Phân tích dữ liệu và dự đoán giá nhà bằng thuật toán học máy

Thành viên: Lê Đoàn Kim Ngân, Lê Thị Trúc Ly, Lâm Tú Nhi, Nguyễn Đăng Khoa.

Liên hệ : E-mail(s) : khoanguyen12062005@gmail.com;

Đồng tác giả: nhilam150@gmail.com;

imtrucly@gmail.com;

ledoankimngan2005@gmail.com;

Các tác giả đóng góp công bằng trong bài báo này.

Tóm tắt:

Trong bối cảnh đô thị hóa nhanh và nhu cầu mua bán bất động sản ngày càng tăng, việc dự đoán giá nhà chính xác đóng vai trò quan trọng trong phân tích thị trường và hỗ trợ ra quyết định cho cả người mua lẫn nhà đầu tư. Nghiên cứu này tập trung xây dựng mô hình dự đoán giá nhà dựa trên bộ dữ liệu *House Prices: Advanced Regression Techniques* thông qua quy trình gồm các bước: phân tích khám phá dữ liệu (EDA), xử lý dữ liệu thiếu, mã hóa biến phân loại, chuẩn hóa đặc trưng và huấn luyện mô hình học máy. Các mô hình hồi quy được triển khai và so sánh bao gồm **Linear Regression, Lasso Regression, K-Nearest Neighbors, Support Vector Regression, Decision Tree, Random Forest và Gradient Boosting Regressor.** Hiệu năng của mô hình được đánh giá bằng các chỉ số RMSE và R^2. Kết quả cho thấy **CatBoosting** đạt hiệu quả dự đoán cao nhất, với giá trị RMSE thấp và R^2 cao, chứng tỏ khả năng nắm bắt mối quan hệ phi tuyến giữa các thuộc tính và giá nhà. Nghiên cứu góp phần khẳng định tiềm năng của các mô hình ensemble trong bài toán dự đoán giá bất động sản.

Từ khóa : Dự đoán giá nhà; Học máy; Hồi quy; Phân tích dữ liệu khám phá (EDA); Random Forest; Gradient Boosting; Xử lý dữ liệu thiếu; Chuẩn hóa đặc trưng.

1. Giới thiệu

Trong những năm gần đây, cùng với sự phát triển nhanh chóng của đô thị hóa và thị trường bất động sản, nhu cầu dự đoán giá nhà chính xác trở thành một yêu cầu cấp thiết nhằm hỗ trợ ra quyết định cho nhà đầu tư, tổ chức tài chính và người mua nhà. Giá nhà chịu ảnh hưởng của nhiều yếu tố phức tạp như vị trí địa lý, diện tích, vật liệu xây dựng, số lượng phòng, tiện ích xung quanh và các đặc điểm kiến trúc khác. Do đó, việc xây dựng mô hình dự đoán hiệu quả đòi hỏi phải khai thác được mối quan hệ phi tuyến giữa các thuộc tính này.

Các phương pháp truyền thống như hồi quy tuyến tính đơn giản thường không đủ khả năng mô tả tính phức tạp của dữ liệu bất động sản hiện nay. Sự phát triển của **học máy (Machine Learning)** đã mở ra hướng tiếp cận mới, cho phép mô hình tự động học từ dữ liệu và cải thiện độ chính xác của dự đoán. Tuy nhiên, việc lựa chọn mô hình phù hợp và xử lý dữ liệu đầu vào đúng cách vẫn là thách thức lớn, đặc biệt khi dữ liệu chứa nhiều giá trị thiếu, biến phân loại, và sự khác biệt về phân phối.

Nghiên cứu này tập trung vào việc **phân tích dữ liệu và so sánh hiệu năng của các mô hình hồi quy** trong bài toán dự đoán giá nhà, sử dụng bộ dữ liệu *House Prices: Advanced Regression Techniques* do Kaggle cung cấp. Quy trình nghiên cứu bao gồm: phân tích khám phá dữ liệu (EDA), xử lý dữ liệu thiếu, mã hóa và chuẩn hóa đặc trưng, huấn luyện nhiều mô hình hồi quy khác nhau, và đánh giá kết quả dựa trên các chỉ số **RMSE** và $\mathbf{R^2}$.

Kết quả cho thấy các mô hình *ensemble* như *Random Forest* và *Gradient Boosting* đạt hiệu năng cao, tuy nhiên *CatBoost* thể hiện *hiệu suất vượt trội nhất* trong tất cả các mô hình được thử nghiệm. Điều này chứng tỏ khả năng mạnh mẽ của CatBoost trong việc *xử lý dữ liệu dạng bảng (tabular data), mô hình hóa các mối quan hệ phi tuyến phức tạp*, đồng thời *giảm thiểu hiện tượng overfitting* nhờ cơ chế xử lý đặc trưng hạng mục (categorical features) tối ưu.

2 . Tổng quan và các nghiên cứu liên quan

2.1 . Tổng quan về bài toán

Bài toán dự đoán giá nhà (*House Price Prediction*) là một trong những ứng dụng điển hình của học máy trong lĩnh vực kinh tế và bất động sản. Mục tiêu của bài toán là xây dựng mô hình dự đoán giá bán của một căn nhà dựa trên các đặc trưng sẵn có như diện tích, chất lượng xây dựng, số phòng, vị trí, năm xây dựng, diện tích tầng hầm hay gara,… Việc ước lượng chính xác giá nhà có ý nghĩa quan trọng trong nhiều khía cạnh như hỗ trợ người mua và người bán ra quyết định, giúp ngân hàng và tổ chức tài chính đánh giá tài sản thế chấp, hoặc hỗ trợ các công ty bất động sản trong chiến lược định giá và đầu tư.

Tuy nhiên, bài toán này không đơn giản vì giá nhà chịu ảnh hưởng của nhiều yếu tố phi tuyến, có sự tương quan phức tạp giữa các thuộc tính, đồng thời dữ liệu thường chứa nhiễu và sự khác biệt giữa các khu vực địa lý. Do đó, việc lựa chọn mô hình học máy phù hợp và xử lý dữ liệu hiệu quả là yếu tố then chốt để đạt được kết quả dự đoán chính xác. Trong bối cảnh đó, các phương pháp học máy hiện đại như *Random Forest, Gradient Boosting, XGBoost và CatBoost* ngày càng được sử dụng rộng rãi nhằm khai thác tốt hơn mối quan hệ phi tuyến giữa các biến đầu vào và giá bán.

2.2 . Các nghiên cứu

Các hướng tiếp cận học máy: Học máy được ứng dụng rộng rãi trong các bài toán hồi quy, đặc biệt là dự đoán giá nhà – một bài toán kinh điển trong phân tích dữ liệu bất động sản. Các công trình đầu tiên sử dụng *hồi quy tuyến tính đa biến (Multiple Linear Regression - MLR)* để xác định mối liên hệ giữa đặc trưng vật lý của ngôi nhà (diện tích, số phòng, vị trí, năm xây dựng, chất lượng vật liệu) và giá bán. Dù có ưu điểm về khả năng giải thích, các mô hình tuyến tính gặp hạn chế khi quan hệ giữa biến độc lập và phụ thuộc mang tính phi tuyến. Để cải thiện, nhiều nghiên cứu áp dụng *hồi quy có điều chuẩn như Ridge, Lasso*,... giúp giảm hiện tượng đa cộng tuyến, hạn chế overfitting, đồng thời chọn lọc được các đặc trưng quan trọng. Song song đó, *tiền xử lý dữ liệu* được xem là khâu quyết định độ chính xác của mô hình. Các công trình gần đây nhấn mạnh việc xử lý giá trị thiếu theo ngữ cảnh, mã hóa biến phân loại bằng *One-Hot Encoding*, biến đổi *log(SalePrice*) để ổn định phương sai, và chuẩn hóa dữ liệu liên tục. Ngoài ra, *feature engineering* – như tạo đặc trưng tổng diện tích, tỷ lệ diện tích sàn, hay tuổi nhà – cũng được chứng minh là làm tăng đáng kể hiệu năng mô hình.

Phương pháp học ensemble: Các nghiên cứu hiện đại chỉ ra rằng *ensemble learning* là một trong những hướng tiếp cận hiệu quả nhất cho bài toán dự đoán giá nhà. Các mô hình như *CatBoosting, Gradient Boosting Machine (GBM), XGBoost, LightGBM*, ... có khả năng mô hình hóa quan hệ phi tuyến, tương tác đặc trưng, và xử lý dữ liệu có cả biến phân loại lẫn biến số. Bên cạnh đó, nhiều công trình đã đề xuất sử dụng *Stacking* hoặc *Blending* để kết hợp nhiều mô hình khác nhau — ví dụ: hồi quy tuyến tính, mô hình cây và phương pháp kernel — nhằm khai thác thế mạnh riêng của từng nhóm mô hình. Các nghiên cứu thực nghiệm trên bộ dữ liệu *Ames Housing* cho thấy các thuật toán boosting hiện đại thường đạt *RMSE thấp nhất*, và được coi là tiêu chuẩn đánh giá trong dự đoán giá nhà.

Các phương pháp nâng cao và mô hình kernel: Ngoài các mô hình tuyến tính và cây quyết định, một số công trình cũng nghiên cứu việc áp dụng các thuật toán *phi tuyến* như *Support Vector Regression (SVR), K-Nearest Neighbors (KNN)* cho bài toán này. Các phương pháp này đặc biệt hiệu quả khi mối quan hệ giữa đặc trưng và giá nhà mang tính cục bộ hoặc không thể biểu diễn tuyến tính. SVR sử dụng hàm kernel (Radial Basis Function, Polynomial) để ánh xạ dữ liệu sang không gian đặc trưng cao hơn, giúp mô hình hóa các quan hệ phức tạp hơn mà vẫn duy trì tính ổn định. KNN, trong khi đơn

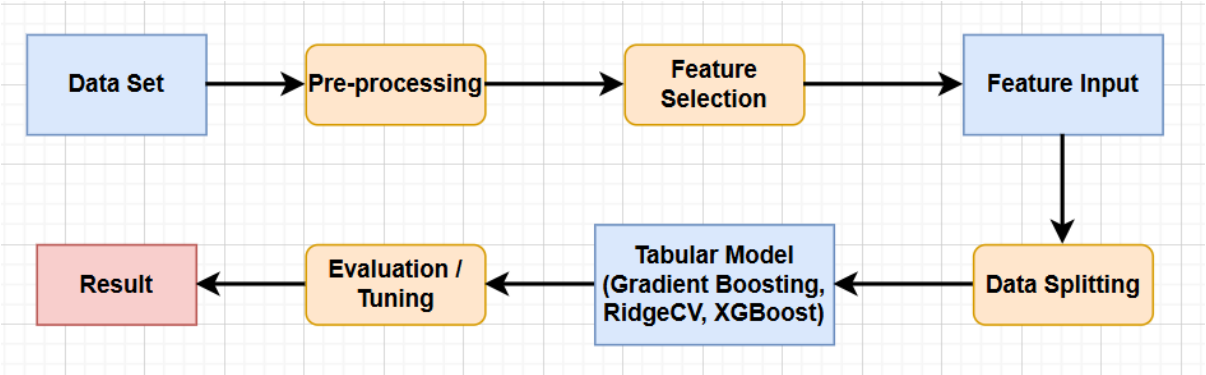
giản, lại mang tính trực quan và hoạt động tốt khi dữ liệu có phân cụm tự nhiên theo khu vực hoặc loại nhà. Tuy nhiên, các nghiên cứu cho thấy các mô hình kernel thường cần hiệu chỉnh siêu tham số (hyperparameter tuning) kỹ lưỡng và thời gian tính toán cao hơn so với các phương pháp boosting.

