BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

ĐỒ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI:

PHÂN TÍCH VÀ DỰ BÁO GIÁ NHÀ BẰNG PHƯƠNG PHÁP HỒI QUY

Khoa : **KHOA HỌC CƠ BẢN**

Ngành : **TOÁN ỨNG DỤNG**

Chuyên ngành : **TOÁN-TIN ỨNG DỤNG**

Giảng viên hướng dẫn : PGS.TS TRẦN VĂN LONG

Sinh viên thực hiện : TRẦN VĂN TIẾN

MSV : 203010478

Khóa : 61

Hà Nội, năm 2024

MỤC LỤC

DANH MỤC HÌNH ẢNH

LỜI MỞ ĐẦU

Trong thế giới ngày nay, với sự phát triển nhanh chóng của công nghệ và dữ liệu, việc áp dụng các phương pháp khoa học để giải quyết các vấn đề thực tiễn trở nên cần thiết hơn bao giờ hết. Một trong những vấn đề thực tiễn quan trọng mà chúng ta thường xuyên gặp phải là dự báo giá nhà - một yếu tố quan trọng ảnh hưởng đến cả nền kinh tế và quyết định cá nhân của mỗi người khi mua, bán hoặc đầu tư vào bất động sản. Với bối cảnh đó, việc tìm kiếm một phương pháp hiệu quả để phân tích và dự báo giá nhà trở thành một nhu cầu cấp thiết, đặc biệt là trong lĩnh vực toán ứng dụng và khoa học dữ liệu.

Phương pháp hồi quy, một công cụ mạnh mẽ trong thống kê và học máy, đã được chứng minh là có khả năng đáng kể trong việc dự báo và phân tích các vấn đề liên quan đến dữ liệu lớn, bao gồm cả việc dự báo giá nhà. Khả năng của hồi quy không chỉ giới hạn ở việc xác định mối quan hệ giữa các biến số mà còn giúp phát hiện các yếu tố ảnh hưởng đến giá nhà, từ đó cung cấp cái nhìn sâu sắc và cơ sở khoa học cho các quyết định đầu tư và mua bán bất động sản.

Đề tài "Phân Tích Và Dự Báo Giá Nhà Bằng Phương Pháp Hồi Quy" không chỉ nhằm mục đích áp dụng các lý thuyết hồi quy đã được biết đến mà còn khám phá sự phong phú và đa dạng của chúng trong bối cảnh thực tiễn, cụ thể là thông qua tập dữ liệu bất động sản Ames, Iowa. Tập dữ liệu này, với độ phức tạp và chi tiết cao, cung cấp một cơ hội tuyệt vời để áp dụng và so sánh các mô hình hồi quy khác nhau, từ hồi quy tuyến tính cơ bản đến các mô hình nâng cao như Ridge, Lasso, Elastic.

Với sự hướng dẫn của PGS.TS TRẦN VĂN LONG và sự cố gắng của bản thân, em hy vọng đề tài này không chỉ giúp em củng cố và mở rộng kiến thức về hồi quy mà còn góp phần vào việc tìm ra phương pháp dự báo giá nhà hiệu quả, phục vụ cho nhu cầu thực tiễn của xã hội. Đồng thời, thông qua việc nghiên cứu và phát triển mô hình, em mong muốn khám phá tiềm năng ứng dụng của toán ứng dụng và khoa học dữ liệu trong việc giải quyết các vấn đề thực tiễn, đặc biệt là trong lĩnh vực bất động sản.

PHẦN I: ĐẶT VẤN ĐỀ

1. Lý do chọn đề tài

Trong những năm gần đây, thị trường bất động sản Việt Nam chứng kiến nhiều biến động mạnh mẽ. Sự phát triển nhanh chóng của kinh tế, đô thị hóa và dòng vốn đầu tư dẫn đến sự thay đổi trong giá bất động sản. Cùng với đó là những yếu tố không ổn định như chính sách pháp lý thay đổi, sự biến động của thị trường tài chính,… làm cho việc dự đoán giá nhà trở nên khó khăn và phức tạp hơn bao giờ hết.

Trong bối cảnh đó, việc xây dựng một mô hình dự báo giá nhà chính xác và hiệu quả trở nên cực kỳ cần thiết. Mô hình này không chỉ giúp các nhà đầu tư, doanh nghiệp bất động sản có cái nhìn sâu sắc và khoa học về thị trường mà còn hỗ trợ người mua nhà đưa ra quyết định hợp lý, tránh rủi ro tài chính. Đồng thời, cung cấp dữ liệu và dự báo cho các cơ quan quản lý nhà nước, hỗ trợ trong việc xây dựng và điều chỉnh các chính sách phát triển nhà ở và quản lý thị trường bất động sản.

Một trong những thách thức lớn khi nghiên cứu thị trường bất động sản Việt Nam là sự thiếu hụt dữ liệu đầy đủ và chính xác. Việc thu thập và xử lý dữ liệu tại Việt Nam gặp nhiều khó khăn do chưa có hệ thống dữ liệu thống nhất và công khai. Vì vậy, đồ án sử dụng dữ liệu từ tập dữ liệu Ames, Iowa. Mục tiêu của đồ án là phát triển một mô hình dự đoán giá bán của một ngôi nhà nhất định tại Ames, Iowa. Thông qua phân tích hồi quy toàn diện, đồ án sẽ xây dựng mô hình dự đoán mạnh mẽ, chính xác và nắm bắt các mô hình cơ bản trong dữ liệu và cho phép đưa ra những dự đoán đáng tin cậy.

2. Mục tiêu nghiên cứu

- Nghiên cứu và phân tích các phương pháp hồi quy (tuyến tính, Ridge, Lasso,Elasstic) trong dự báo giá nhà.
- Đánh giá và so sánh hiệu quả của các mô hình hồi quy trên tập dữ liệu thực tế.
- Xây dựng mô hình dự báo giá nhà chính xác nhất dựa trên dữ liệu thu thập được.

3. Phương pháp nghiên cứu

Đề tài sẽ sử dụng phương pháp định lượng, áp dụng các mô hình hồi quy khác nhau để phân tích và dự báo giá nhà. Dữ liệu được thu thập sẽ bao gồm các thông số của nhà như diện tích, vị trí, số lượng phòng, tuổi của nhà, và các yếu tố khác ảnh hưởng đến giá nhà. Công cụ hỗ trợ bao gồm phần mềm R hoặc Python và các thư viện chuyên biệt cho phân tích dữ liệu và hồi quy.

4. Đối tượng nghiên cứu

Đề tài tập trung vào việc áp dụng mô hình hồi quy tuyến tính để dự báo giá nhà dựa trên tập dữ liệu Ames, Iowa. Tập dữ liệu này bao gồm các thông tin chi tiết và đa dạng về các ngôi nhà ở Ames, bao gồm cả các đặc điểm vật lý và khu vực, cung cấp một cơ sở dữ liệu phong phú để phân tích và xây dựng mô hình dự báo.

5. Phạm vi nghiên cứu

Phân tích được thực hiện trên tập dữ liệu Ames, Iowa, với mục tiêu xác định các yếu tố ảnh hưởng đến giá nhà và mức độ ảnh hưởng của chúng.

Đề tài sẽ tập trung vào việc phát triển mô hình hồi quy tuyến tính và các biến thể như Ridge và Lasso để dự báo giá nhà một cách chính xác.

Thời gian nghiên cứu và phân tích được giới hạn từ ngày bắt đầu đến ngày nộp đồ án tốt nghiệp.

6. Kết cấu đề tài

Kết cấu đề tài gồm 3 phần:

- Phần I: Đặt vấn đề
- Phần II: Nội dung và kết quả nghiên cứu của đề tài, gồm có:

  Chương I. Hồi quy tuyến tính

  1.1 Hồi quy tuyến tính đa biến

  1.2 Một số mô hình hồi quy có hiệu chỉnh

PHẦN II: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

CHƯƠNG I. HỒI QUY TUYẾN TÍNH

1.1 Hồi quy tuyến tính đa biến

Trong phần này nghiên cứu về hàm hồi quy đa biến

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p + \varepsilon \tag{1.1}$$

Giả sử ta có p biến độc lập $X_1, X_2, \dots, x_p$ và các biến phụ thuộc y với n giá trị trong tập huấn luyện. Ta ký hiệu:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}, \quad w = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_p \end{pmatrix} \tag{1.2}$$

Để thuận tiện, ta ký hiệu $x_{i,0} = 1$. Các tham số w xác định bằng phương pháp bình phương tối thiểu, nghĩa là tìm w để sai số trung bình bình phương đạt giá trị nhỏ nhất.

$$E = \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=0}^{p} x_{ij} w_j)^2 \rightarrow min \tag{1.3}$$

Đạo hàm của E đối với w tính bởi công thức

$$\frac{\partial E}{\partial w} = -\frac{2}{n} X^T (y - Xw) = -\frac{2}{n} \begin{pmatrix} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{p} x_{ij} w_j \right) \\ \sum_{i=1}^{n} x_{i1} \left(y_i - \sum_{j=0}^{p} x_{ij} w_j \right) \\ \vdots \\ \sum_{i=1}^{n} x_{ip} \left(y_i - \sum_{j=0}^{p} x_{ij} w_j \right) \end{pmatrix} = 0 \tag{1.4}$$

Hệ phương trình trên được viết dưới dạng phương trình chính tắc

$$X^T X w = X^T y \tag{1.5}$$

Khi đó nếu ma trận $X^T X$ khả nghịch thì hệ số w tính bởi công thức

$$w = (X^T X)^{-1} X^T y \quad (1.6)$$

1.2 Một số mô hình hồi quy có hiệu chỉnh

Trong phần này chúng ta xem xét đối với trường hợp ma trận $X^T X$ không khả nghịch, nghĩa là các biến $X_1, X_2, \dots, X_p$ có mối quan hệ tuyến tính với nhau. Khi đó ta xét hai mô hình hồi quy hiệu chỉnh là hàm hồi quy Ridge và hồi quy Lasso (least absolute shrinkage and selection operator

1.2.1. Hồi quy Ridge

Trong hàm hồi quy Ridge ta thêm hạng số hiệu chỉnh

$$\lambda\|x\|^2 = \lambda\left(w_0^2 + w_1^2 + \cdots + w_p^2\right) \quad (1.7)$$

Với hệ số $\lambda > 0$ vào hàm sai số E nghĩa là

$$E = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{p} x_{ij} w_j\right)^2 + \lambda\|x\|^2 \to \min \quad (1.8)$$

Khi đó phương trình chính tắc trở thành

$$(X^T X + n\lambda I)w = X^T y.$$

Với ma trận $X^T X + n\lambda I$ luôn khả nghịch và tham số w xác định bởi công thức

$$w = (X^T X + n\lambda I)^{-1} X^T y. \quad (1.9)$$

1.2.2. Hồi quy Lasso

Trong hàm hồi quy Lasso ta thêm số hạng hiệu chỉnh

$$\lambda\|w\|_1 = \lambda\left(|w_0| + |w_1| + \cdots + |w_p|\right)$$

Với hệ số $\lambda > 0$ vào hàm sai số E, nghĩa là

$$E = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{j=0}^{p} x_{ij} w_j)^2 + \lambda\|w\|_1 \to min \quad (1.10)$$

Để tối ưu hàm trên ta cần sử dụng phương pháp tối ưu theo tọa độ. Để thuận tiện ta viết hàm mục tiêu dưới dạng

$$E = \frac{1}{2n} \|y - Xw\|^2 + \lambda\|w\|_1 = \frac{1}{2n} \left\| y - \sum_{j=0}^{p} X_j w_j \right\|^2 + \lambda\|w\|_1 \qquad (1.11)$$

Trong đó $X_0, X_1, \dots, X_p$ là các cột của ma trận X.

