# KẾT LUẬN

Với mục tiêu ban đầu là ứng dụng thuật toán Linear Regression vào bài toán dự đoán giá nhà / đất. Nhóm nhận thấy đã thực hiện được khoảng 90%. Đưa ra được một model hợp lý. Accuracy đạt giá trị cao. Có thể nhận xét được các thuộc tính liên quan đến giá nhà.

Sau khi hoàn thành đồ án, nhóm nhận thấy mình đã hiểu hơn về Machine learning nói chung, và thuật toán Linear Regression nói riêng. Biết cách xử lý dữ liệu cho hợp lý, áp dụng thuật toán, phân tích lỗi và tối ưu model để cho ra kết quả tốt nhất.

# TÀI LIỆU THAM KHẢO

[1] Phương pháp thẩm định giá nhà . Retrieved from: <https://www.hqa.com.vn/5-phuong-phap-tham-dinh-gia-dat-dai.html>

[2]Mô hình hồi quy ứng dụng trong bài toán dự đoán giá bất động sản – machine learning. Retrieved from: <https://dothanhlong.org/mo-hinh-hoi-quy-ung-dung-trong-bai-toan-du-doan-gia-bat-dong-san-machine-learning-phan-2/>

[3] House-prices-prediction using keras. Retrieved from: <https://www.kaggle.com/hugosjoberg/house-prices-prediction-using-keras>

[4] Dataset housing price. Retrieved from: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>