3 . Phương pháp

Bài báo này sử dụng *hai bộ dữ liệu khác nhau* để tiến hành huấn luyện và dự đoán về giá cả nhà ở. Mục tiêu của các thí nghiệm là đề xuất một giá cả thích hợp cho một căn nhà dựa trên những đặc điểm về căn nhà, môi trường xung quanh của căn nhà,.. từ một loạt thông tin đầu vào ngẫu nhiên không biết trước dựa trên việc học tập trước đó từ nhiều mẫu dữ liệu có sẵn.

Để nâng cao hiệu suất mô hình và giảm thời gian chạy, nhiều kỹ thuật *tiền xử lý dữ liệu* được áp dụng, bao gồm *lựa chọn đặc trưng, giảm số chiều dữ liệu, và xử lý giá trị thiếu.*

Các bước cơ bản từ lúc bắt đầu đến khi kết thúc của một quy trình huấn luyện mô hình



3.1 . Tiền xử lý dữ liệu

Dữ liệu thô thường chứa rất nhiều vấn đề gây cản trở đến việc huấn luyện và học tập của mô hình như : *giá trị bị thiếu, tỷ lệ đặc trưng không đồng nhất*, hoặc *dữ liệu dạng chuỗi (serial data)* không thể đưa trực tiếp vào mô hình. Do đó, *tiền xử lý dữ liệu* là một bước *quan trọng* nhằm *nâng cao chất lượng dữ liệu và cải thiện độ chính xác* của mô hình để có thể dự đoán được giá nhà một cách chính xác nhất.

Trước hết, các *đặc trưng có hơn 70% giá trị bị thiếu* trong bộ dữ liệu sẽ không được sử dụng trong việc huấn luyện , vì chúng cung cấp rất ít thông tin hữu ích cho quá trình huấn luyện. *Lựa chọn đặc trưng (feature selection) giúp giảm hiện tượng overfitting, rút ngắn thời gian huấn luyện, và nâng cao hiệu quả của mô hình.*

Với các tập dữ liệu lớn, *giảm chiều dữ liệu* là cần thiết để tiết kiệm *tài nguyên tính toán* và *thời gian huấn luyện.* Trong nghiên cứu này, chỉ *10% dữ liệu gốc* được sử dụng cho mục đích *kiểm thử (testing),* đồng thời *giữ nguyên tỷ lệ* giữa các giao dịch gian lận và hợp lệ. Sau khi lựa chọn đặc trưng, các *giá trị còn thiếu* được *điền bằng một hằng số cố định,* giúp mô hình *học được toàn bộ thông tin* mà *không bị ảnh hưởng bởi dữ liệu thiếu.*

Tiếp theo, *chuẩn hóa đặc trưng (feature scaling)* được áp dụng để đưa tất cả các đặc trưng về cùng một thang đo. Quá trình này sử dụng *chuẩn hóa Z-score*, được định nghĩa bởi công thức: $z = \frac{x - \mu}{\sigma}$ trong đó: x là giá trị dữ liệu, μ là giá trị trung bình, σ là độ lệch chuẩn.

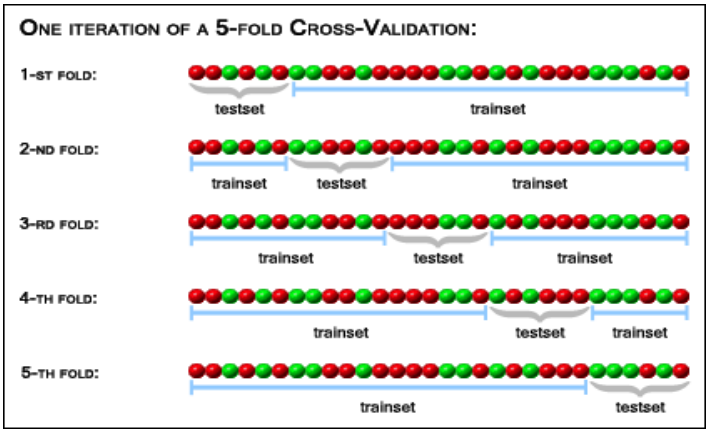
Ngoài ra, trong dữ liệu thực tế, các *đặc trưng phân loại (categorical features)* thường được biểu diễn dưới dạng *chuỗi ký tự (string)*, vốn *không tương thích với các mô hình học máy.* Để khắc phục vấn đề này, *phép mã hóa one-hot (one-hot encoding)* được sử dụng để *chuyển đổi chúng thành dạng nhị phân*, phù hợp cho quá trình huấn luyện mô hình.

3.2 . Chia dữ liệu

Ngoài việc tiền xử lý dữ liệu, để có thể huấn luyện mô hình với hiệu suất tốt nhất thì phương pháp chia dữ liệu cũng là một trong những yếu tố quan trọng để tăng tốc độ cũng như chất lượng của mô hình. Trong thí nghiệm lần này, chúng tôi sẽ sử dụng phương pháp **K-Fold Cross Validation** để tiến hành chia dữ liệu.

Đây là một kỹ thuật đánh giá mô hình học máy phổ biến, được sử dụng để đo lường khả năng tổng quát hóa của mô hình trên dữ liệu chưa thấy đối với các tập dữ liệu có số lượng mẫu vừa và nhỏ. Nguyên tắc của phương pháp này là chia tập dữ liệu thành **K** phần (fold) có kích thước gần bằng nhau. Lặp qua từng fold, fold thứ iii được dùng làm tập kiểm thử, trong khi **K−1** fold còn lại được dùng làm tập huấn luyện. Quá trình này được thực hiện **K** lần, mỗi lần sử dụng một fold khác làm tập kiểm thử, giúp đảm bảo rằng mọi dữ liệu đều được sử dụng cho huấn luyện và kiểm thử.

Sau khi hoàn tất, các kết quả đánh giá trên từng fold (ví dụ RMSE, R^2) được tổng hợp bằng *trung bình*: $\bar{s} = \frac{1}{K}\sum_{i=1}^{K} s_i$ trong đó s_i là giá trị chỉ số đánh giá của fold thứ i.



K-fold cross-validation giúp giảm *rủi ro đánh giá thiên lệch*, phát hiện *overfitting*, và cung cấp *ước lượng hiệu suất mô hình ổn định* hơn so với việc chia dữ liệu một lần. Đây là phương pháp được khuyến nghị cho hầu hết các bài toán học máy, đặc biệt khi dữ liệu hạn chế hoặc phân phối không đồng đều.

3.3 . RMSE (Root Mean Squared Error)

Đây là là một trong những chỉ số đánh giá phổ biến nhất cho các mô hình hồi quy. Chỉ số này đo lường mức độ sai lệch trung bình giữa giá trị dự đoán của mô hình và giá trị thực tế. RMSE nhấn mạnh các sai số lớn nhờ việc bình phương từng sai số trước khi tính trung bình, do đó nó cung cấp một đại lượng phản ánh *mức độ chính xác tổng thể* của mô hình. Trong thực tế, giá trị RMSE càng nhỏ, hiệu suất dự đoán của mô hình càng cao. **RMSE** được triển khai như sau : $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$. Trong đó, n là số lượng mẫu, y_i là giá trị thực tế và $\hat{y}_i$ là giá trị dự đoán của mẫu thứ i. Bằng cách bình phương sai số, RMSE đảm bảo rằng các sai số lớn được nhấn mạnh hơn, đồng thời, do lấy căn bậc hai của trung bình bình phương sai số, RMSE vẫn giữ *đơn vị tương đồng với biến mục tiêu*, giúp dễ dàng diễn giải và so sánh.

RMSE có ý nghĩa quan trọng trong đánh giá mô hình. Nó cung cấp thông tin trực quan về mức độ khác biệt giữa dự đoán và giá trị thực tế. Khi kết hợp với các chỉ số khác như MAE (Mean Absolute Error) hay R^2, RMSE cho phép người nghiên cứu đánh giá toàn diện về hiệu suất mô hình, đồng thời nhận biết các dự đoán lệch nhiều mà mô hình cần cải thiện.

Trong thực hành, RMSE thường được sử dụng để *so sánh hiệu suất của các mô hình hồi quy* trên cùng một tập dữ liệu hoặc trong quá trình *cross-validation*. Khi áp dụng K-fold cross-validation, RMSE được tính trên từng fold và sau đó lấy giá trị trung bình để đánh giá hiệu suất tổng thể, giúp giảm thiên lệch do phân chia dữ liệu ngẫu nhiên.

Tuy nhiên, RMSE cũng có một số hạn chế. Vì nhấn mạnh các sai số lớn, RMSE có thể bị ảnh hưởng quá mức bởi các outlier trong dữ liệu. Do đó, trong các tập dữ liệu có nhiều giá trị ngoại lai, RMSE nên được sử dụng cùng các chỉ số khác như MAE để đưa ra đánh giá toàn diện hơn.

3.4. Các mô hình được sử dụng

Thí nghiệm này được thực nghiệm với 11 mô hình chính là: *Linear Regression, Ridge, Lasso, ElasticNet, Random Forest, Gradient Boosting, XGBoost, LightGB, CatBoost, SVR, K-Neighbors.* Các mô hình này được phân chia thành bốn nhóm chính dựa trên nguyên lý học và cấu trúc thuật toán: *nhóm tuyến tính (Linear Models), nhóm dựa trên cây quyết định (Tree-based Models), nhóm boosting nâng cao (Boosting Models) và nhóm khác.*

Nhóm tuyến tính bao gồm Linear Regression, Ridge, Lasso và ElasticNet, là những mô hình cơ bản trong học máy với giả định mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc. Linear Regression là mô hình đơn giản nhất, đóng vai trò làm chuẩn tham chiếu cơ sở, cho phép đánh giá khả năng dự đoán ban đầu của dữ liệu. Tuy nhiên, mô hình này thường gặp hạn chế trong trường hợp tồn tại hiện tượng đa cộng tuyến hoặc dữ liệu có nhiễu cao. Ridge Regression được cải tiến bằng cách bổ sung hệ số phạt L2, giúp giảm độ lớn của các trọng số và hạn chế overfitting, trong khi Lasso Regression sử dụng hệ số phạt L1 để tự động loại bỏ những đặc trưng không quan trọng, làm cho mô hình trở nên gọn nhẹ và dễ diễn giải hơn. ElasticNet là sự kết hợp giữa hai phương pháp L1 và L2, vừa duy trì được khả năng chọn lọc đặc trưng của Lasso, vừa đảm bảo tính ổn định của Ridge, giúp mô hình hoạt động hiệu quả hơn trong các tập dữ liệu có nhiều biến tương quan chặt chẽ.

Nhóm mô hình dựa trên cây quyết định bao gồm Random Forest và Gradient Boosting, đều là các phương pháp học tập hợp (ensemble learning) sử dụng nhiều cây quyết định để cải thiện hiệu suất. Random Forest xây dựng một tập hợp các cây được huấn luyện độc lập trên các mẫu dữ liệu ngẫu nhiên, sau đó tổng hợp kết quả dự đoán bằng cách trung bình hóa hoặc bỏ phiếu đa số. Phương pháp này có khả năng giảm

phương sai, tăng tính ổn định và khả năng khái quát hóa của mô hình. Gradient Boosting, ngược lại, huấn luyện các cây quyết định theo trình tự, trong đó mỗi cây mới được xây dựng nhằm sửa lỗi của cây trước, giúp cải thiện dần độ chính xác. Hai mô hình này có ưu điểm nổi bật là khả năng mô hình hóa mối quan hệ phi tuyến và tương tác giữa các đặc trưng, đồng thời không yêu cầu chuẩn hóa dữ liệu đầu vào. Tuy nhiên, nếu không điều chỉnh tham số phù hợp, các mô hình này có thể dễ dàng rơi vào tình trạng quá khớp dữ liệu huấn luyện.