Ta xét trường hợp đơn giản $X^T X = I$. Khai triển hàm mục tiêu E dưới hàm bậc hai dạng

$$E = \frac{1}{2n} y^T y + \frac{1}{2n} w^T w - \frac{1}{n} \sum_{j=0}^{p} (y^T X_j) w_j + \lambda \left(\sum_{j=0}^{p} |w_j| \right) \qquad (1.12)$$

Lấy đạo hàm riêng của E theo biến w_j, chú ý hàm $|w_j|$ không khả vi tại $w_j = 0$ nên ta có 3 trường hợp sau

- Nếu $w_j > 0$,

$$\frac{\partial E}{\partial w_j} = \frac{1}{n} w_j - \frac{1}{n} y^T X_j + \lambda = 0 => w_j^{lasso} = y^T X_j - n\lambda$$

- Nếu $w_j < 0$,

$$\frac{\partial E}{\partial w_j} = \frac{1}{n} w_j - \frac{1}{n} y^T X_j - \lambda = 0 => w_j^{lasso} = y^T X_j + n\lambda$$

- Nếu $w_j = 0$, khi đó ta đặt

$$w_j^{lasso} = 0$$

Vậy tối ưu hóa theo tọa độ của hệ số w_j được tính theo công thức

$$w_j^{lasso} = \begin{cases} y^T X_j - n\lambda & nếu\ y^T X_j > n\lambda; \\ y^T X_j + n\lambda & nếu\ y^T X_j < -n\lambda; \\ 0\ nếu - n\lambda \le y^T X_j \le n\lambda; \end{cases} \qquad (1.13)$$

Ta đặt hàm ngưỡng mềm

$$S(x,\lambda) = \begin{cases} x - \lambda \, nếu \, x > \lambda; \\ x + \lambda \, nếu \, x < \lambda; \\ 0 \, nếu - \lambda \leq x \leq \lambda; \end{cases}$$

Khi đó

$$w_j^{Lasso} = S(y^T X_j, n\lambda)$$

Các hệ số của hàm hồi quy sẽ quy về 0 nếu hệ số đủ nhỏ. Do đó hàm hồi quy lasso còn được gọi là hàm hồi quy thưa.

Bây giờ ta xét trường hợp tổng quát, hàm mục tiêu được viết lại dưới dạng

$$E = \frac{1}{2n} y^T y + \frac{1}{2n} w^T (X^T X) w - \frac{1}{n} \sum_{j=0}^{p} (y^T X_j) w_j + \lambda \left(\sum_{j=0}^{p} |w_j| \right) \qquad (1.14)$$

Lấy đạo hàm đối với biến w_j ta được

$$\frac{\partial E}{\partial w_j} = \frac{1}{n} X^T X w - \frac{1}{n} y^T X_j + \lambda sgn(w_j)$$

$$= \frac{1}{n} \left(X_j^T X_j w_j + X_{0:\, j-1}^T X_{0:\, j-1} + X_{j+1:\, n}^T X_{j+1:n} w_{j+1:n} - y^T X_j \right) + \lambda sgn(w_j)$$

Giải phương trình $\frac{\partial E}{\partial w_j} = 0$ ta được

$$w_j^{lasso} = \frac{1}{X_j^T X_j} S\left(y^T X_j - X_{0:\, j-1}^T X_{0:j-1} w_{0:j-1} - X_{j+1:n}^T X_{j+1:n} w_{j+1:n}, n\lambda \right) \qquad (1.15)$$

Thuật toán lặp tìm các các hệ số hồi quy Lasso như sau :

Thuật toán 1.1 Tối ưu hoá hệ số hồi quy lasso.

Data: $(x_i, y_i), i = 1,2,\ldots,n.$
Result: $w_0, w_1, \ldots, w_p$
Khởi tạo $w^{(0)}$;
repeat
  for $j \leftarrow 0$ **to** p **do**
    $w_j^{(k+1)} = \frac{1}{X_j^T X_j} S\left(y^T X_j - X_{0:j-1}^T X_{0:j-1} w_{0:j-1}^{(k+1)} - X_{j+1:n}^T X_{j+1:n} w_{j+1:n}^{(k)}, n\lambda\right)$
  end
until $||w^{(k+1)} - w^{(k)}|| < \varepsilon$;

1.2.3. Hồi quy Elasstic

Hồi quy Elastic Net là một phương pháp hồi quy tuyến tính điều chuẩn kết hợp cả hai hạng số điều chuẩn từ hồi quy Ridge và hồi quy Lasso. Phương trình của hồi quy Elastic Net có dạng:

$$E = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{j=0}^{p} x_{ij} w_j)^2 + \alpha\lambda \sum_{j=0}^{p}|w_j| + (1-\alpha)\lambda \sum_{j=0}^{p} w_j^2 \rightarrow min \quad (1.16)$$

Với α là tham số cân bằng giữa Lasso và Ridge ($0 \leq \alpha \leq 1$)

Khi đó phương trình chính tắc:

$$(X^T X + n(1-\alpha)\lambda I)w + \alpha\lambda s = X^T y \quad (1.17)$$

Trong đó I là ma trận đơn vị và s là vector đặc biệt mà mỗi phần tử của s_j phụ thuộc vào giá trị của w_j

$$s_j = \begin{cases} 1 \text{ nếu } w_j > 0 \\ 0 \text{ nếu } w_j = 0 \\ -1 \text{ nếu } w_j < 0 \end{cases}$$

Chương II. Phân tích tập dữ liệu Ames Iowa và dự đoán

2.1 Phân tích dữ liệu

Bộ dữ liệu bao gồm hơn 2900 quan sát và 81 thuộc tính khác nhau liên quan đến việc đánh giá giá trị ngôi nhà. Trong đó có tất cả 43 thuộc tính phân loại và 38 thuộc tính định lượng.

Thuộc tính định tính	Thuộc tính định lượng
• Nominal: thuộc tính dữ liệu danh nghĩa MSZoning (Loại khu vực) Street, Alley (Loại đường truy cập) LotShape, LandContour, LotConfig (Hình dạng lô, định hình đất, cấu hình lô) Neighborhood (Khu vực) Condition1 (Tình trạng gần các điều kiện nhất định) RoofStyle, RoofMatl (Kiểu mái, vật liệu mái) Exterior1st, Exterior2nd (Vật liệu ngoài của ngôi nhà) Foundation (Loại móng) MasVnrType (Loại vật liệu trang trí mặt ngoài) Heating (Loại hệ thống sưởi) MiscFeature (Tính năng đặc biệt không được liệt kê ở các danh mục khác)	• Discrete: thuộc tính dữ liệu rời rạc YearBuilt, YearRemodAdd (Năm xây dựng và năm cải tạo) BsmtFullBath, BsmtHalfBath, FullBath, HalfBath (Số lượng phòng tắm) Bedroom, Kitchen (Số phòng ngủ và bếp) TotRmsAbvGrd (Tổng số phòng) Fireplaces (Số lò sưởi) GarageCars (Sức chứa xe trong garage) MoSold, YrSold (Tháng và năm bán)

SaleType (Loại hình bán hàng) SaleCondition (Điều kiện bán hàng)	
• Ordinal: thuộc tính dữ liệu thứ tự Utilities (Loại tiện ích sẵn có) LandSlope (Độ dốc của lô đất) ExterQual, ExterCond, BsmtQual... (Chất lượng và điều kiện ngoài, tầng hầm) HeatingQC (Chất lượng hệ thống sưởi) KitchenQual (Chất lượng nhà bếp) GarageQual, PoolQC, Fence (Chất lượng garage, hồ bơi, hàng rào)	• Continuous: thuộc tính dữ liệu liên tục LotFrontage, LotArea (Chiều rộng và diện tích lô đất) MasVnrArea (Diện tích vật liệu trang trí mặt ngoài) BsmtFinSF1, BsmtFinSF2, BsmtUnfSF (Diện tích tầng hầm hoàn thiện và không hoàn thiện) TotalBsmtSF (Tổng diện tích tầng hầm) 1stFlr, 2ndFlrSF (Diện tích tầng 1 và 2) GrLivArea (Tổng diện tích sử dụng sống) GarageArea (Diện tích garage) WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch ScreenPorch, PoolArea (Diện tích sàn gỗ, hiên mở, hiên kín, hiên ba mùa, hiên chắn, diện tích hồ bơi)

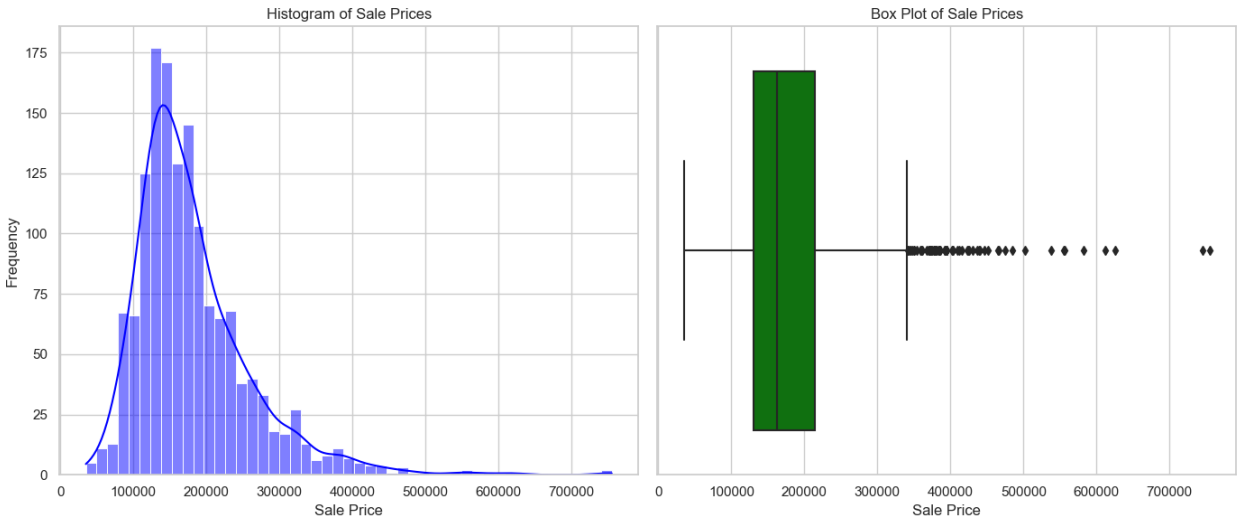
Bảng mô tả các thuộc tính

2.1.1 Giá bán

Xét trong tập huấn luyện, dữ liệu bao gồm 1460 mục với 81 cột khác nhau. Biến

SalePrice, giá bán của ngôi nhà, có các thống kê cơ bản như sau:

- Số lượng mục: 1460
- Giá trị trung bình: $180,921
- Độ lệch chuẩn: $79,442
- Giá trị nhỏ nhất: $34,900
- Phân vị 25%: $129,975
- Giá trị trung vị (50%): $163,000
- Phân vị 75%: $214,000
- Giá trị cao nhất: $755,000



Hình 1.1 Biểu đồ phân bố giá bán

Nhận xét:

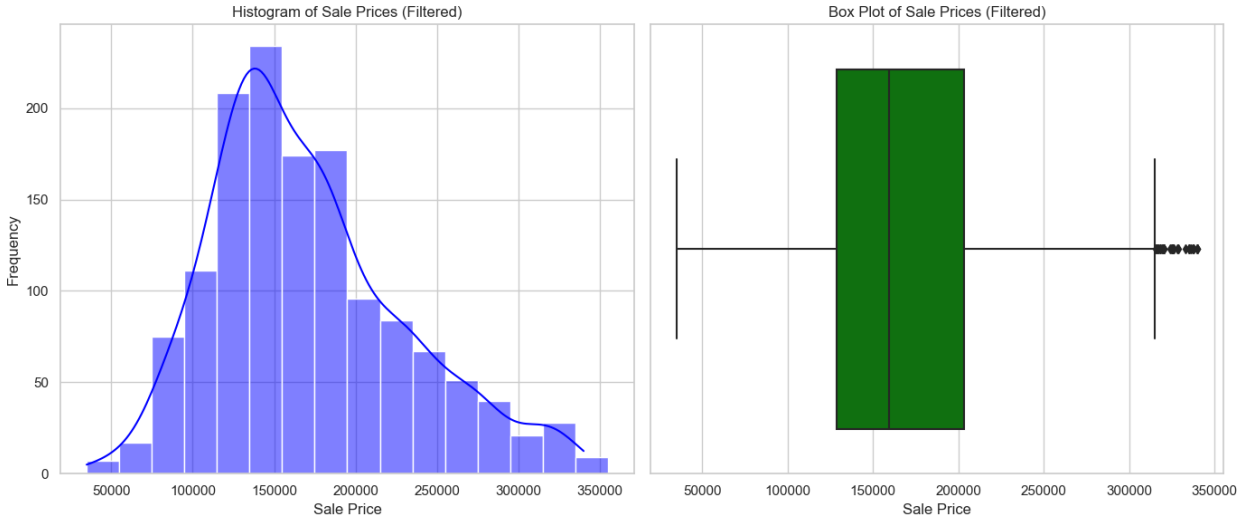
Histogram (Biểu Đồ Tần Suất)

- Hình dạng và phân bố: Histogram cho thấy phân bố của giá bán có hình dạng xấp xỉ phân phối chuẩn nhưng lệch phải (positive skew), với một đuôi dài kéo dài về phía giá cao. Điều này cho thấy rằng trong khi hầu hết các ngôi nhà có giá bán ở mức vừa phải, một số ít các ngôi nhà có giá bán rất cao so với phần lớn dữ liệu.

- Độ lệch (Skewness): Độ lệch về phía giá cao hơn cho thấy sự tập trung của các ngôi nhà giá rẻ hơn là nhiều, trong khi có một số ít ngôi nhà có giá đắt đỏ nổi bật hơn. Sự lệch này cần được xem xét khi phân tích bởi vì nó có thể ảnh hưởng đến giá trị trung bình, làm cho giá trị này cao hơn so với giá trị trung vị.

Box Plot (Biểu Đồ Hộp)

- Outliers (Các điểm ngoại lệ): Các điểm ngoại lệ được thể hiện bằng các điểm nằm xa hộp chính của biểu đồ. Điều này chỉ ra rằng có những ngôi nhà với giá bán đặc biệt cao so với phần còn lại.
- Median và IQR: Đường kẻ ngang trong hộp thể hiện giá trị trung vị của dữ liệu, và kích thước của hộp (IQR - interquartile range) cho thấy phạm vi giá bán của đa số các ngôi nhà. Giá trị trung vị không chạm vào giữa của hộp, lại một lần nữa xác nhận phân bố lệch phải.
- Phạm vi giá: Box plot cũng cho thấy phạm vi của phần lớn dữ liệu, trừ các outliers. Các giá trị trong khoảng từ Q1 đến Q3 (từ cạnh dưới đến cạnh trên của hộp) đại diện cho giá trị của 50% số ngôi nhà được bán.



Hình 1. 2 Biểu đồ phân bố giá bán sau khi chuyển đổi

Nhận xét:

Histogram (Biểu Đồ Tần Suất Sau Khi Lọc)

- Phân bố: Phân bố của giá bán sau khi lọc trở nên đối xứng hơn so với biểu đồ ban đầu. Điều này cho thấy việc loại bỏ các ngoại lệ đã giúp làm giảm sự lệch phải của phân bố, làm cho nó gần với hình dạng chuẩn hơn.
- Sự phân bố giá trị: Có thể thấy rằng đa số giá bán tập trung nhiều hơn trong khoảng từ 100,000 đến 250,000, với đỉnh cao nhất xấp xỉ 150,000.

Box Plot (Biểu Đồ Hộp Sau Khi Lọc)

- Điểm ngoại lệ: Số lượng điểm ngoại lệ đã giảm đáng kể, cho thấy các điểm còn lại phù hợp hơn với phân phối chung của dữ liệu. Các điểm này ít hơn và nằm gần hơn với phần lớn dữ liệu.
- Median và IQR: Giá trị trung vị (đường kẻ giữa hộp) giờ đây nằm ở vị trí trung tâm của hộp, cho thấy rằng dữ liệu ít bị lệch hơn. Khoảng giá (IQR) cũng thu hẹp lại, phản ánh một mức độ biến động giá thấp hơn so với trước khi lọc ngoại lệ.

Tác động của Việc Loại Bỏ Ngoại Lệ

- Độ tin cậy: Việc loại bỏ các điểm ngoại lệ có thể làm tăng độ tin cậy của các phân tích thống kê và mô hình học máy bằng cách giảm ảnh hưởng của các giá trị cực đoan.
- Biểu diễn dữ liệu: Dữ liệu sau khi điều chỉnh cho thấy một biểu diễn chính xác hơn về giá bán thông thường, hỗ trợ tốt hơn cho việc ra quyết định và dự báo.

```python
import pandas as pd

Q1 = train_df['SalePrice'].quantile(0.25)
Q3 = train_df['SalePrice'].quantile(0.75)
IQR = Q3 - Q1


lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR


filtered_df = train_df[(train_df['SalePrice'] >= lower_bound) & (train_df['SalePrice'] <= upper_bound)]

print("Original data count:", len(train_df))
print("Filtered data count:", len(filtered_df))
```

✓ 0.0s

```
Original data count: 1460
Filtered data count: 1399
```

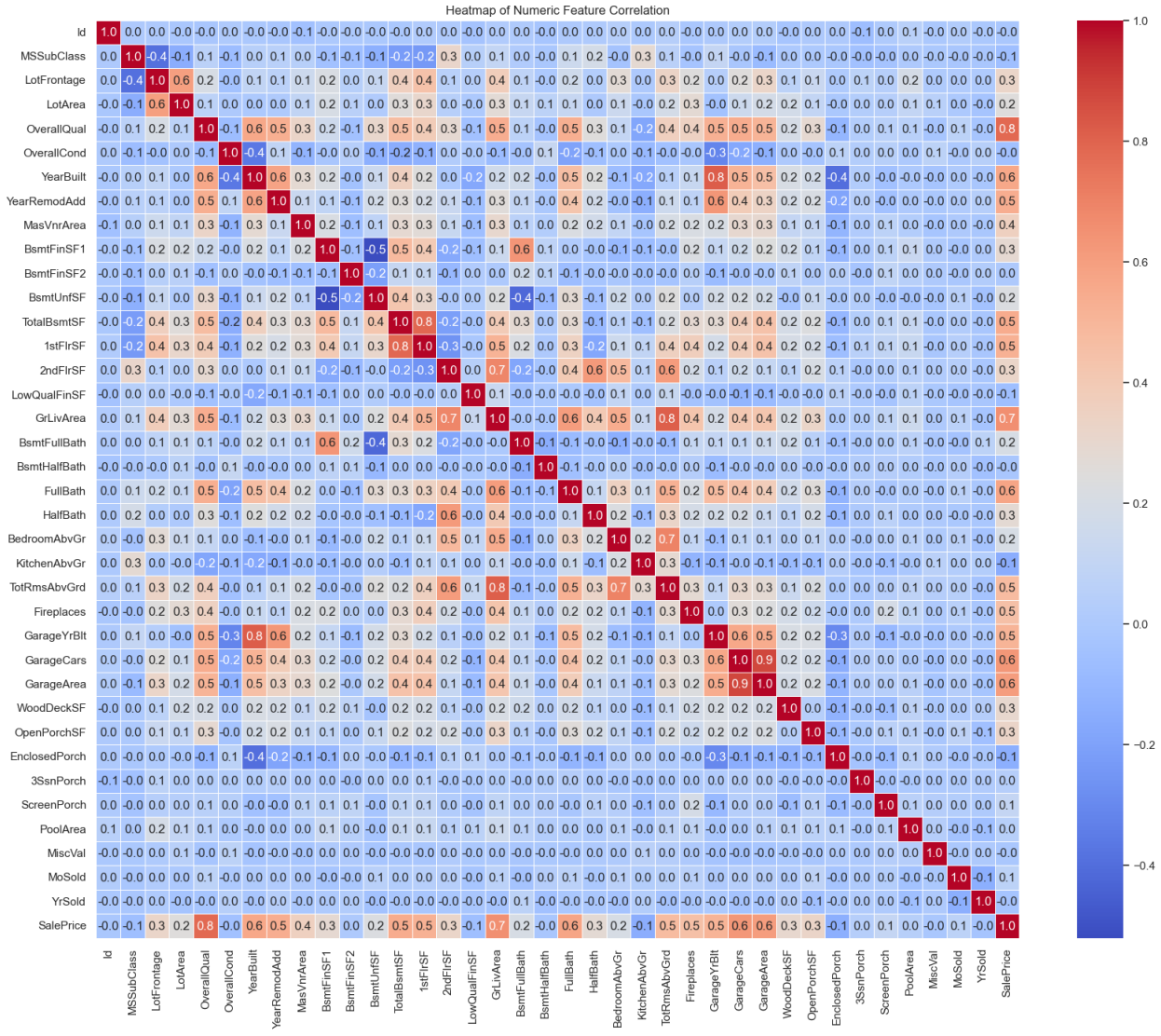
```python
df = pd.read_csv('filtered_data.csv')
df.shape
```

✓ 0.3s

```
(1399, 81)
```

2.1.2 Thuộc tính định lượng

Chúng ta sẽ tạo một biểu đồ nhiệt (heatmap) để trực quan hóa ma trận tương quan giữa các thuộc tính định lượng . Biểu đồ nhiệt sẽ giúp hiểu rõ mối liên hệ giữa tất cả các tính năng



Hình 1. 3 Biểu đồ nhiệt thể hiện mối quan hệ giữa giá bán và các thuộc tính định lượng

Phân tích tương quan tiết lộ cách các đặc điểm số của ngôi nhà tương quan với giá bán. Dưới đây là một số điểm nổi bật:

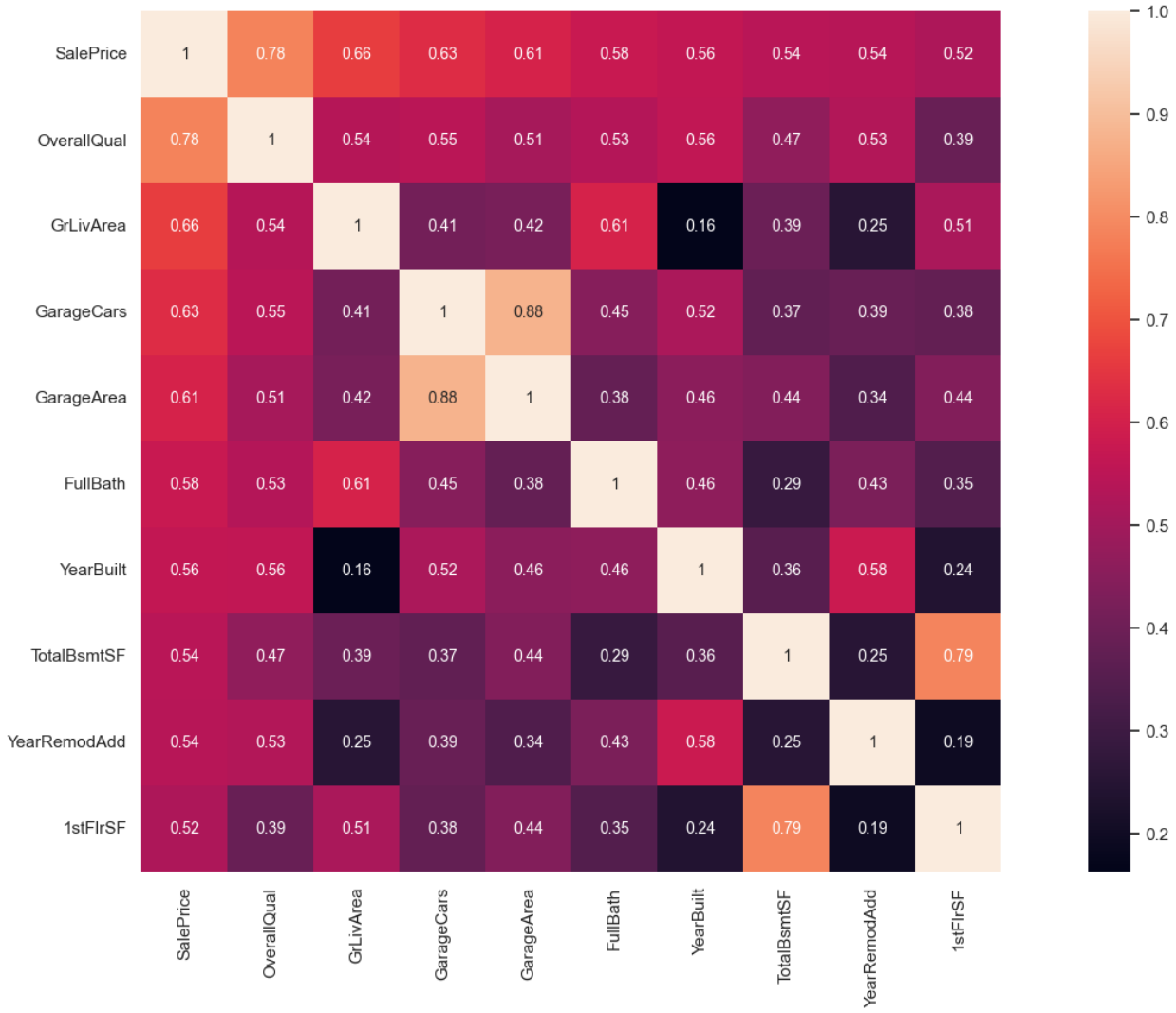
- OverallQual (Chất lượng tổng thể vật liệu và hoàn thiện) có mối tương quan tích cực mạnh nhất với giá bán (0.78), cho thấy những ngôi nhà chất lượng cao thường có giá bán cao hơn.

- GrLivArea (Diện tích sử dụng ở tầng trên mặt đất) cũng cho thấy mối tương quan tích cực mạnh (0.66), chỉ ra rằng những ngôi nhà có diện tích lớn hơn thường có giá cao hơn.
- GarageCars (Số lượng xe có thể chứa trong garage) và GarageArea (Diện tích garage bằng feet vuông) cũng có mối tương quan đáng kể với giá bán, điều này nhấn mạnh giá trị thêm của garage.
- TotalBsmtSF (Tổng diện tích tầng hầm bằng feet vuông) và 1stFlrSF (Diện tích tầng một bằng feet vuông) có liên kết chặt chẽ với giá bán, cho thấy tầm quan trọng của không gian nội thất.
- YearBuilt (Năm xây dựng ban đầu) và YearRemodAdd (Năm cải tạo) cho thấy mối tương quan vừa phải, cho thấy những ngôi nhà mới và những ngôi nhà được cập nhật gần đây có khả năng được bán với giá cao hơn.

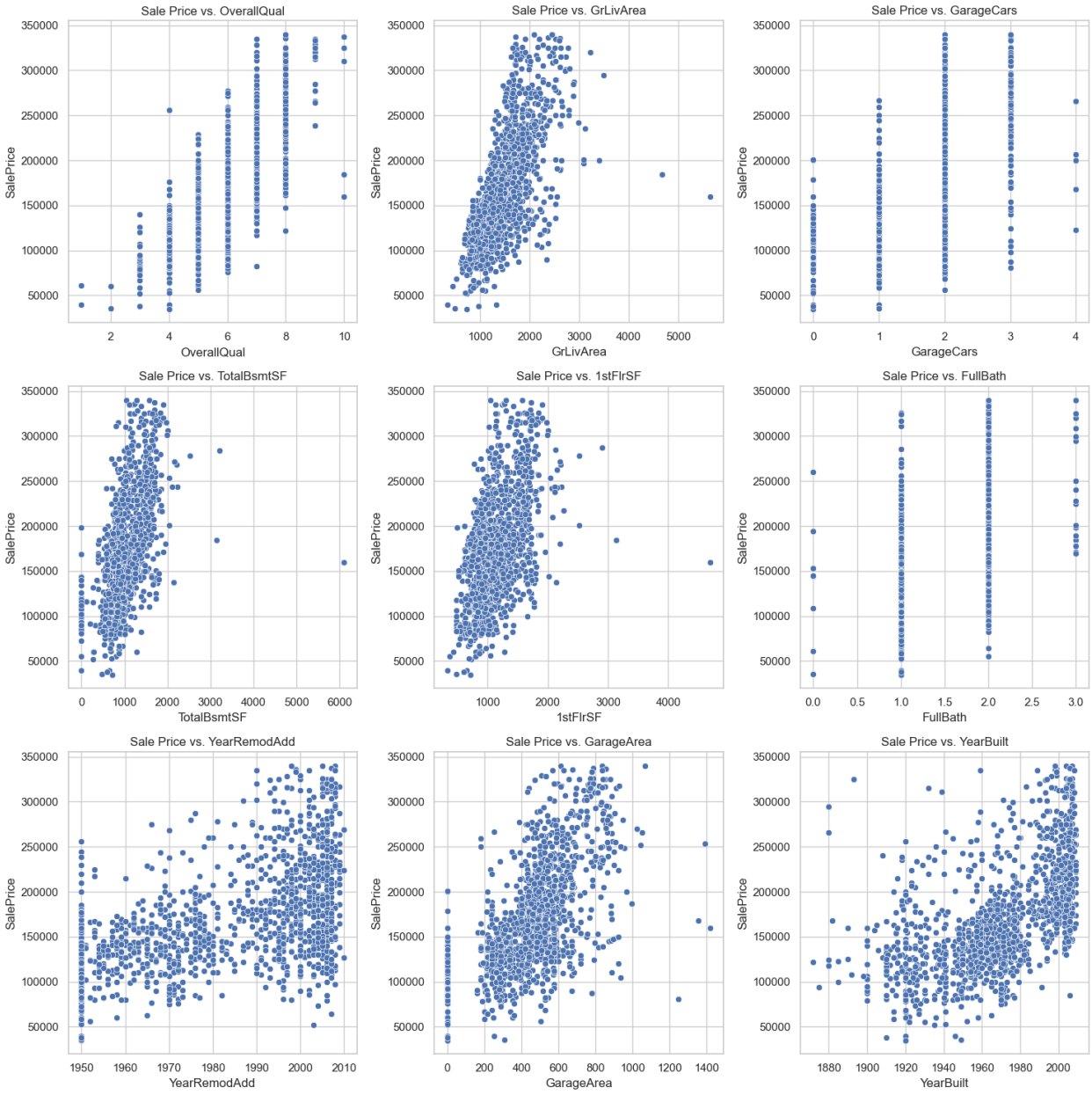
Những tương quan tiêu cực không nổi bật nhưng vẫn mang thông tin hữu ích:

- EnclosedPorch và KitchenAbvGr (Số lượng bếp ở trên mặt đất) có mối tương quan tiêu cực nhẹ với giá bán, chỉ ra những yếu tố có thể làm giảm giá trị.

Trong sơ đồ biểu đồ trên có khá nhiều dữ liệu có độ tương quan thấp nên chúng ta sẽ liệt kê các biến có độ tương quan cao nhất để dễ dàng hơn trong việc thực hiện trực quan hóa. Dưới đây là biểu đồ nhiệt thể hiện các giá trị tương quan cao nhất so với giá bán (SalePrice).



Hình 1. 4 Biểu đồ nhiệt thể hiện mối quan hệ giữa giá bán và các thuộc tính định lượng quan trọng nhất



Hình 1. 5 Biểu đồ phân tán giữa các thuộc tính định lượng so với giá bán

Các biểu đồ phân tán ở trên minh họa mối quan hệ giữa giá bán và một số đặc điểm có mối tương quan cao nhất:

- Chất lượng tổng thể so với Giá Bán: Khi điểm đánh giá chất lượng tăng, giá bán nói chung cũng tăng. Xu hướng này làm nổi bật mối quan hệ tích cực mạnh.