Nhóm boosting nâng cao bao gồm ba mô hình hiện đại là XGBoost, LightGBM và CatBoost, đều được phát triển từ thuật toán Gradient Boosting truyền thống với nhiều cải tiến quan trọng. XGBoost (Extreme Gradient Boosting) tích hợp cơ chế regularization nhằm kiểm soát độ phức tạp của mô hình, đồng thời tối ưu hóa tốc độ huấn luyện và khả năng tổng quát hóa. LightGBM cải tiến thêm bằng cách sử dụng cơ chế Leaf-wise growth thay vì Level-wise như các mô hình truyền thống, giúp rút ngắn đáng kể thời gian huấn luyện và tăng hiệu quả trên tập dữ liệu lớn. CatBoost được thiết kế đặc biệt để xử lý hiệu quả các biến phân loại mà không cần bước mã hóa phức tạp, đồng thời khắc phục vấn đề sai lệch dữ liệu thông qua kỹ thuật ordered boosting. Nhờ khả năng học phi tuyến mạnh mẽ, tốc độ xử lý nhanh và hiệu quả tổng quát hóa cao, các mô hình boosting nâng cao đã trở thành lựa chọn phổ biến trong các bài toán dự đoán giá trị thực, đặc biệt là trong các cuộc thi học máy như Kaggle House Prices Prediction.

Nhóm cuối cùng bao gồm hai mô hình khác là Support Vector Regression (SVR) và K-Neighbors Regressor (KNN). SVR mở rộng nguyên lý của Support Vector Machine cho bài toán hồi quy, bằng cách tìm kiếm siêu phẳng tối ưu sao cho sai số dự đoán của mô hình nằm trong một giới hạn cho phép, đồng thời tối đa hóa biên giữa các điểm dữ liệu. Nhờ sử dụng các hàm kernel phi tuyến, SVR có khả năng mô hình hóa các quan hệ phức tạp giữa các đặc trưng, nhưng hiệu suất tính toán có thể giảm khi kích thước dữ liệu tăng lớn. Trong khi đó, KNN là mô hình đơn giản, dựa trên giả định rằng các điểm dữ liệu có đặc trưng tương tự sẽ có giá trị đầu ra gần nhau. Dự đoán của KNN được thực hiện bằng cách tính trung bình giá trị của các điểm lân cận gần nhất trong không gian đặc trưng. Mặc dù dễ hiểu và không yêu cầu quá trình huấn luyện phức tạp, KNN lại nhạy cảm với nhiễu và phụ thuộc nhiều vào cách lựa chọn số lượng láng giềng cũng như khoảng cách đo.

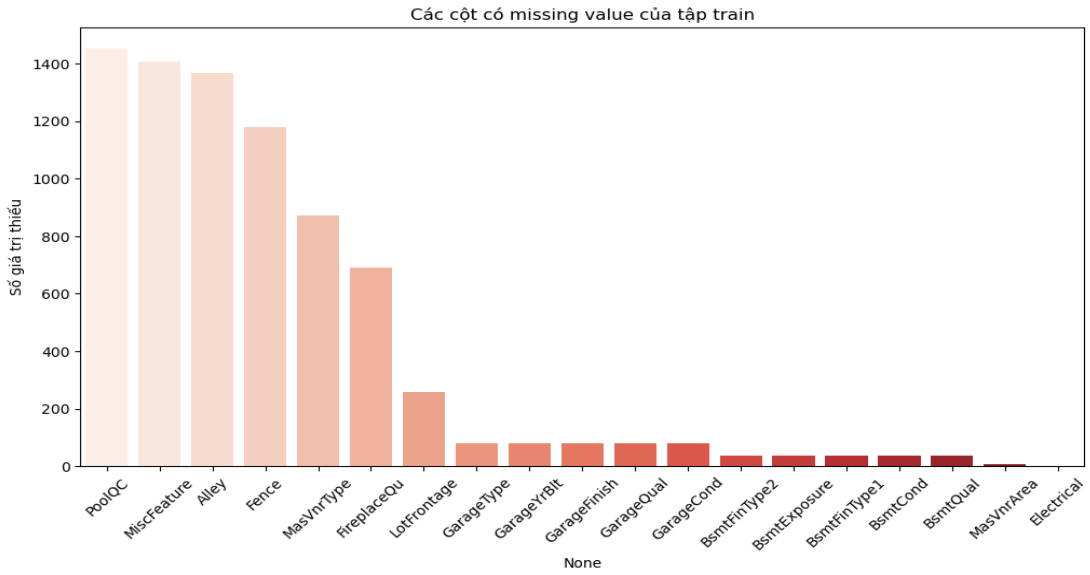
4 . Thí nghiệm và kết quả

4.1 . Phân tích và biến đổi dữ liệu cho quá trình huấn luyện

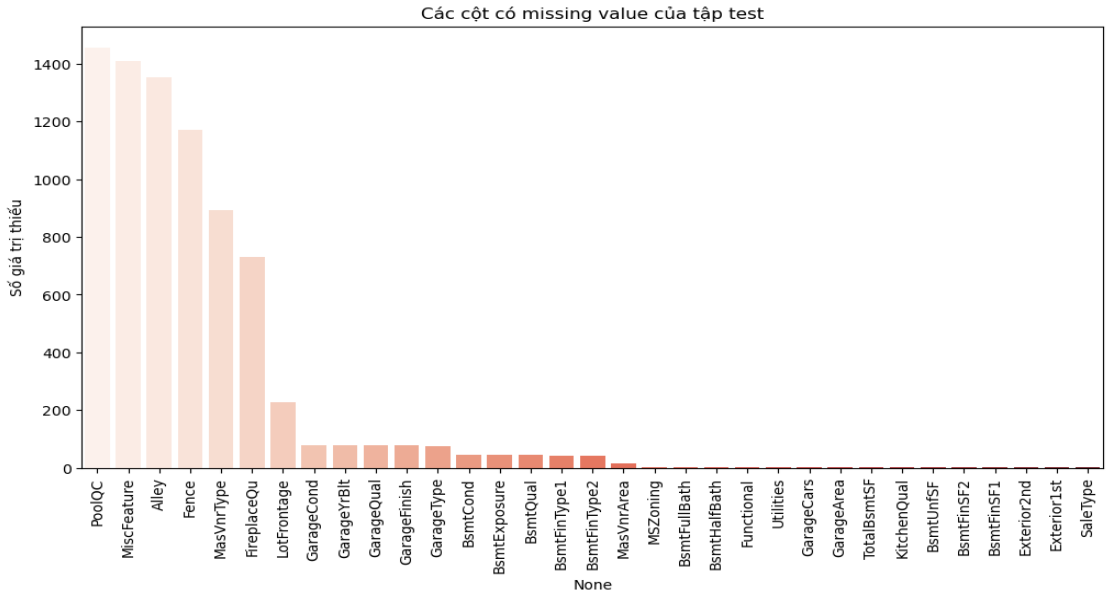
4.1.1 . Xử lý cơ bản các missing value

	Giá trị thiếu	Phần trăm (%)
PoolQC	1453	99.52
MiscFeature	1406	96.30
Alley	1369	93.77
Fence	1179	80.75
MasVnrType	872	59.73
FireplaceQu	690	47.26
LotFrontage	259	17.74
GarageType	81	5.55
GarageYrBlt	81	5.55
GarageFinish	81	5.55
GarageQual	81	5.55
GarageCond	81	5.55
BsmtFinType2	38	2.60
BsmtExposure	38	2.60
BsmtFinType1	37	2.53
BsmtCond	37	2.53
BsmtQual	37	2.53
MasVnrArea	8	0.55
Electrical	1	0.07

Hầu hết các đặc trưng thiếu dữ liệu đa phần đều thuộc kiểu Object. Nên đối với các đặc trưng Object này ta chỉ cần điền vào các vị trí thiếu giá trị None. Còn đối với các giá trị số như : *MasVnrType, LotFrontage, GarageYrBlt hay Electrical* ta điền các giá trị như mode, median. Cụ thể, đối với *MasVnrType* và *GarageYrBlt* ta thêm vào các mẫu thiếu các giá trị 0. Bên cạnh đó, đối với các mẫu thiếu dữ liệu của đặc trưng *LotFrontage* ta cần điền vào vị trí thiếu theo trung vị (median) của từng khu phố.

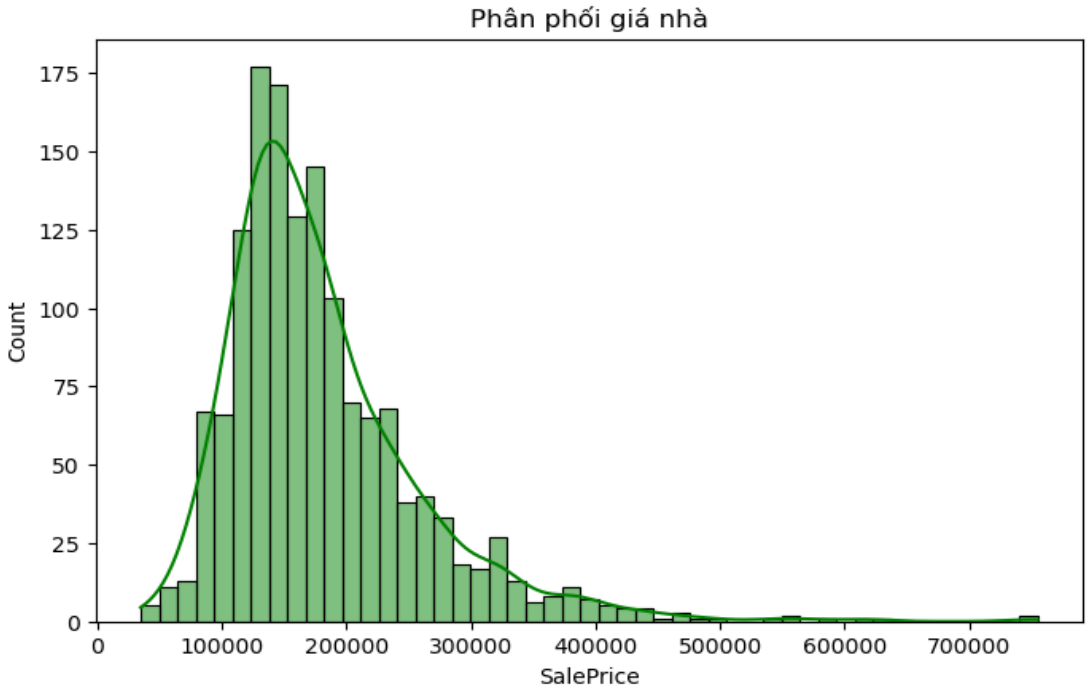


Nhưng đây chỉ là những đặc trưng có dữ liệu bị thiếu trong tập huấn luyện, chúng ta cũng cần phải chú ý đặc biệt đến tập kiểm thử bởi vì trong tập này lại có thêm 1 vài đặc trưng bị thiếu dữ liệu như : *BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF,* ... Các đặc trưng lần này đa số thuộc kiểu số nên chỉ cần thêm dữ liệu theo mode hoặc trung ·vị (median), tùy theo từng trường hợp của các đặc trưng mà có thêm theo từng kiểu cho phù hợp.