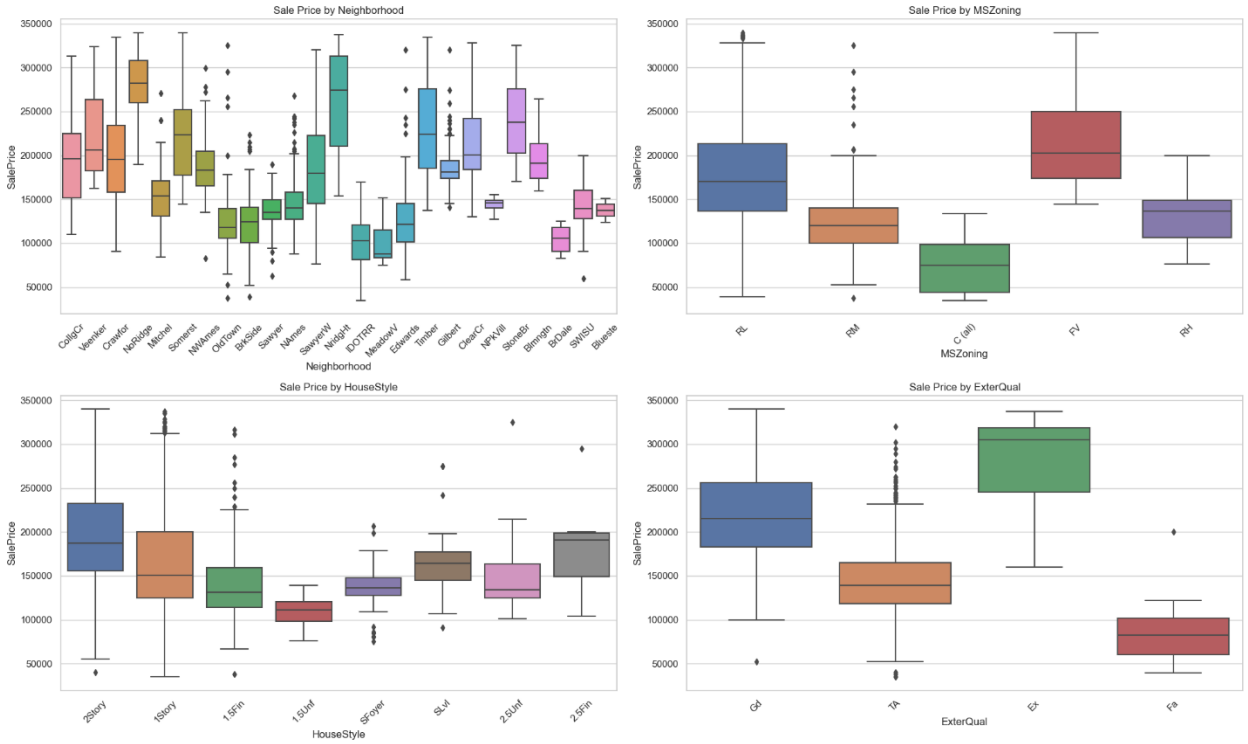
- Diện tích sử dụng trên mặt đất (GrLivArea) so với Giá Bán: Các khu vực sinh hoạt lớn hơn có liên quan đến giá bán cao hơn, cho thấy một xu hướng tuyến tính rõ ràng.
- Số chỗ đỗ xe Garage Cars so với Giá Bán: Nhiều chỗ đỗ xe garage có xu hướng tương quan với giá bán cao hơn, mặc dù sự tăng giá có vẻ như bị bão hòa ở một điểm nhất định.
- Tổng diện tích tầng hầm (TotalBsmtSF) so với Giá Bán: Tương tự như diện tích sinh hoạt, những tầng hầm lớn hơn có xu hướng dẫn đến giá cao hơn.
- Những biểu đồ này xác nhận các mối tương quan mà chúng ta đã quan sát và giúp hiểu cách những đặc điểm này có thể ảnh hưởng đến giá nhà.

2.1.3 Thuộc tính phân loại

Phân tích tác động của dữ liệu phân loại đến giá bán bao gồm việc hiểu cách các danh mục khác nhau trong dữ liệu có thể ảnh hưởng đến giá cả của các ngôi nhà. Chúng ta sẽ sử dụng biểu đồ hộp để kiểm tra trực quan sự biến đổi của giá bán qua các danh mục khác nhau của một số đặc điểm được chọn. Cách tiếp cận này sẽ giúp chúng ta xác định liệu có bất kỳ sự khác biệt đáng kể nào về giá bán dựa trên các thuộc tính phân loại hay không. Bắt đầu bằng cách chọn một số đặc điểm phân loại có thể ảnh hưởng đáng kể đến giá bán. Một số đặc điểm phân loại có ảnh hưởng thường thấy có thể bao gồm:

- Neighborhood (Khu vực): Chỉ ra vị trí địa lý trong giới hạn thành phố Ames.
- MSZoning (Phân loại quy hoạch sử dụng đất): Xác định phân loại quy hoạch chung của bất động sản.
- HouseStyle (Kiểu nhà): Kiểu dáng của ngôi nhà.
- ExterQual (Chất lượng ngoại thất): Đánh giá chất lượng của vật liệu bên ngoài.

Chúng ta có thể tạo biểu đồ hộp cho các đặc điểm này để xem giá bán thay đổi như thế nào giữa các danh mục khác nhau trong từng đặc điểm.



Hình 1. 6 Biểu đồ hộp thể hiện mối quan hệ giữa giá bán và các thuộc tính phân loại

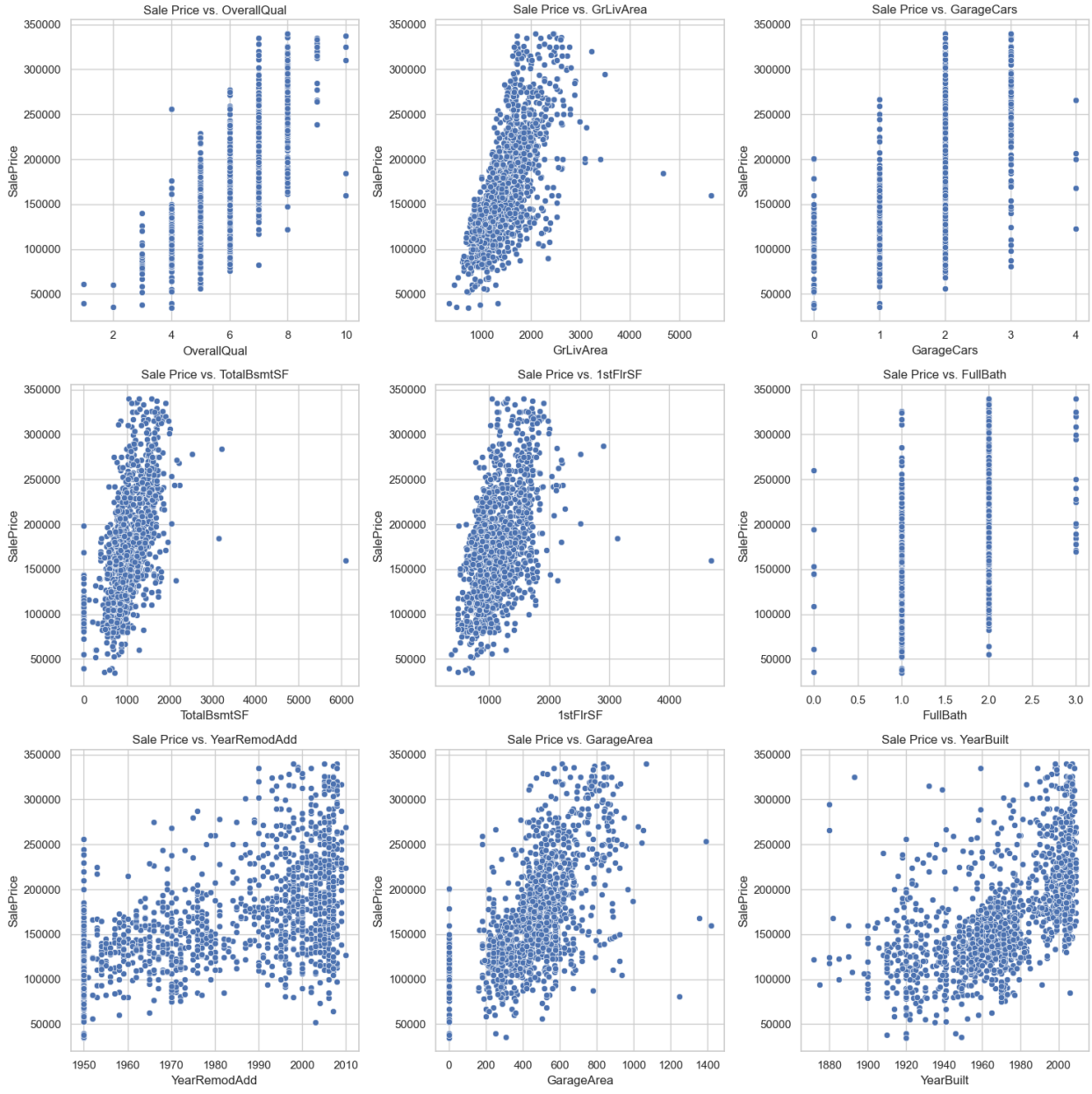
Nhận xét :

- Neighborhood (Khu vực): Có sự thay đổi đáng kể về giá bán giữa các khu vực khác nhau. Một số khu vực có giá trung bình cao hơn và phạm vi giá rộng hơn, cho thấy đây là những khu vực cao cấp hơn.
- MSZoning (Phân loại quy hoạch sử dụng đất đô thị): Các phân loại quy hoạch cũng cho thấy các mức giá khác nhau. Ví dụ, các khu vực dân cư mật độ thấp (RL) thường có giá bán cao hơn so với khu thương mại (C) hoặc khu dân cư mật độ cao (RM).
- HouseStyle (Kiểu nhà): Các kiểu nhà khác nhau có mức giá khác nhau, với một số kiểu như nhà hai tầng thường có giá trung bình cao hơn.
- ExterQual (Chất lượng ngoại thất): Chất lượng ngoại thất có ảnh hưởng rõ ràng đến giá bán. Những ngôi nhà có chất lượng ngoại thất xuất sắc (Ex) có giá cao hơn nhiều so với những ngôi nhà có đánh giá chất lượng thấp hơn.

Những trực quan này giúp làm nổi bật cách các biến phân loại khác nhau có thể ảnh hưởng đến giá nhà, điều này rất hữu ích để hiểu thị trường và xây dựng các mô hình dự báo.

2.2 Tiền xử lý dữ liệu

2.2.1 Dữ liệu ngoại lai



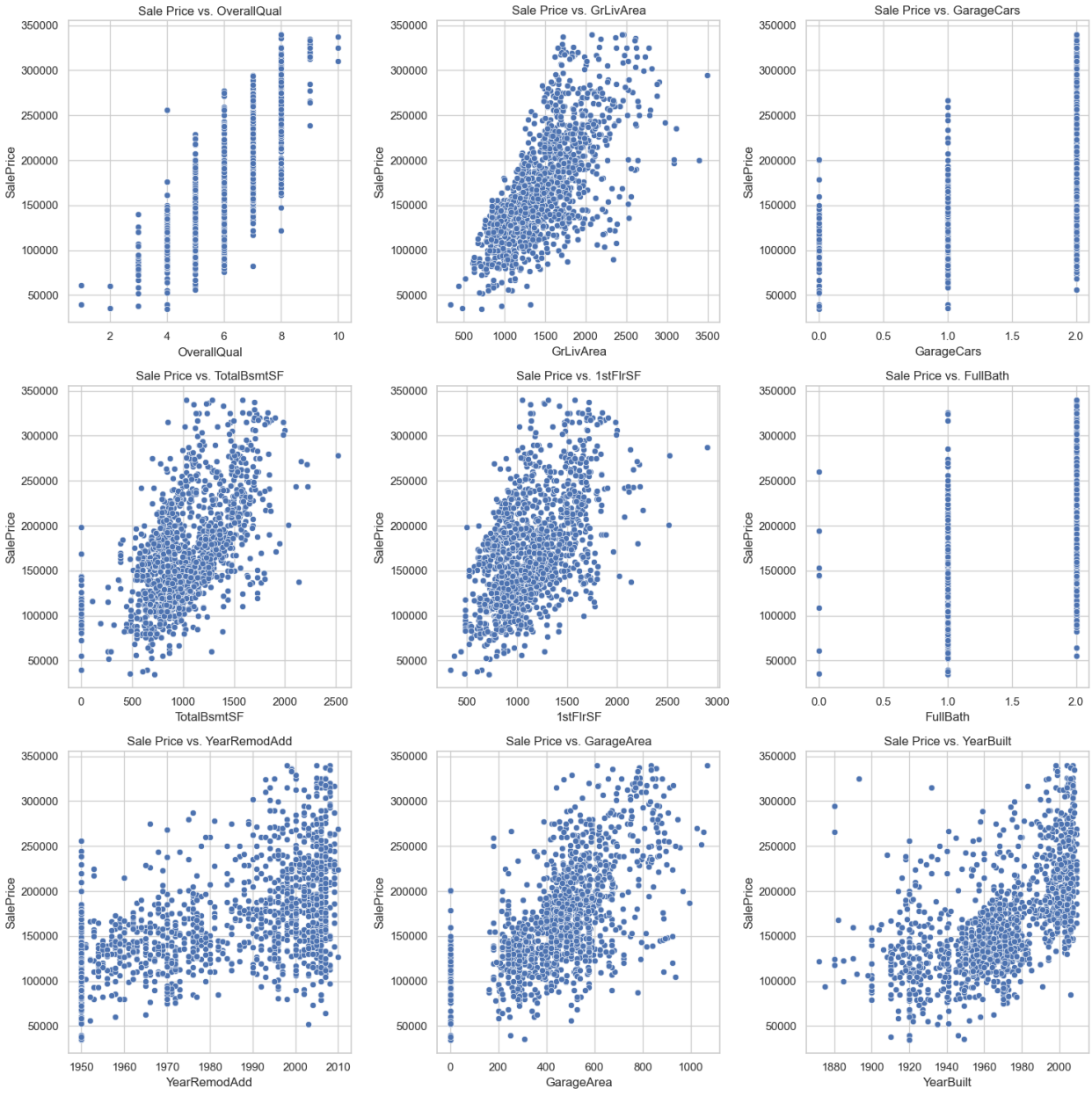
Hình 1. 7 Biểu đồ các thuộc tính định lượng quan trọng so với SalePrice

Các Bước Xử Lý Dữ Liệu

1. Loại bỏ các ngôi nhà có OverallQual từ 4 đến 7 nhưng có SalePrice cao bất thường (trên 300,000): Các ngôi nhà có chất lượng tổng thể (OverallQual) từ 4 đến 7 và giá bán cao hơn 300,000 được xác định là ngoại lệ. Những ngôi nhà này không tương xứng về chất lượng và giá bán, có thể gây sai lệch trong phân tích dữ liệu.
2. Loại bỏ các ngôi nhà có diện tích GrLivArea lớn hơn 4000 nhưng SalePrice dưới 300,000: Các ngôi nhà có diện tích sử dụng trên mặt đất (GrLivArea) lớn hơn 4000 và giá bán thấp hơn 300,000 được coi là ngoại lệ. Những ngôi nhà này có diện tích lớn nhưng giá bán không tương xứng.
3. Gộp các giá trị 4,3 trong cột GarageCars thành 2: Tất cả các giá trị 4, 3 trong cột số chỗ đậu xe trong garage (GarageCars) sẽ được thay thế bằng giá trị 2. Điều này giúp giảm sự phức tạp của biến và gộp các giá trị tương tự vào một nhóm.
4. Loại bỏ các ngôi nhà có diện tích TotalBsmtSF lớn hơn 3000: Các ngôi nhà có tổng diện tích tầng hầm (TotalBsmtSF) lớn hơn 3000 được coi là ngoại lệ vì diện tích tầng hầm quá lớn có thể không đại diện cho phần lớn các ngôi nhà khác.
5. Loại bỏ các ngôi nhà có diện tích 1stFlrSF lớn hơn 3000: Các ngôi nhà có diện tích tầng một (1stFlrSF) lớn hơn 3000 được coi là ngoại lệ vì diện tích tầng một quá lớn có thể không đại diện cho phần lớn các ngôi nhà khác.
6. Gộp các giá trị 3 trong cột FullBath thành 2: Tất cả các giá trị 3 trong cột số phòng tắm đầy đủ (FullBath) sẽ được thay thế bằng giá trị 2. Điều này giúp giảm sự phức tạp của biến và gộp các giá trị tương tự vào một nhóm.
7. Loại bỏ các ngôi nhà có diện tích GarageArea lớn hơn 1200: Các ngôi nhà có diện tích garage (GarageArea) lớn hơn 1200 được coi là ngoại lệ vì diện tích garage quá lớn có thể không đại diện cho phần lớn các ngôi nhà khác.

```
outliers = df[(df['OverallQual'].between(4, 7)) & (df['SalePrice'] > 300000)].index
df = df.drop(outliers)
outliers = df[(df['GrLivArea'] > 4000) & (df['SalePrice'] < 300000)].index
df = df.drop(outliers)
df.loc[df['GarageCars']==4,'GarageCars'] = 3
df.loc[df['GarageCars']==3,'GarageCars'] = 2
outliers = df[(df['TotalBsmtSF'] > 3000) ].index
df = df.drop(outliers)
outliers = df[(df['1stFlrSF'] > 3000) ].index
df = df.drop(outliers)
df.loc[df['FullBath']==3,'FullBath'] = 2
outliers = df[(df['GarageArea'] > 1200) ].index
df = df.drop(outliers)
```

Các thuộc tính quan trọng sau khi xử lý ngoại lai



Hình 1. 8 Biểu đồ so sánh giữa các thuộc tính định lượng quan trọng so với SalePrice sau khi xử lý ngoại lai

Thông tin cúa tập dữ liệu sau khi xử lý ngoại lai như sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2842 entries, 0 to 2841
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             2842 non-null   int64
 1   MSSubClass     2842 non-null   int64
 2   MSZoning       2838 non-null   object
 3   LotFrontage    2362 non-null   float64
 4   LotArea        2842 non-null   int64
 5   Street         2842 non-null   object
 6   Alley          198 non-null    object
 7   LotShape       2842 non-null   object
 8   LandContour    2842 non-null   object
 9   Utilities      2840 non-null   object
 10  LotConfig      2842 non-null   object
 11  LandSlope      2842 non-null   object
 12  Neighborhood   2842 non-null   object
 13  Condition1     2842 non-null   object
 14  Condition2     2842 non-null   object
 15  BldgType       2842 non-null   object
 16  HouseStyle     2842 non-null   object
 17  OverallQual    2842 non-null   int64
 18  OverallCond    2842 non-null   int64
 19  YearBuilt      2842 non-null   int64
...
 79  SaleCondition  2842 non-null   object
 80  SalePrice      1383 non-null   float64
dtypes: float64(12), int64(26), object(43)
memory usage: 1.8+ MB
```

2.2.2 Dữ liệu thiếu

```
full_df['MSSubClass'] = full_df['MSSubClass'].apply(str)
full_df['MoSold'] = full_df['MoSold'].apply(str)
```

Mục Đích của Việc Chuyển Đổi: Đảm bảo xử lý phù hợp: Khi các giá trị như MSSubClass và MoSold được xử lý như các chuỗi, các phân tích sau này có thể dễ dàng nhận biết đây là các biến phân loại, từ đó áp dụng các phương pháp phân tích thích hợp như mã hóa one-hot. Chuẩn bị cho mã hóa: Đối với các mô hình học máy, biến phân loại thường cần được mã hóa. Chuyển đổi các cột này sang chuỗi chuẩn bị chúng cho việc mã hóa one-hot hoặc các kỹ thuật mã hóa khác. Tránh nhầm lẫn: Giữ cho các giá trị này là chuỗi ngăn chặn việc thực hiện các phép tính toán học không mong muốn hoặc sai lầm trên các biến phân loại

Chúng ta xét các giá trị bị thiếu trong tập thử nghiệm

	% Missing Values
PoolQC	99.718508
MiscFeature	96.340605
Alley	93.033075
Fence	80.084448
SalePrice	51.337087
FireplaceQu	49.859254
LotFrontage	16.889514
GarageCond	5.594652
GarageYrBlt	5.594652
GarageFinish	5.594652
GarageQual	5.594652
GarageType	5.524279
BsmtExposure	2.885292
BsmtCond	2.885292
BsmtQual	2.850106
BsmtFinType1	2.779733
BsmtFinType2	2.779733
MasVnrType	0.809289
MasVnrArea	0.774103
MSZoning	0.140746
Functional	0.070373
BsmtHalfBath	0.070373
BsmtFullBath	0.070373
Utilities	0.070373
...	
Exterior2nd	0.035186
Exterior1st	0.035186
SaleType	0.035186
Electrical	0.035186

Chúng ta sẽ không xử lý ‘SalePrice’ vì đó là mục tiêu trong tập kiểm tra

```
mode_cols = 'Electrical, SaleType, Exterior1st, Exterior2nd, KitchenQual, Utilities, Functional, MSZoning'.split(', ')

for col in mode_cols:
    full_df[col] = full_df[col].fillna(full_df[col].mode()[0])
```

Các cột xử lý: 'Electrical', 'SaleType', 'Exterior1st', 'Exterior2nd', 'KitchenQual', 'Utilities', 'Functional', 'MSZoning'. Sử dụng vòng lặp for để xử lý từng cột trong danh sách mode_cols. Với mỗi cột, tìm giá trị xuất hiện nhiều nhất (mode) và dùng giá trị đó để điền vào chỗ trống (giá trị bị thiếu) trong cột.

```
NA_cols = 'PoolQC, MiscFeature, Alley, Fence, FireplaceQu, GarageCond, GarageFinish, GarageQual, GarageType, BsmtExposure, BsmtCond, BsmtQual,

for col in NA_cols:
    full_df[col] = full_df[col].fillna("NA")
```

Tạo danh sách NA_cols gồm các cột cần điền giá trị "NA", bao gồm 'PoolQC, MiscFeature, Alley, Fence, FireplaceQu, GarageCond, GarageFinish, GarageQual, GarageType, BsmtExposure, BsmtCond, BsmtQual, BsmtFinType2, BsmtFinType1'. Điền giá trị "NA"Sử dụng một vòng lặp for để xử lý từng cột trong danh sách NA_cols. Với mỗi cột, điền vào các giá trị bị thiếu (NaN) bằng chuỗi "NA". Phương pháp này giúp xử lý các giá trị bị thiếu bằng cách sử dụng "NA" để chỉ ra rằng một tính năng nhất định không tồn tại hoặc không áp dụng cho một quan sát cụ thể. Điều này thường được dùng trong các dữ liệu bất động sản, nơi mà không phải tất cả các nhà đều có garage, bể bơi, lối đi, v.v. Việc điền "NA" giúp đảm bảo dữ liệu được hoàn chỉnh, cho phép các phân tích hoặc mô hình hóa sau này không bị ảnh hưởng bởi các giá trị bị thiếu và phản ánh chính xác tình trạng của các tính năng tại các ngôi nhà được khảo sát.