4.1.2 . Phân tích đơn biến

Trong phần này, biến đầu tiên và cũng là quan trọng nhất của thí nghiệm mà ta cần quan tâm chính là biến mục tiêu giá nhà (SalePrice). Dựa vào biểu đồ bên dưới ta cũng có thể thấy sự phân phối của SalePrice không tuân theo phân phối chuẩn, mà nó đã lệch phải (right-skewed) rõ rệt với phần lớn giá nhà tập trung trong khoảng **100,000–**

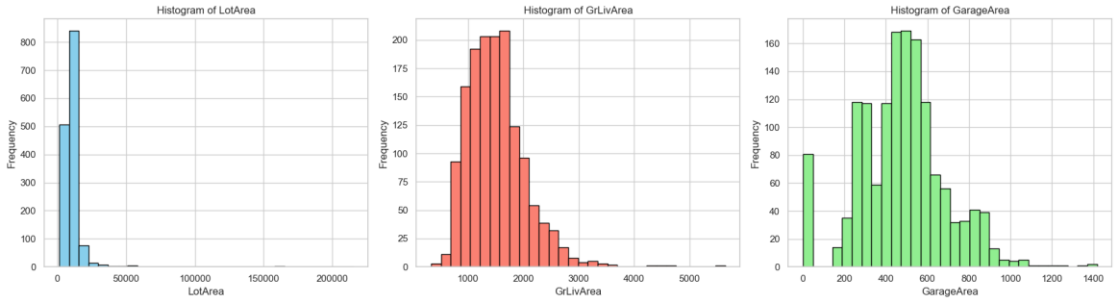


200,000, trong khi các giá trị lớn hơn giảm nhanh và xuất hiện một số **ngoại lệ (outliers)** ở vùng giá cao. Phân phối không chuẩn này có thể gây ảnh hưởng đến hiệu suất của mô hình hồi quy. Do đó, việc **biến đổi logarit (log-transformation)** được xem là cần thiết

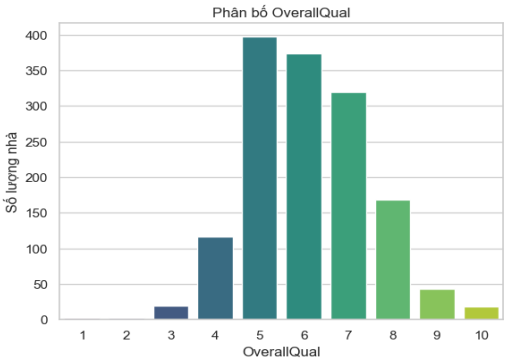
nhằm giảm độ lệch, hạn chế tác động của outliers và giúp mô hình học được mối quan hệ ổn định hơn giữa các đặc trưng và giá trị mục tiêu.

Các biến định lượng như LotArea, GrLivArea, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, GarageArea, YearBuilt, YearRemodAdd, và OverallQual có ảnh hưởng trực tiếp đến giá nhà.

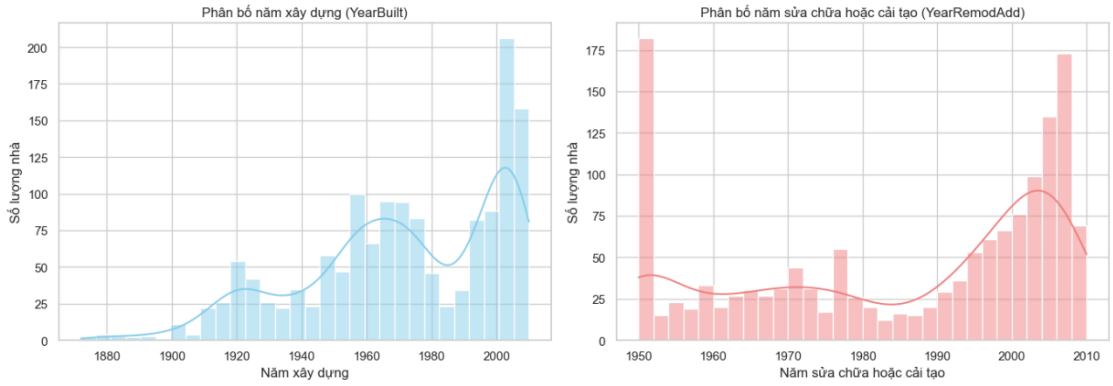
Phân tích histogram cho thấy hầu hết các biến diện tích (LotArea, GrLivArea, GarageArea) đều có phân bố lệch phải tương tự, nghĩa là phần lớn các căn nhà có diện tích vừa phải, trong khi một số ít có diện tích rất lớn. Điều này phù hợp với thực tế của thị trường bất động sản, nơi các căn nhà lớn chiếm tỷ lệ nhỏ nhưng có giá trị cao.



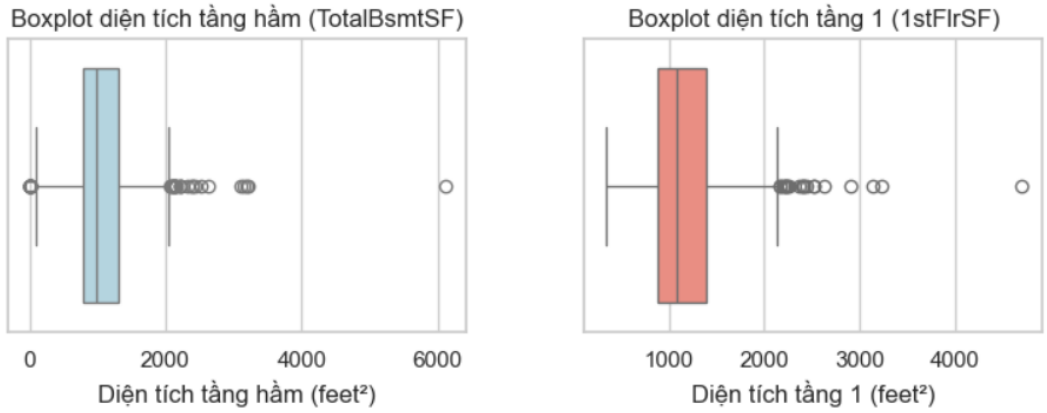
Biến OverallQual (đánh giá tổng thể về chất lượng nhà từ 1–10) thể hiện xu hướng gần như tuyến tính với SalePrice. Phân bố của nó cho thấy phần lớn các căn nhà có chất lượng từ mức 5 đến 7, trong khi rất ít đạt điểm 9 hoặc 10. Đây là một trong những thuộc tính quan trọng nhất vì nó tổng hợp đánh giá cảm quan về chất lượng xây dựng và hoàn thiện nội thất.



Biến YearBuilt có phân bố khá đều trong khoảng từ 1900 đến 2010, nhưng mật độ cao hơn ở giai đoạn sau năm 1970, phản ánh sự phát triển mạnh mẽ của các khu dân cư hiện đại. Tương tự, YearRemodAdd (năm sửa chữa hoặc cải tạo) cho thấy nhiều căn nhà được nâng cấp trong khoảng 1990–2005, thể hiện xu hướng duy trì và cải thiện chất lượng nhà ở.

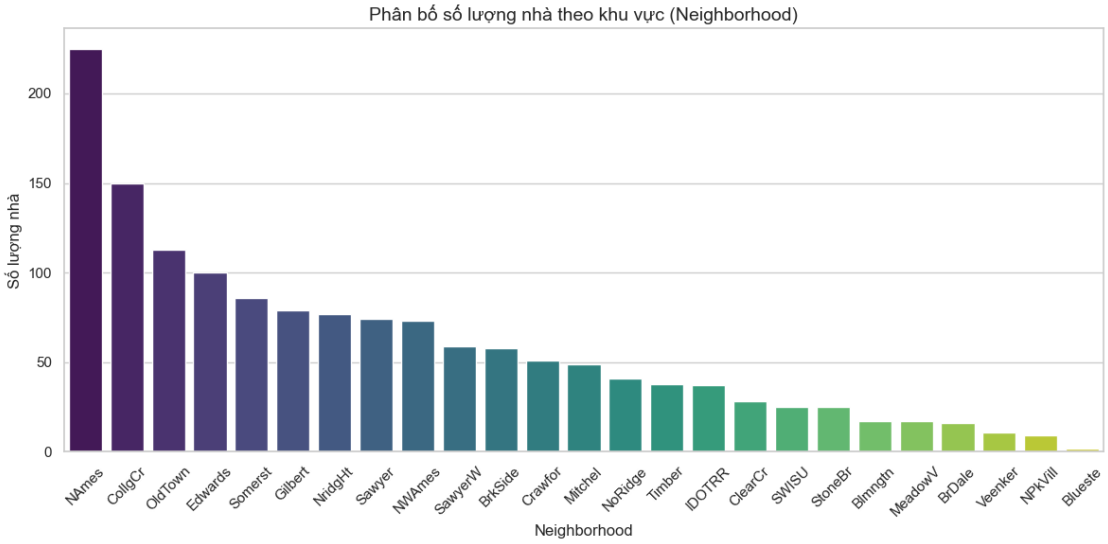


Đối với biến TotalBsmtSF (diện tích tầng hầm) và 1stFlrSF, histogram cho thấy xu hướng tương đồng — phần lớn các căn nhà có diện tích tầng hầm vừa phải (dưới 1000 feet vuông). Boxplot chỉ ra một số giá trị cực đại vượt xa mức trung bình, cho thấy sự tồn tại của outliers cần được xử lý trong giai đoạn tiền xử lý.

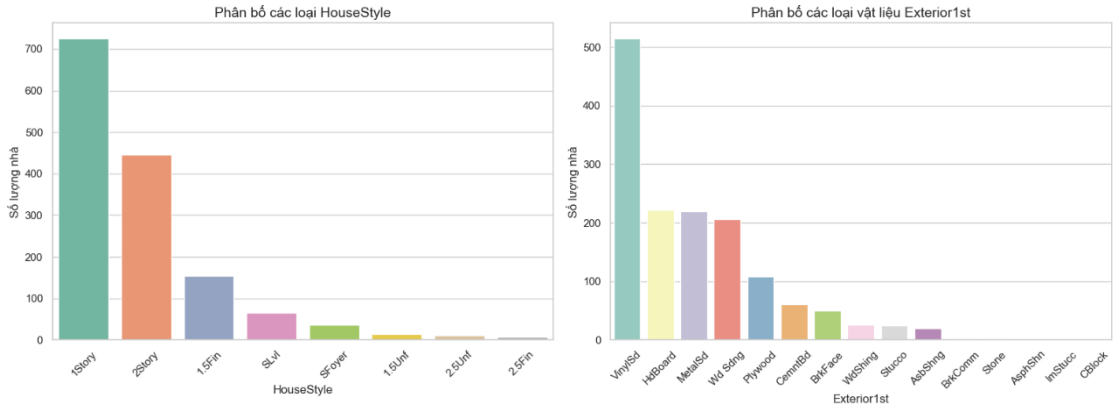


Ngoài các đặc trưng số thì bên cạnh đó vẫn có các đặc trưng phân loại nên chú ý vì sự quan trọng của chúng như Neighborhood, HouseStyle, Exterior1st, RoofStyle, Foundation, GarageType, Heating, SaleCondition, …

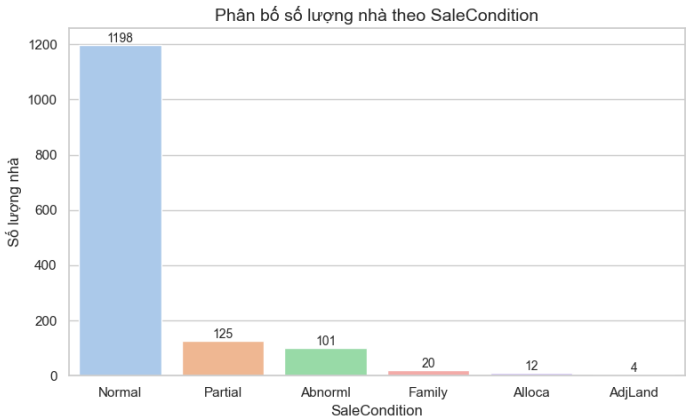
Các biểu đồ cột (countplot) cho thấy sự phân bố không đồng đều giữa các nhóm giá trị. Ví dụ, Neighborhood có hơn 20 khu vực khác nhau, trong đó các khu như *NAmes*, *CollgCr* và *OldTown* chiếm tỷ lệ cao nhất. Điều này cho thấy dữ liệu bị chi phối bởi một số vùng dân cư lớn, còn các khu vực khác chỉ xuất hiện với tần suất thấp.