```
full_df['MasVnrType'] = full_df['MasVnrType'].fillna('None')
```

Điền 'None' vào cột 'MasVnrType' giúp xử lý các trường hợp mà thông tin về loại vật liệu trang trí mặt ngoài không có sẵn . Giá trị 'None' ở đây có ý nghĩa rằng "không có giá trị trong loại vật liệu trang trí mặt ngoài"

```
null_cols = 'TotalBsmtSF, BsmtUnfSF, GarageCars, GarageArea, BsmtFinSF2, BsmtFinSF1, BsmtHalfBath, BsmtFullBath, GarageYrBlt, MasVnrArea'

for col in null_cols:
    full_df[col] = full_df[col].fillna(0)
```

‘Null_cols’: Tạo danh sách các cột cần xử lý, bao gồm các cột: 'TotalBsmtSF, BsmtUnfSF, GarageCars, GarageArea, BsmtFinSF2, BsmtFinSF1, BsmtHalfBath, BsmtFullBath, GarageYrBlt, MasVnrArea' Các cột này chứa các thông số về diện tích nhà, garage và các tính năng bổ sung khác của ngôi nhà. Việc điền 0 vào các cột này phản ánh ý nghĩa rằng nếu không có thông tin về một tính năng nhất định (ví dụ như diện tích tầng hầm, số xe garage có thể chứa, hoặc năm xây dựng garage), thì giá trị mặc định là 0, tức là không có diện tích, không có garage, vv.

```
mean_lot_frontage = full_df.groupby('Neighborhood')['LotFrontage'].mean()
```

Tính Giá Trị Trung Bình Của 'LotFrontage'. Dòng này sử dụng phương thức groupby để nhóm dữ liệu theo cột 'Neighborhood'. Sau đó, nó lấy cột 'LotFrontage' (chiều dài mặt tiền của lô đất) và tính giá trị trung bình (mean()) của cột này cho mỗi nhóm (mỗi khu vực lân cận).

```
mapping = dict(zip(mean_lot_frontage.index,mean_lot_frontage))

full_df['LotFrontage'] = full_df['LotFrontage'].fillna(full_df['Neighborhood'].map(mapping))
```

Tạo Bản Đồ Ánh Xạ. Dòng này tạo một từ điển (dict) từ các giá trị trung bình đã tính được. ‘mean_lot_frontage.index’ chứa tên các khu vực lân cận, và ‘mean_lot_frontage’ là giá trị trung bình tương ứng của 'LotFrontage' cho mỗi khu vực đó. Điền giá trị bị thiếu trong ‘LotFrontage’

df['LotFrontage']=df['LotFrontage'].fillna(df['Neighborhood'].map(mapping))’

Đây là bước cuối cùng nơi chúng ta dùng phương thức fillna() để điền các giá trị bị thiếu trong cột 'LotFrontage'. df['Neighborhood'].map(mapping) sử dụng từ điển mapping để thay thế mỗi giá trị trong 'Neighborhood' bằng giá trị trung bình của 'LotFrontage' đã tính toán cho khu vực đó. Nếu một hàng trong df có giá trị 'Neighborhood' là "A", và giá trị trung bình của 'LotFrontage' cho "A" là 50, thì 50 sẽ được dùng để điền vào khoảng trống tại hàng đó trong cột 'LotFrontage'. Điều này đảm

bảo rằng các giá trị được điền vào phản ánh chính xác đặc điểm của khu vực lân cận, giúp duy trì tính nhất quán và độ chính xác của dữ liệu. Cách tiếp cận này giúp giải quyết vấn đề dữ liệu bị thiếu mà không làm mất đi thông tin quan trọng hoặc bóp méo các phân tích thống kê sau này.

2.2.3 Biến đổi thuộc tính phân loại

```
df = pd.get_dummies(df,drop_first=True)
```

Áp dụng kỹ thuật mã hóa one-hot cho tất cả các cột phân loại (categorical) trong DataFrame . Mã hóa one-hot là một phương pháp phổ biến để chuyển đổi các biến phân loại thành các biến số nhị phân (binary) để có thể sử dụng chúng trong các mô hình học máy.

Cách Thức Hoạt Động:

- pd.get_dummies(df): Phương thức get_dummies của pandas được sử dụng để tạo ra một DataFrame mới từ df bằng cách chuyển đổi mỗi giá trị duy nhất trong mỗi cột phân loại thành một cột mới. Mỗi cột mới này sẽ có giá trị 1 nếu hàng tương ứng có giá trị đó trong cột ban đầu và 0 nếu không có.
- drop_first=True: Tùy chọn này được sử dụng để tránh vấn đề đa cộng tuyến (multicollinearity) trong mô hình học máy, nơi hai hoặc nhiều biến dự đoán có mối quan hệ tuyến tính hoàn hảo. Bằng cách đặt drop_first=True, chúng ta loại bỏ một trong các cột được tạo cho mỗi biến phân loại, cụ thể là cột đầu tiên. Điều này giúp giảm số lượng đặc trưng tổng thể và giữ lại tính khả dụng của dữ liệu mà không làm mất đi thông tin quan trọng.

Mục Đích:

- Biến Đổi Dữ Liệu Phân Loại: Một số thuật toán học máy không thể xử lý trực tiếp dữ liệu phân loại hoặc chỉ xử lý dữ liệu số. Việc chuyển đổi các biến phân loại thành số nhị phân giúp những thuật toán này có thể dễ dàng phân tích dữ liệu.

- Chuẩn Bị Dữ Liệu: Mã hóa one-hot là bước chuẩn bị dữ liệu cần thiết để đảm bảo rằng các mô hình học máy nhận được đầu vào thích hợp và có thể phát hiện ra các mẫu hoặc xu hướng từ các biến phân loại.
- Tránh Multicollinearity: Loại bỏ cột đầu tiên sau mã hóa one-hot giúp giảm khả năng xảy ra multicollinearity, từ đó cải thiện độ chính xác và hiệu quả của mô hình học máy.

```
categorical_features = full_df.select_dtypes(include=['object']).columns.tolist()
categorical_summary = full_df[categorical_features].head()
categorical_features, categorical_summary
```

✓ 0.0s

```
([],
 Empty DataFrame
 Columns: []
 Index: [0, 1, 2, 3, 4])
```

2.3 Mô hình hoá

Chuẩn bị dữ liệu

```
df = full_df[full_df['SalePrice'].notnull()]
test_df = full_df[full_df['SalePrice'].isnull()].drop('SalePrice', axis=1)
```
✓ 0.0s

```
X = df.drop('SalePrice',axis=1)
y = df['SalePrice']
```
✓ 0.0s

```
df.shape
```
✓ 0.0s

```
(1383, 284)
```

```
test_df.shape
```
✓ 0.0s

```
(1459, 283)
```

Chia dữ liệu

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Chia 30% dữ liệu sẽ được dùng để kiểm tra, và 70% còn lại sẽ được dùng để huấn luyện mô hình.

Các chỉ số đánh giá thuật toán:

1. MAE (Mean Absolute Error): MAE là trung bình của các giá trị tuyệt đối của sai số giữa giá trị dự đoán và giá trị thực tế.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

2. RMSE (Root Mean Squared Error): RMSE là căn bậc hai của trung bình các bình phương sai số giữa giá trị dự đoán và giá trị thực tế.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3. R^2 (R-squared): R^2 đo lường mức độ mà mô hình dự đoán giải thích được phương sai của biến phụ thuộc.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$$

2.3.1 Hồi quy tuyến tính

```
linear_model = LinearRegression()
linear_model.fit(X_train,y_train)
y_lin_pred = linear_model.predict(X_test)
mae = mean_absolute_error(y_test, y_lin_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_lin_pred))
r2 = r2_score(y_test, y_lin_pred)
print(mae,rmse,r2)
```

✓ 0.2s

```
14599.440031910162 21717.312630358134 0.861797425967376
```

Đánh giá hiệu suất của mô hình hồi quy tuyến tính:

- MAE (Mean Absolute Error): 14599.44
- RMSE (Root Mean Squared Error): 21717.32
- R^2 (Coefficient of Determination): 0.862

2.3.2 Hồi quy Ridge

```
ridge_model = RidgeCV(alphas=(0.1, 0.5, 1, 5, 10, 50, 100), scoring='neg_mean_squared_error')
ridge_model.fit(X_train,y_train)
y_ridge_pred = ridge_model.predict(X_test)
mae = mean_absolute_error(y_test, y_ridge_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_ridge_pred))
r2 = r2_score(y_test, y_ridge_pred)

print(mae,rmse,r2)
```

✓ 0.2s

```
12660.832152699535 18407.28167603113 0.9007150819628935
```

Đánh giá hiệu suất của mô hình hồi quy Ridge:

- MAE (Mean Absolute Error): 12660.832
- RMSE (Root Mean Squared Error): 18407.281
- R^2 (Coefficient of Determination): 0.9

2.3.3 Hồi quy Lasso

```
lasso_model = LassoCV(alphas=[1, 10, 100], cv=10, random_state=42, max_iter=100000)
lasso_model.fit(X_train,y_train)
y_lasso_pred = lasso_model.predict(X_test)
mae = mean_absolute_error(y_test, y_lasso_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_lasso_pred))
r2 = r2_score(y_test, y_lasso_pred)

print(mae,rmse,r2)
```

✓ 18.0s

```
12521.666997370734 18169.461896040866 0.9032640064880179
```

Đánh giá hiệu suất của mô hình hồi quy Lasso:

- MAE (Mean Absolute Error): 12521.666
- RMSE (Root Mean Squared Error): 18169.46
- R^2 (Coefficient of Determination): 0.903

2.3.4 Hồi quy Elasstic

```
elastic_model = ElasticNetCV(l1_ratio=[.1, .5, .7, .9, .95, .99, 1], cv=10)
elastic_model.fit(X_train, y_train)
y_elastic_pred = elastic_model.predict(X_test)

mae = mean_absolute_error(y_test, y_elastic_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_elastic_pred))
r2 = r2_score(y_test, y_elastic_pred)