Biến HouseStyle tập trung chủ yếu ở các loại nhà *1Story*, *2Story* và *1.5Fin*, phản ánh xu hướng phổ biến của các mô hình nhà ở tại Mỹ. Exterior1st cho thấy các vật liệu phổ biến là *VinylSd* (vinyl siding) và *HdBoard*, trong khi *BrkFace* (gạch ốp) xuất hiện ít hơn.



Đối với SaleCondition, đa số các căn nhà được bán trong điều kiện *Normal*, còn các trường hợp *Abnorml* hoặc *Partial* chiếm tỷ lệ rất nhỏ. Điều này cho thấy phần lớn các giao dịch trong dữ liệu diễn ra trong điều kiện bình thường, ít bị tác động bởi yếu tố ép giá hoặc thay đổi bất thường.

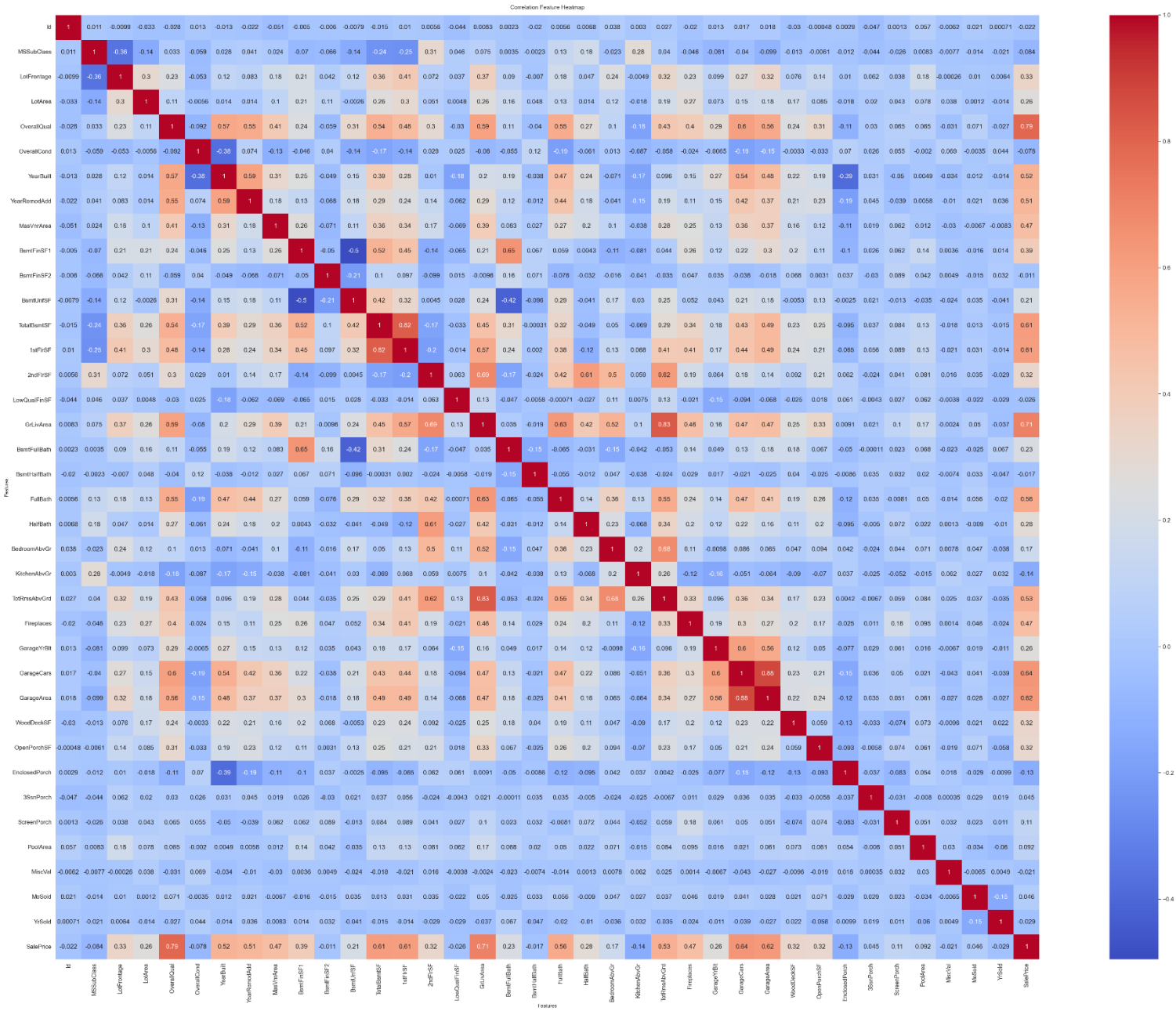


Tổng hợp kết quả phân tích cho thấy các biến định lượng như GrLivArea, OverallQual, GarageArea, TotalBsmtSF và YearBuilt có mối liên hệ chặt chẽ với giá

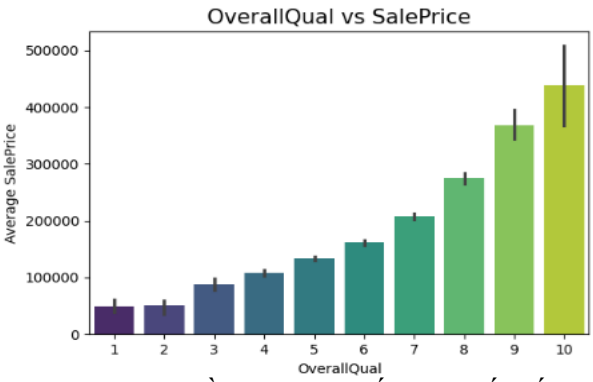
bán. Trong khi đó, các biến phân loại như Neighborhood, HouseStyle và SaleCondition phản ánh bối cảnh vị trí và loại nhà — các yếu tố có ảnh hưởng gián tiếp nhưng quan trọng đến giá trị căn nhà.

Sự hiện diện của nhiều phân bố lệch phải gợi ý rằng việc chuẩn hóa và biến đổi logarit sẽ giúp dữ liệu trở nên cân đối hơn, hỗ trợ mô hình hồi quy hoạt động hiệu quả. Ngoài ra, một số biến chứa giá trị ngoại lai hoặc tần suất hiếm cần được xử lý trong bước phân tích đa biến tiếp theo nhằm giảm nhiễu và cải thiện tính tổng quát của mô hình.

4.1.3 . Phân tích đa biến

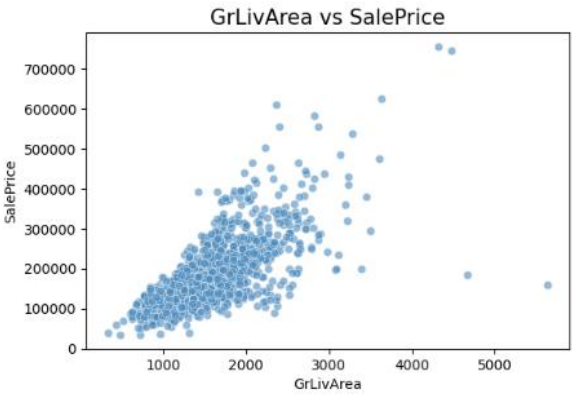


Sau khi thực hiện phân tích đơn biến để hiểu đặc trưng và phân bố của từng biến độc lập, bước tiếp theo là tiến hành phân tích đa biến nhằm khám phá mối quan hệ giữa các biến đầu vào và biến mục tiêu *SalePrice*. Phân tích đa biến giúp xác định các đặc trưng có ảnh hưởng mạnh nhất đến giá nhà, đồng thời phát hiện sự tương quan giữa các thuộc tính có thể gây ra hiện tượng đa cộng tuyến trong mô hình hồi quy. Trong nghiên cứu này, ma trận tương quan (correlation matrix) và biểu đồ heatmap được sử dụng để trực quan hóa mức độ liên hệ giữa các biến, qua đó hỗ trợ quá trình lựa chọn và tối ưu hóa đặc trưng cho các mô hình học máy.



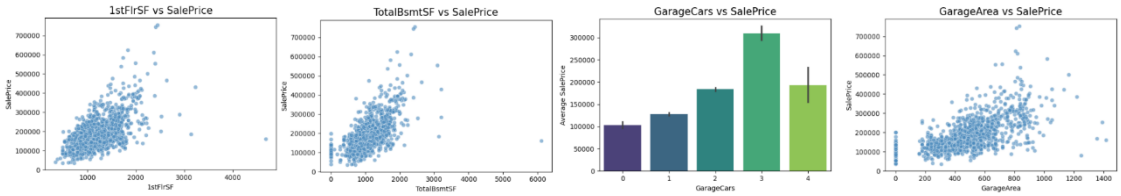
Trước hết, biến *SalePrice* thể hiện mối tương quan dương mạnh với một số đặc trưng quan trọng phản ánh quy mô, chất lượng và giá trị sử dụng của ngôi nhà. Trong đó, *OverallQual* có hệ số tương quan cao nhất (≈ 0.79), chứng tỏ rằng yếu tố *chất lượng tổng thể của vật liệu và mức độ hoàn thiện* đóng vai trò quyết định trong việc hình thành giá trị bất động sản. Các căn nhà được xây dựng bằng vật liệu tốt, có thiết kế nội thất và kiến trúc hoàn thiện, thường có giá bán cao hơn đáng kể so với các căn có chất lượng trung bình hoặc thấp. Điều này cũng phản ánh hành vi tiêu dùng trong thị trường nhà ở, khi người mua sẵn sàng chi trả nhiều hơn cho những căn nhà có chất lượng xây dựng vượt trội, độ bền cao và tính thẩm mỹ tốt.

Tiếp theo, biến *GrLivArea* (diện tích sàn sinh hoạt trên mặt đất) có hệ số tương quan khoảng **0.71**, cho thấy *diện tích sử dụng thực tế* là yếu tố quan trọng thứ hai ảnh hưởng đến giá nhà. Những ngôi nhà có không gian sinh hoạt rộng rãi hơn, đặc biệt ở khu vực tầng chính, thường được định giá cao hơn do mang lại tiện nghi và sự thoải mái lớn hơn cho người sử dụng. Kết quả này cũng phù hợp với các nghiên cứu trước đây trong lĩnh vực định giá bất động sản, khi diện tích luôn là biến đầu vào quan trọng trong các mô hình hồi quy giá nhà.

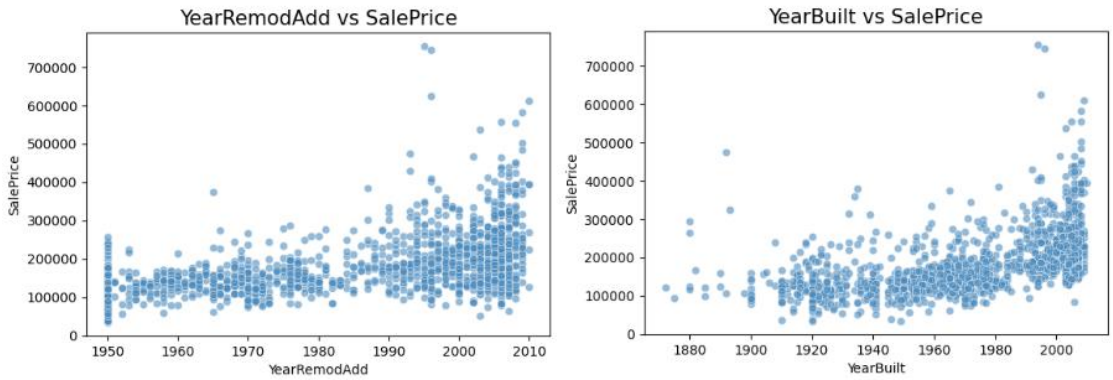


Ngoài ra, nhóm các biến liên quan đến *khu vực gara và diện tích nền móng* cũng cho thấy mối tương quan đáng kể với SalePrice, cụ thể: *GarageCars* (≈ 0.66), *GarageArea* (≈ 0.62), *1stFlrSF* (≈ 0.63) và *TotalBsmtSF* (≈ 0.61). Những hệ số này chỉ ra rằng các đặc trưng về *không gian phụ trợ và tiện ích sử dụng* – chẳng hạn như chỗ đậu xe hoặc diện tích tầng hầm – có ảnh hưởng đáng kể đến giá trị tài sản. Một ngôi nhà có gara rộng rãi hoặc tầng hầm được hoàn thiện tốt thường được người mua đánh giá

cao nhờ tính tiện nghi và khả năng mở rộng công năng sử dụng. Điều này cũng cho thấy các yếu tố không gian phụ không chỉ là đặc điểm kỹ thuật mà còn là *chỉ báo gián tiếp cho mức độ hiện đại và giá trị sử dụng lâu dài của ngôi nhà.*



Cuối cùng, các biến *YearBuilt* và *YearRemodAdd* có hệ số tương quan trung bình (≈ 0.53–0.55) nhưng vẫn mang ý nghĩa quan trọng trong định giá. Giá trị tương quan dương này phản ánh thực tế rằng *những căn nhà được xây mới hoặc cải tạo gần đây thường có giá bán cao hơn*, do chi phí bảo trì thấp hơn, tính thẩm mỹ và công năng sử dụng phù hợp hơn với nhu cầu hiện đại. Ngoài ra, năm xây dựng hoặc cải tạo cũng có thể được xem là *thước đo gián tiếp cho tuổi thọ công trình*, cho phép mô hình đánh giá mức độ khấu hao và giá trị còn lại của bất động sản.



Tiếp theo, biểu đồ heatmap cũng chỉ ra nhiều mối tương quan mạnh giữa các đặc trưng cấu trúc. Chẳng hạn, *GarageCar***s** và *GarageArea* có hệ số tương quan rất cao (r ≈ 0.88), điều này hoàn toàn hợp lý vì diện tích gara lớn thường đồng nghĩa với khả năng chứa nhiều xe hơn. Tương tự, *TotalBsmtSF* và *1stFlrSF* (r ≈ 0.82) hay *GrLivArea và TotRmsAbvGrd* (r ≈ 0.83) cho thấy rằng những căn nhà có nhiều phòng hoặc diện tích sử dụng lớn thường có quy mô tầng một và tầng hầm rộng rãi hơn. Những mối tương quan này gợi ý khả năng xuất hiện hiện tượng *đa cộng tuyến (multicollinearity)* trong mô hình dự đoán, cần được xem xét kỹ lưỡng trong quá trình lựa chọn đặc trưng.

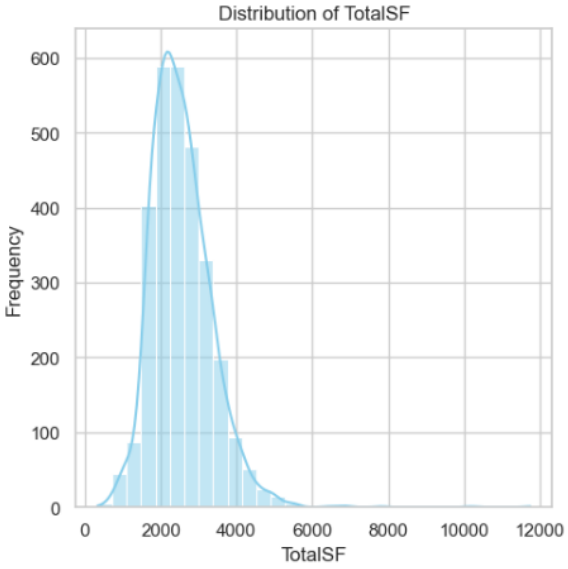
Ngược lại, một số biến thể hiện mối tương quan yếu hoặc gần như không đáng kể với SalePrice. Các thuộc tính như *PoolArea, MiscVal* hay *LowQualFinSF* có hệ số tương quan thấp, cho thấy chúng ít có giá trị dự báo đối với giá nhà. Hầu hết các đặc trưng mang tính kích thước hoặc chất lượng đều có mối tương quan dương, trong khi các mối tương quan âm gần như không đáng kể.

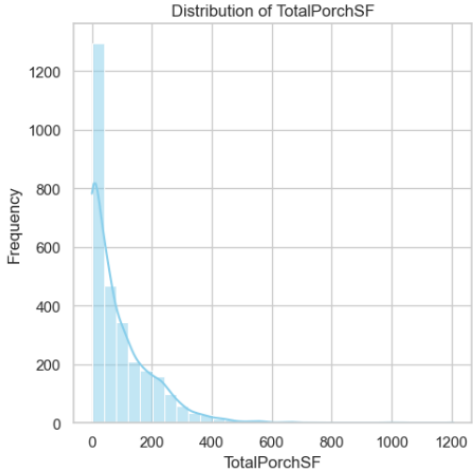
Tổng thể, biểu đồ heatmap mang lại cái nhìn tổng quan về mối liên hệ giữa các biến trong tập dữ liệu. Kết quả cho thấy *chất lượng tổng thể, diện tích không gian sống và năm xây dựng hoặc cải tạo* là những yếu tố chủ đạo ảnh hưởng đến giá bán nhà. Đồng thời, việc tồn tại các mối tương quan mạnh giữa các biến kích thước cũng cảnh báo cần thực hiện bước *lựa chọn và xử lý đặc trưng hợp lý* nhằm tránh hiện tượng trùng lặp thông tin, giúp mô hình dự đoán hoạt động ổn định và chính xác hơn.

4.1.4 . Feature Engineering

Chỉ với các đặc trưng đầu vào của dữ liệu thì chắc chắn sẽ không đủ để các mô hình có thể học và dự đoán được tốt nhất. Chính vì đó, ngoài việc xử lý các missing value, ngoại lệ, … ta còn cần phải tạo thêm các đặc trưng mới từ nhưng đặc trưng có sẵn dựa trên mối quan hệ giữa các đặc trưng hiện có. Việc tạo ra các đặc trưng mới (feature engineering) giúp mô hình khai thác được nhiều thông tin hơn từ dữ liệu gốc, từ đó cải thiện khả năng học và dự đoán.

Trong thí nghiệm này, chúng tôi đã bổ sung thêm 10 đặc trưng mới để có thể giúp cho mô hình có thể cải thiện được hiệu suất tốt hơn. Thứ nhất, đặc trưng *TotalSF* được tạo ra bằng cách cộng tổng diện tích của tầng hầm (*TotalBsmtSF*), tầng một (*1stFlrSF*) và tầng hai (*2ndFlrSF*). Đây là một đặc trưng tổng hợp có ý nghĩa quan trọng, phản ánh *tổng diện tích sử dụng của toàn bộ căn nhà*. Trong thực tế, diện tích là một trong những yếu tố có ảnh hưởng lớn nhất đến giá trị của bất động sản. Việc gộp diện tích các tầng lại giúp mô hình hiểu được quy mô tổng thể thay vì xem xét từng tầng riêng lẻ, vốn có thể khiến mô hình bỏ sót mối quan hệ tuyến tính giữa diện tích và giá nhà.

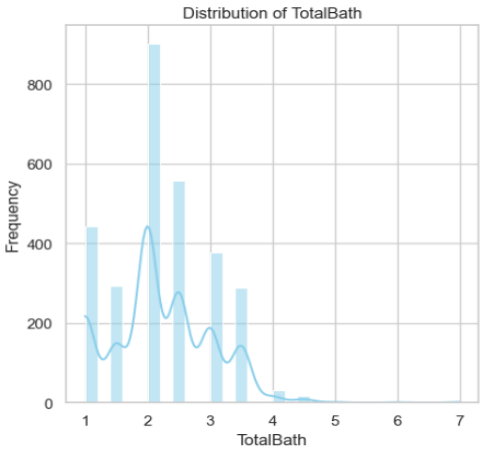


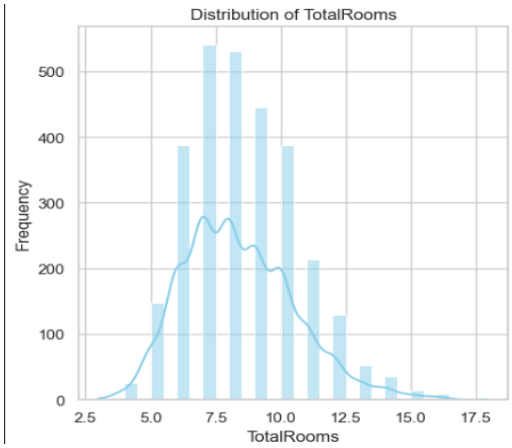


Tiếp theo, đặc trưng *TotalPorchSF* biểu thị tổng diện tích các khu vực hiên, ban công và không gian mở bên ngoài, được tính từ các cột *OpenPorchSF*, *EnclosedPorch*, *3SsnPorch* và *ScreenPorch*. Đặc trưng này thể hiện mức độ *tiện nghi và không gian sinh hoạt ngoài trời* của căn nhà. Những ngôi nhà có hiên rộng hoặc nhiều không gian mở thường được đánh giá cao hơn do mang lại trải nghiệm sống

thoải mái và gần gũi với thiên nhiên hơn, đặc biệt trong các khu dân cư cao cấp.

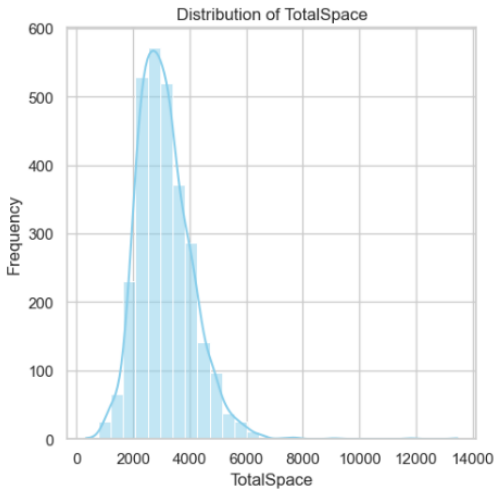
Đặc trưng *TotalBath* được xây dựng dựa trên tổng số phòng tắm đầy đủ và bán phần, với quy ước rằng một phòng tắm bán phần chỉ được tính 0.5 đơn vị. Ngoài ra, đặc trưng này còn bao gồm cả các phòng tắm ở tầng hầm (*BsmtFullBath*, *BsmtHalfBath*). Việc quy đổi như vậy giúp đặc trưng phản *ánh mức độ tiện nghi và khả năng đáp ứng sinh hoạt của căn nhà,* vì số lượng phòng tắm thường tỉ lệ thuận với số lượng phòng ngủ và số thành viên có thể sinh sống.





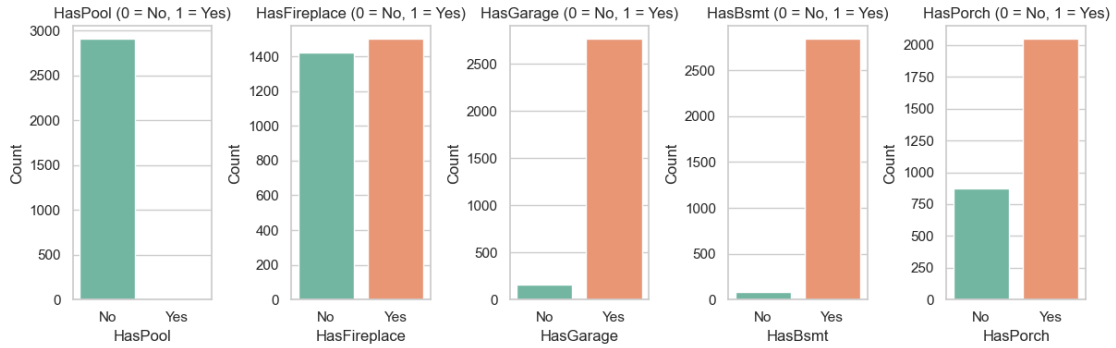
Đặc trưng *TotalRooms* được tạo ra bằng cách cộng tổng số phòng ở trên mặt đất (*TotRmsAbvGrd*) với số lượng phòng tắm đầy đủ và bán phần. Mặc dù dữ liệu gốc đã có thông tin về số phòng ở, việc bổ sung thêm các phòng tắm vào tổng số phòng giúp đặc trưng này phản *ánh tổng số không gian sử dụng thực tế,* mô tả rõ ràng hơn quy mô sinh hoạt của căn nhà.

Tiếp theo, đặc trưng *TotalSpace* được tính bằng tổng diện tích sàn (*TotalSF*), diện tích gara (*GarageArea*) và tổng diện tích hiên (*TotalPorchSF*). Đây là đặc trưng tổng hợp có khả năng mô tả *toàn bộ không gian sử dụng của ngôi nhà*, bao gồm cả các khu vực chức năng phụ trợ như gara hoặc ban công. Đặc trưng này có ý nghĩa đặc biệt trong việc phân biệt những ngôi nhà có cùng diện tích ở nhưng khác nhau về các không gian phụ, từ đó phản ánh giá trị thực tế sát hơn.



Bên cạnh các đặc trưng định lượng, nghiên cứu cũng bổ sung thêm một số đặc trưng nhị phân nhằm thể hiện sự hiện diện của các tiện ích cụ thể. Các đặc trưng này bao gồm *HasPool, HasFireplace, HasGarage, HasBsmt* và *HasPorch*, tương ứng với việc căn nhà có hồ bơi, lò sưởi, gara, tầng hầm hoặc hiên. Mỗi đặc trưng được gán giá trị 1 nếu tồn tại tiện ích đó và 0 nếu không. Việc chuyển đổi này giúp mô hình học dễ

dàng hơn, đặc biệt khi các biến diện tích tương ứng (*PoolArea*, *GarageArea*, v.v.*) có phân phối rất lệch với phần lớn giá trị bằng 0. Chẳng hạn, đặc trưng *HasPool* giúp mô hình nhanh chóng phân biệt các căn nhà có hồ bơi — một yếu tố thường gắn liền với những bất động sản cao cấp. Tương tự, *HasFireplace* thể hiện mức độ sang trọng và tiện nghi, trong khi *HasGarage* và *HasBsmt* lại phản ánh tính thực dụng và giá trị sử dụng của ngôi nhà.



Nhìn chung, các đặc trưng mới này đóng vai trò quan trọng trong việc làm giàu dữ liệu và giúp mô hình nắm bắt được *mối quan hệ phức tạp giữa quy mô, tiện nghi và giá trị bất động sản*. Bằng cách tổng hợp và biểu diễn lại thông tin từ nhiều thuộc tính gốc, mô hình không chỉ học được tốt hơn về cấu trúc của ngôi nhà mà còn hiểu sâu hơn về những yếu tố thực tế ảnh hưởng đến giá bán, từ đó cải thiện đáng kể hiệu suất dự đoán.

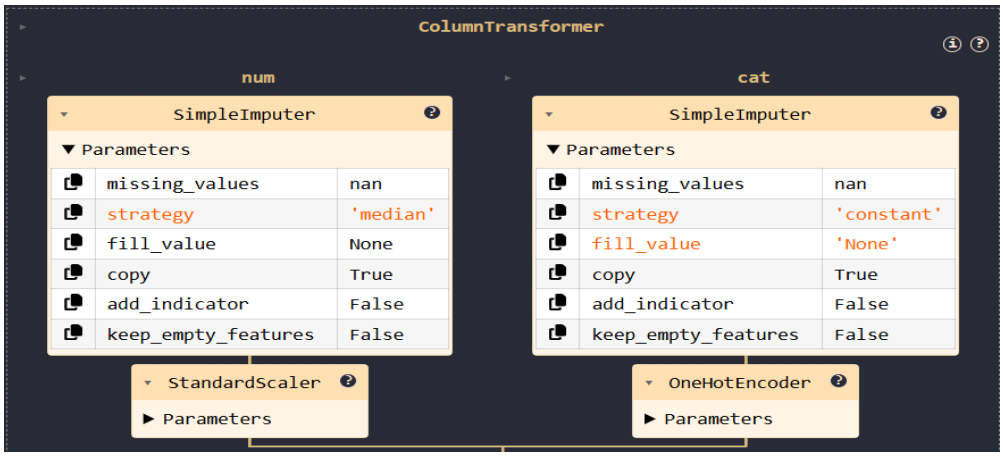
4.1.5 . Chuẩn hóa và mã hóa dữ liệu

Ngoài việc xây dựng thêm một số đặc trưng mới, chuẩn hóa dữ liệu cũng là một khâu vô cùng quan trọng để có thể giúp cho mô hình học tốt hơn. Có khá nhiều cách để chuẩn hóa dữ liệu như *MinMaxScaler, StandardScaler, RobustScaling, MaxAbsScaler, Normalizer*, … Ở nghiên cứu này chúng ta sẽ tiến hành chuẩn hóa dữ liệu theo phương pháp *StandardScaler*, đây là kỹ thuật được sử dụng rộng rãi nhất hiện nay. Phương pháp này thực hiện chuẩn hóa theo phân phối chuẩn, đưa mỗi đặc trưng của nó về dạng trung bình bằng 0 và độ lệch bằng 1, theo công thức : $x' = \frac{x - \mu}{\sigma}$. Trong đó: x là giá trị gốc của đặc trưng, μ là giá trị trung bình của đặc trưng đó, σ là độ lệch chuẩn tương ứng. Các đặc trưng cần chuẩn hóa dữ liệu là các đặc trưng có kiểu dữ liệu số.

Ngoài việc chuẩn hóa ta cũng cần mã hóa các dữ liệu để mô hình đạt được hiệu suất tối ưu hơn. Nếu chuẩn hóa dữ liệu là xử lý các đặc trưng dạng số thì mã hóa dữ liệu là phân loại các đặc trưng không phải dạng số để mô hình có thể hiểu và xử lý được. Ở thí nghiệm này các đặc trưng phân loại sẽ được mã hóa bằng kỹ thuật One-Hot Encoding. Cách mã hóa này giúp loại bỏ mối quan hệ giả giữa các giá trị phân loại mà *Label Encoding* có thể vô tình tạo ra, đồng thời đảm bảo rằng mô hình học máy không hiểu sai các giá trị này như các đại lượng có thứ tự hay khoảng cách. Việc áp *dụng One-*

Hot Encoding giúp mô hình học sâu và các thuật toán như *XGBoost, CatBoost,* hay *Random Forest* có thể khai thác hiệu quả hơn mối quan hệ giữa các loại đặc trưng khác nhau, từ đó *nâng cao độ chính xác trong dự đoán giá nhà.*

Nhưng việc xử lý chuẩn hóa hoặc mã hóa đơn lẻ từng đặc trưng sẽ gây rối hơn dẫn đến việc sẽ sai lầm. Thay vào đó, ta sử dụng kỹ thuật *Pipeline* để có thể *tự động hóa và thống nhất toàn bộ quy trình xử lý dữ liệu.* Pipeline cho phép gộp tất cả các bước như chuẩn hóa, mã hóa, và huấn luyện mô hình vào cùng một cấu trúc liền mạch. Khi đó, dữ liệu sẽ được truyền qua từng giai đoạn xử lý theo đúng thứ tự định sẵn, đảm bảo tính đồng nhất giữa quá trình huấn luyện và dự đoán. Việc này không chỉ giúp giảm thiểu rủi ro sai lệch do xử lý thủ công mà còn giúp dễ dàng tái sử dụng hoặc mở rộng quy trình cho các thí nghiệm khác.



Tham số *missing_values* xác định loại giá trị nào được xem là thiếu, thường là *nan* trong dữ liệu số. Tiếp theo, tham số *strategy* quy định cách thay thế các giá trị thiếu, ví dụ như dùng trung vị (*'median'*) cho dữ liệu số để giảm ảnh hưởng của ngoại lệ, hoặc dùng một giá trị cố định (*'constant'*) cho dữ liệu phân loại. Khi sử dụng chiến lược *'constant'*, tham số *fill_value* sẽ chỉ định giá trị cụ thể để thay thế, chẳng hạn *'None'* hoặc *'missing'*. Tham số *copy* đảm bảo rằng dữ liệu gốc không bị thay đổi bằng cách tạo bản sao trước khi biến đổi. Ngoài ra, *add_indicator* cho phép thêm một cột phụ để đánh dấu các vị trí từng bị thiếu, giúp mô hình học được thông tin này nếu cần. Cuối cùng, *keep_empty_features* kiểm soát việc giữ lại các cột hoàn toàn trống sau khi biến đổi, thường được đặt là *False* để loại bỏ những cột không hữu ích.

4.2 . Thiết lập thí nghiệm

Quy trình huấn luyện: Trước khi bắt đầu huấn luyện cần chia dữ liệu của tập train theo phương pháp K-Fold Cross Validation (K = 5) nhằm đảm bảo tính *khách quan* và giảm thiểu hiện tượng *quá khớp*. Phương pháp này cho phép mô hình được huấn luyện và kiểm thử luân phiên trên nhiều tập con khác nhau, từ đó cung cấp đánh giá *ổn định* và toàn diện hơn về *hiệu năng* mô hình.

Trước khi huấn luyện, toàn bộ dữ liệu được xử lý theo các bước đã trình bày, bao gồm *xử lý giá trị khuyết*, chuẩn hóa bằng *StandardScaler* và mã hóa các biến phân loại bằng *One-Hot Encoding*. Ngoài các đặc trưng dữ liệu cung cấp ta cũng cần bổ sung thêm các đặc trưng khác như : TotalSF , TotalPorchSF , TotalBath , TotalRooms, … nhằm tăng khả năng mô hình hóa mối *quan hệ phi tuyến* giữa các yếu tố.

Trong quá trình huấn luyện, các mô hình Random Forest, LinearRegression, Gradient Boosting, XGBoost, CatBoost, … được tối ưu các tham số thông qua Grid Search kết hợp với K-Fold Cross Validation để lựa chọn bộ siêu tham số tối ưu. Sau đó dùng chính bộ siêu tham số tối ưu đó để huấn luyện lại tất cả dữ liệu một lần nữa trước khi thử nghiệm. Sau khi huấn luyện với bộ *siêu tham số tối ưu*, các mô hình được so sánh dựa trên *chỉ số RMSE (Root Mean Squared Error)* – thước đo sai số phổ biến trong các bài toán hồi quy, thể hiện mức độ chênh lệch trung bình giữa giá trị dự đoán và giá trị thực tế. Ngoài ra, để đảm bảo tính ổn định, kết quả RMSE được lấy trung bình từ các lần chia K-Fold.

Cuối cùng, mô hình có *giá trị RMSE thấp nhất* trên tập kiểm tra sẽ được lựa chọn là *mô hình tối ưu*, sau đó có thể được sử dụng để *dự đoán giá nhà (SalePrice)* trên tập dữ liệu mới hoặc trong thực tế.

4.3 . Kết quả

Sau khi thực hiện các thao tác phân tích, biến đổi, và huấn luyện dữ liệu qua nhiều lượt chung tôi đã thu nhận được các kết quả khả quan thể hiện ở cả mặt định lượng và định tính. Các mô hình đã được sử dụng để huấn luyện bao gồm *Linear Regression, Lasso Regression, Ridge Regression, Random Forest Regressor và XGBoost Regressor*, nhằm so sánh hiệu suất giữa các phương pháp hồi quy tuyến tính và phi tuyến tính trong bài toán dự đoán giá nhà. Với lần đầu tiên thí nghiệm bằng các baseline cùng với việc chỉ hoàn thiện các giá trị thiếu của dữ liệu ta có được kết quả khá khả quan như bảng sau.

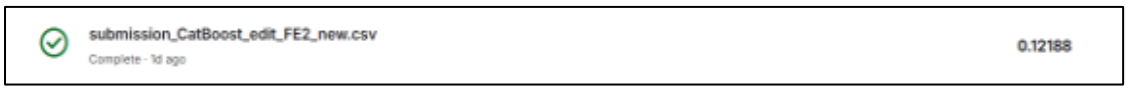
	Model	**RMSE_mean**	**RMSE_std**
1	Random Forest	0.145145	0.019519
2	Gradient Boosting	0.132391	0.021074
3	XGBoost	0.128737	0.017623
4	LightGBM	0.133515	0.109214
5	**CatBoost**	**0.125452**	**0.017374**
6	SVR	0.147832	0.011020

7	K-Neighbors	0.178529	0.035055
8	Linear Regreesion	0.171952	0.013403

Kết quả cho thấy các mô hình boosting như XGBoost, LightGBM và đặc biệt là **CatBoost** đạt hiệu suất cao nhất, với RMSE_mean thấp hơn đáng kể so với các mô hình tuyến tính. Trong đó, **CatBoost** cho kết quả tốt nhất (RMSE_mean = 0.125), thể hiện khả năng học phi tuyến mạnh mẽ và ổn định. Ngược lại, các mô hình tuyến tính như **Linear Regression** cho sai số cao hơn (RMSE ≈ 0.17), chứng tỏ hạn chế trong việc mô hình hóa mối quan hệ phức tạp giữa các đặc trưng. Nhìn chung, nhóm **boosting nâng cao** cho thấy hiệu quả vượt trội và ổn định nhất trong bài toán dự đoán giá nhà.

Với lần thí nghiệm thứ hai, chúng tôi tiến hành huấn luyện lại các mô hình trên tập dữ liệu đã được bổ sung các đặc trưng mới nhằm nâng cao hiệu suất dự đoán. Các biến tương tác phản ánh mối quan hệ giữa diện tích, chất lượng và năm xây dựng. Việc mở rộng tập đặc trưng giúp mô hình khai thác sâu hơn các mối quan hệ phi tuyến giữa các yếu tố cấu thành giá nhà.

	Model	**RMSE_mean**	**RMSE_std**
1	Random Forest	0.141537	0.018187
2	Gradient Boosting	0.131579	0.019280
3	XGBoost	0.129877	0.015487
4	LightGBM	0.134341	0.020826
5	**CatBoost**	**0.125091**	**0.016224**
6	SVR	0.148838	0.011515
7	K-Neighbors	0.171239	0.016183
8	Linear Regreesion	0.198159	0.063407

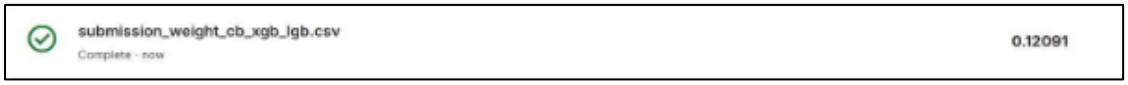


Kết quả ở lần thí nghiệm thứ hai cho thấy hiệu suất của các mô hình nhìn chung được cải thiện, đặc biệt là ở nhóm tree-based và boosting. Các mô hình CatBoost và XGBoost tiếp tục giữ vị trí dẫn đầu với RMSE_mean lần lượt là 0.125 và 0.129, thể hiện khả năng học sâu và ổn định hơn khi được cung cấp các đặc trưng giàu thông tin hơn.

Gradient Boosting và Random Forest cũng cho kết quả tốt, chứng tỏ rằng việc mở rộng tập đặc trưng giúp các mô hình này mô hình hóa tốt hơn mối quan hệ phi tuyến giữa các biến. Ngược lại, các mô hình tuyến tính như Linear Regression cho thấy hiệu suất giảm đáng kể (RMSE_mean ≈ 0.198), cho thấy chúng không phù hợp khi dữ liệu chứa nhiều tương tác phức tạp. Nhìn chung, bộ đặc trưng mở rộng đã giúp các mô hình phi tuyến thể hiện rõ ưu thế, đồng thời khẳng định vai trò quan trọng của kỹ thuật tạo đặc trưng trong việc nâng cao chất lượng dự đoán giá nhà.

Với lần thí nghiệm cuối cùng, để đạt hiệu suất dự đoán tốt hơn, chúng tôi áp dụng phương pháp kết hợp mô hình (ensemble stacking), trong đó các mô hình mạnh nhất ở các lần thí nghiệm trước như CatBoost, XGBoost, Gradient Boosting và Random Forest được sử dụng làm mô hình cơ sở (base models), trong khi Linear Regression đóng vai trò mô hình tổng hợp (meta-model) để kết hợp kết quả dự đoán từ các mô hình con. Cách tiếp cận này giúp tận dụng được ưu điểm riêng của từng mô hình, giảm thiểu sai số ngẫu nhiên và tăng khả năng khái quát hóa trên tập dữ liệu kiểm thử.

	Model	RMSE_mean	RMSE_std
1	Stacking_cb+rf+gb	0.125496	0.018625
2	Stacking_cb+lgb+knn+svr+rf+gb	0.126202	0.018574
3	Stacking_rf+svr+knn	0.398782	0.025132



Từ kết quả thí nghiệm, có thể thấy rằng phương pháp kết hợp mô hình *(ensemble stacking*) mang lại hiệu suất dự đoán vượt trội so với việc sử dụng từng mô hình đơn lẻ. Trong các tổ hợp được thử nghiệm, tổ hợp *Stacking_cb+lgb+knn+svr+rf+gb* đạt giá trị RMSE trung bình thấp nhất (0.1262) và độ lệch chuẩn nhỏ (0.01857), cho thấy mô hình vừa chính xác vừa ổn định trên các lần thử nghiệm khác nhau. Kết quả này chứng minh rằng việc kết hợp nhiều mô hình mạnh với Linear Regression làm meta-model giúp tận dụng ưu điểm riêng của từng mô hình cơ sở, giảm thiểu sai số ngẫu nhiên và tăng khả năng khái quát hóa trên dữ liệu kiểm thử. Do đó, phương pháp ensemble stacking được xác nhận là hướng tiếp cận hiệu quả nhất cho bài toán dự đoán giá nhà, đặc biệt trong bối cảnh dữ liệu có nhiều đặc trưng phức tạp và phân phối không đồng nhất.

5. Kết luận

Qua quá trình phân tích và thử nghiệm, nghiên cứu cho thấy việc áp dụng các mô hình học máy vào bài toán dự đoán giá bất động sản mang lại hiệu quả cao. Đặc biệt, mô hình *CatBoost* vượt trội hơn các mô hình khác như Random Forest, Gradient

Boosting, XGBoost hay Linear Regression về độ chính xác (RMSE thấp nhất) và khả năng khái quát hóa.

Kết quả này khẳng định *tính ưu việt của các mô hình ensemble hiện đại,* nhất là CatBoost, trong việc xử lý dữ liệu dạng bảng có nhiều đặc trưng phi tuyến và biến hạng mục.

Về mặt thực tiễn, mô hình được xây dựng có thể được ứng dụng như *một công cụ hỗ trợ định giá bất động sản tự động,* giúp tiết kiệm thời gian, giảm thiểu sai lệch chủ quan và tăng độ tin cậy trong quá trình ra quyết định. Trong tương lai, hướng phát triển có thể mở rộng bằng cách *tích hợp thêm dữ liệu không gian (vị trí, tiện ích xung quanh) hoặc ứng dụng các mô hình deep learning cho dữ liệu phi cấu trúc* như hình ảnh hoặc văn bản mô tả nhà ở.

Phụ lục:

Maloku, F., Maloku, B., & Agarwal, A. (2024). *House Price Prediction Using Machine Learning and Artificial Intelligence*. *Journal of Artificial Intelligence & Cloud Computing*, 3(4), 1-10. DOI:10.47363/JAICC/2024(3)357. (ResearchGate)

Foryś, I. (2022). *Machine learning in house price analysis: regression and other methods*. *Procedia Computer Science*, or similar. (Article addressing automatic house price determination using multiple regression models and ML). (ScienceDirect)

Truong, Q. (2020). *Housing Price Prediction via Improved Machine Learning*. *Procedia (?) or ScienceDirect article*. (ScienceDirect)

Yalgudkar, S. S. (2022?). *A Literature Survey on Housing Price Prediction*. *Journal of Computer Science & Computational Mathematics*, hoặc tương đương. (jcscm.net)

Ju, X., Chakma, V., Amin, M., & Chakma, J. A. (2025). *What Drives House Prices? A Linear Regression Approach to Size, Condition, and Features*. *Journal of Artificial Intelligence and Data Mining (JAIDM)*, Vol. 13, No. 1, 41-51. DOI:10.22044/jadm.2025.15529.2668. (jad.shahroodut.ac.ir)

Çetin, A. İ. (2025). *Evaluating Machine Learning Models for House Price Prediction with Different Sampling Techniques*. *International Journal of Computational Engineering & Science (IJCESEN)*, Vol. 11 No. 2. DOI:10.22399/ijcesen.2870. (Ijcesen)