print(mae,rmse,r2)
```

✓ 6.2s

```
19631.88740371119 27386.33250512344 0.7802283246622009
```

Đánh giá hiệu suất của mô hình hồi quy Elasstic:

- MAE (Mean Absolute Error): 19631.88
- RMSE (Root Mean Squared Error): 27386.33
- R^2 (Coefficient of Determination): 0.78

Nhận xét chụng:

Dựa trên các chỉ số đánh giá hiệu suất của bốn mô hình hồi quy (Tuyến tính, Ridge, Lasso, và Elastic Net), có thể đưa ra các nhận xét tổng quát sau:

1. Hiệu suất tổng thể:
    - Mô hình hồi quy Lasso có hiệu suất tốt nhất với các chỉ số MAE, RMSE thấp nhất và R^2 cao nhất. Điều này cho thấy Lasso không chỉ giảm thiểu sai số mà còn giải thích tốt hơn sự biến động của giá nhà.
    - Mô hình hồi quy Ridge cũng cho thấy hiệu suất tốt với cải thiện đáng kể so với mô hình hồi quy tuyến tính.
    - Mô hình hồi quy tuyến tính cơ bản có hiệu suất tương đối thấp hơn so với Ridge và Lasso nhưng vẫn tốt hơn so với Elastic Net.
    - Mô hình hồi quy Elastic Net có hiệu suất kém nhất trong bốn mô hình, với MAE cao nhất và R^2 thấp nhất, cho thấy khả năng dự đoán yếu hơn.

2. MAE (Mean Absolute Error):
    - MAE đo lường độ chính xác trung bình của dự đoán. Mô hình Lasso có MAE thấp nhất (12521.666), cho thấy dự đoán của mô hình này là gần đúng nhất với giá trị thực tế.
    - Mô hình Elastic Net có MAE cao nhất (19631.88), cho thấy sai số trung bình lớn nhất trong dự đoán.
3. RMSE (Root Mean Squared Error):
    - RMSE đo lường độ lệch chuẩn của các sai số dự đoán. Mô hình Lasso có RMSE thấp nhất (18169), cho thấy độ lệch chuẩn của các sai số dự đoán là nhỏ nhất.
4. R^2 (Coefficient of Determination):
    - R^2 đo lường mức độ mà biến phụ thuộc (giá nhà) được giải thích bởi biến độc lập trong mô hình. Mô hình Lasso có R^2 cao nhất (0.903), cho thấy khả năng giải thích sự biến động của giá nhà là tốt nhất.
    - Mô hình Elastic Net có R^2 thấp nhất (0.78), cho thấy khả năng giải thích kém nhất.

Kết luận:

- Mô hình Lasso là lựa chọn tốt nhất trong các mô hình hồi quy mà bạn đã thử nghiệm. Nó có khả năng dự đoán tốt nhất với sai số thấp nhất và khả năng giải thích sự biến động của giá nhà cao nhất.
- Mô hình Ridge cũng là một lựa chọn tốt và có hiệu suất gần với mô hình Lasso. Nếu bạn muốn một sự cân bằng giữa độ chính xác và đơn giản, Ridge cũng là một sự lựa chọn hợp lý.
- Mô hình hồi quy tuyến tính cơ bản có hiệu suất kém hơn so với Lasso và Ridge nhưng vẫn có thể sử dụng trong trường hợp cần một mô hình đơn giản và dễ hiểu.

- Mô hình Elastic Net có hiệu suất kém nhất và cần được điều chỉnh thêm hoặc thay thế bằng các mô hình khác đã cho kết quả tốt hơn.

2.3.5 Chọn lựa mô hình và dự đoán:

Từ 4 mô hình đã xét, ta chọn được mô hình hồi quy Lasso vì có hiệu suất mô hình tốt nhất. Dự đoán giá nhà trên tập thử nghiệm:

```
lasso = Lasso(alpha=100)
lasso.fit(X,y)
predictions = lasso.predict(test_df)
```

✓ 0.1s

```
submission1= pd.DataFrame(data = {'Id': range(1461,2920),
                                  'SalePrice': predictions})
submission1
```

✓ 0.0s

Dữ liệu giá bán được dự đoán:

	Id	SalePrice
0	1461	123921.609413
1	1462	158071.341018
2	1463	181430.079482
3	1464	199224.333140
4	1465	188129.200051
...	...	...
1454	2915	79741.809260
1455	2916	72662.980156
1456	2917	175248.943400
1457	2918	113349.029244
1458	2919	210729.590673

1459 rows × 2 columns

```
submission1.to_csv("submission1.csv", index=False)
```

✓ 0.0s

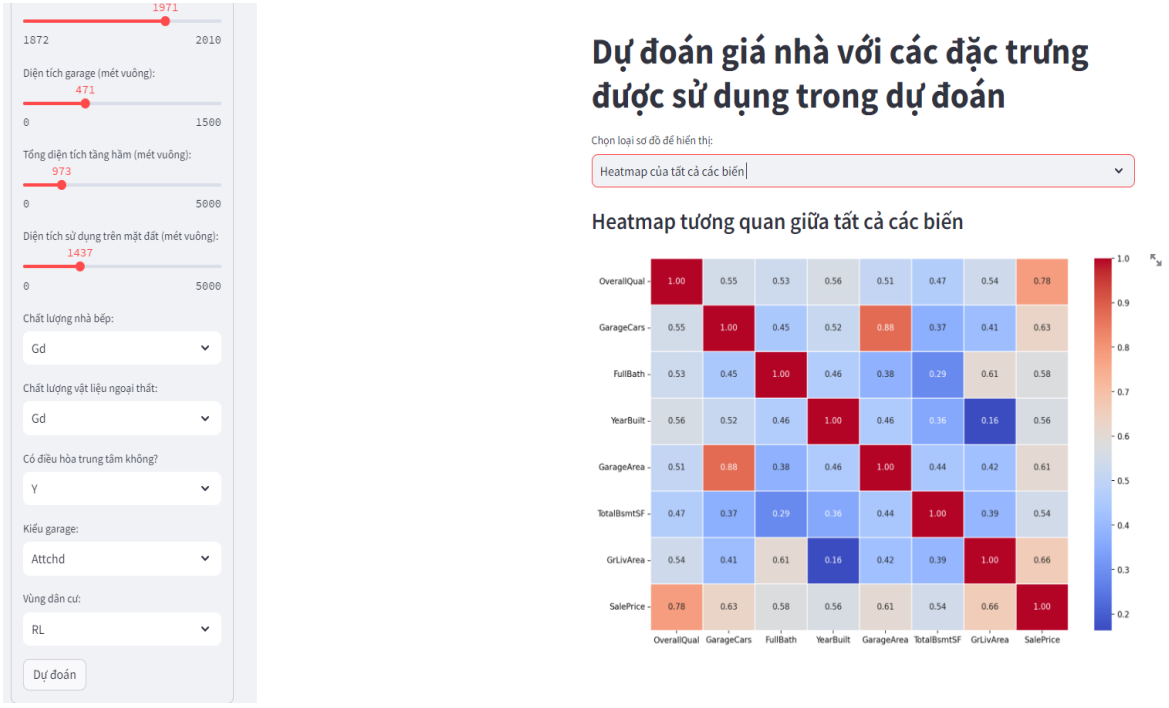
2.4 Giao diện dự đoán

2.4.1 Chọn biến dữ liệu

Do dữ liệu khá lớn nên trong phần giao diện dự đoán chúng ta sẽ chỉ chọn các biến dữ liệu phù hợp để làm mô hình dự đoán gồm: 'OverallQual', 'GarageCars', 'FullBath', 'YearBuilt', 'GarageArea', 'TotalBsmtSF', 'GrLivArea', 'KitchenQual', 'ExterQual', 'CentralAir', 'GarageType', 'MSZoning'. Có thể thấy lựa chọn các biến này rất phù hợp vì đây là những yếu tố quan trọng trong việc mua bán nhà và là mối quan tâm hàng đầu của người mua đối với căn nhà họ quan tâm.

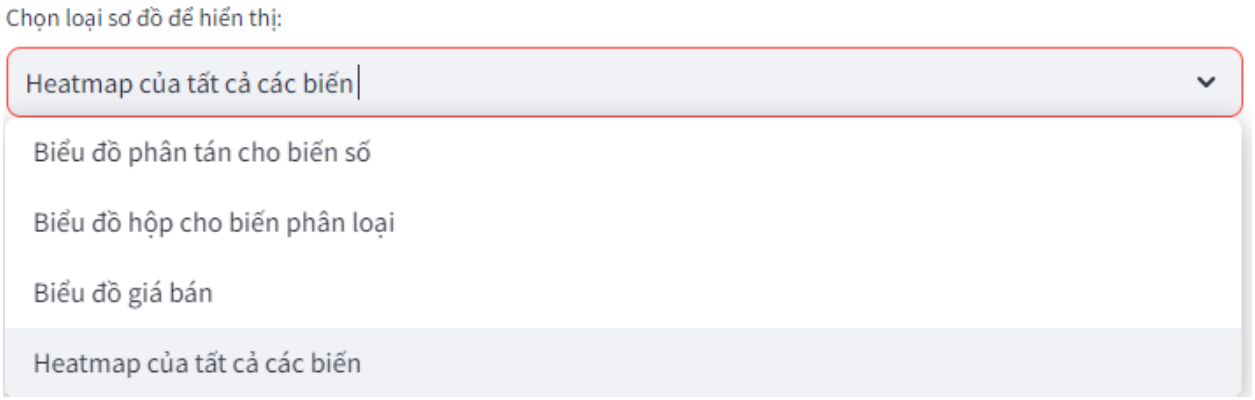
2.4.2 Giao diện và các tính năng

Hình 1. 9 Giao diện và tính năng



Hình 1. 10 Mô tả giao diện

Trong giao diện chúng ta sẽ có hai phần một là cột tính năng nằm về bên trái gồm có các thanh trượt để có thay đổi số liệu trong các biến định lượng và các hộp để chọn đặc điểm tính năng trong các biến phân loại, dưới cùng của cột tính năng là nút dự đoán. Còn lại là cột biểu đồ để hiện các biểu đồ tương ứng nhằm nêu lên mối quan hệ giữa các cột và Giá bán.



Hình 1. 11 Lựa chọn sơ đồ

Cuối cùng sau khi lựa chọn số liệu và các đặc điểm phù hợp chúng ta sẽ nhấn nút 'Dự Đoán' và giá bán của căn nhà sẽ hiện ra bên dưới cột biểu đồ.

Ví dụ minh họa:

Hình 1. 12 Hiện giá trị dự đoán

Phần III: Kết luận và hướng phát triển.

❖ Đã Đạt Được :

Trong dự án này, đồ án đã phát triển một mô hình dự đoán giá nhà sử dụng các phương pháp hồi quy như Lasso để xác định các yếu tố ảnh hưởng đến giá bất động sản tại Ames, Iowa. Mô hình đã cho thấy khả năng dự đoán chính xác với các chỉ số đánh giá như MAE và RMSE đều trong ngưỡng chấp nhận được, cung cấp một công cụ hữu ích cho những người quan tâm đến việc đánh giá giá trị bất động sản.

❖ Hạn Chế :

Tuy nhiên, mô hình còn một số hạn chế như:

1. Phạm vi dữ liệu hạn chế : Mô hình chỉ dựa trên dữ liệu từ Ames, có thể không phản ánh chính xác thị trường bất động sản ở các khu vực khác.
2. Giới hạn của giao diện: Chưa đưa được toàn các biến dự đoán vào giao diện dự đoán , mới chỉ dừng lại được ở mức minh họa .
3. Xử lý Dữ liệu: Các vấn đề về dữ liệu thiếu và ngoại lai cần được áp dụng nhiều phương thức xử lý khác nhau để xử lý một cách chi tiết hơn để không ảnh hưởng đến hiệu suất của mô hình.

❖ Hướng Phát Triển :

Để cải thiện mô hình hiện tại và khắc phục các hạn chế, đề xuất các bước phát triển tiếp theo như sau:

1. Mở Rộng Tập Dữ liệu: Thu thập dữ liệu từ nhiều khu vực địa lý khác nhau để cải thiện khả năng tổng quát của mô hình.
2. Cải Tiến Kỹ Thuật: Áp dụng các kỹ thuật học sâu hoặc học tăng cường để khám phá các mối quan hệ phức tạp hơn giữa các biến.
3. Phân Tích Đa Dạng: Xem xét ảnh hưởng của các yếu tố vĩ mô như kinh tế và chính trị đến giá bất động sản.

Những hướng phát triển này không chỉ giúp tăng cường khả năng dự đoán của mô hình mà còn mở rộng ứng dụng của nó trong thực tiễn, hỗ trợ quyết định chính xác hơn trong các giao dịch bất động sản.

TÀI LIỆU THAM KHẢO

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.
2. De Cock, D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. Journal of Statistics Education, 19(3).
3. . Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing.