

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



**MACHINE LEARNING**  
**(CO3061)**

---

**BÁO CÁO BÀI TẬP LỚN**  
**DỰ ĐOÁN GIÁ NHÀ**  
**SỬ DỤNG RANDOM FOREST VÀ GRADIENT BOOSTING**

---

**Giảng viên hướng dẫn:** Võ Thanh Hùng.

**Sinh viên thực hiện:** Phan Đình Tuấn Anh - 2210118  
Chu Minh Tâm - 2213009  
Nguyễn Trọng Tài - 2212995  
Lưu Chí Lập - 2211830

Thành phố Hồ Chí Minh, 04/2025



# Mục lục

<b>Mục lục</b>	<b>1</b>
<b>Danh Sách thành viên</b>	<b>3</b>
<b>Lời mở đầu</b>	<b>3</b>
<b>1 Cơ sở lý thuyết</b>	<b>5</b>
1.1 Mô tả bài toán	5
1.2 Random Forest	5
1.3 Gradient Boosting	6
1.4 Các chỉ số đánh giá mô hình	6
<b>2 Dữ liệu sử dụng</b>	<b>8</b>
2.1 Mô tả bộ dữ liệu	8
2.2 Phân tích dữ liệu khám phá (EDA)	9
2.2.1 Phân phối của biến mục tiêu (SalePrice)	9
2.2.2 Mối quan hệ giữa SalePrice và các đặc trưng	10
2.2.2.a LotArea vs. SalePrice	10
2.2.2.b SalePrice và Neighborhood	11
2.2.2.c SalePrice và ExterQual	12
2.2.2.d SalePrice và các đặc trưng số	13
2.2.2.e Tương quan giữa các đặc trưng	14
2.2.2.f Đặc trưng được tạo thêm (Feature Engineering)	15
2.3 Tiền xử lý dữ liệu	15
2.3.1 Xử lý dữ liệu bị thiếu	15
2.3.2 Mã hóa biến phân loại	15
2.3.3 Chuẩn hóa đặc trưng	16
2.4 Kết quả sau xử lý	16
<b>3 Xây dựng mô hình</b>	<b>16</b>
3.1 Mô hình Random Forest	16
3.2 Mô hình Gradient Boosting	16
3.3 Tối ưu hóa tham số bằng Grid Search	16
3.4 Lựa chọn đặc trưng quan trọng	17
<b>4 Đánh giá và so sánh kết quả</b>	<b>18</b>
4.1 Hiệu suất trên tập huấn luyện	18
4.2 Đánh giá bằng Cross-Validation	18
4.3 Biểu đồ minh họa	19
4.3.1 Phân phối sai số (Residuals Distribution)	19
4.3.2 Residuals vs. Fitted Values	20
4.3.3 Learning Curves	20
4.3.4 SHAP Feature Importance	21
<b>5 Giao diện người dùng</b>	<b>22</b>
5.0.1 Nhập dữ liệu đầu vào	22



<b>6</b>	<b>Kết luận và hướng phát triển</b>	<b>23</b>
6.1	Kết luận . . . . .	23
<b>7</b>	<b>Mã nguồn</b>	<b>23</b>
<b>8</b>	<b>Tài liệu tham khảo</b>	<b>23</b>
	<b>Tài liệu</b>	<b>23</b>



## Danh sách thành viên

STT	Họ và tên	MSSV	Nhiệm vụ	Hoàn thành
1	Phan Đình Tuấn Anh	2210118	Random Forest	100%
2	Nguyễn Trọng Tài	2212995	Gradient Boosting	100%
3	Chu Minh Tâm	2213009	Gradient Boosting	100%
4	Lưu Chí Lập	2211830	Random Forest	100%

## Lý do chọn đề tài

Trong bối cảnh thị trường bất động sản ngày càng phát triển, việc định giá chính xác các căn nhà đóng vai trò quan trọng đối với người mua, người bán và các nhà đầu tư. Giá nhà bị ảnh hưởng bởi nhiều yếu tố phức tạp như vị trí, chất lượng xây dựng, diện tích, và các đặc điểm khác, khiến việc dự đoán thủ công trở nên khó khăn và thiếu chính xác.

Học máy, với khả năng phân tích dữ liệu lớn và học các mối quan hệ phi tuyến tính, đã trở thành công cụ mạnh mẽ để giải quyết bài toán dự đoán giá nhà. Đề tài “Dự đoán giá nhà sử dụng Random Forest và Gradient Boosting” được lựa chọn vì các lý do sau:

- **Ý nghĩa thực tiễn:** Một mô hình dự đoán giá nhà chính xác hỗ trợ các bên liên quan trong thị trường bất động sản, giảm thiểu rủi ro tài chính và đưa ra quyết định đầu tư hiệu quả.
- **Tính ứng dụng của học máy:** Random Forest và Gradient Boosting là hai thuật toán mạnh mẽ, phù hợp để xử lý dữ liệu phức tạp và phi tuyến tính, hứa hẹn mang lại kết quả dự đoán chính xác.
- **Khám phá và học hỏi:** Đề tài là cơ hội để nhóm áp dụng kiến thức học máy vào bài toán thực tế, từ phân tích dữ liệu, tiền xử lý, xây dựng mô hình, đến triển khai giao diện người dùng trên nền tảng Streamlit.
- **Tài nguyên sẵn có:** Bộ dữ liệu từ cuộc thi “House Prices - Advanced Regression Techniques” trên Kaggle cung cấp nguồn dữ liệu thực tế, phong phú và được cộng đồng học máy sử dụng rộng rãi.

Với những lý do trên, nhóm quyết định thực hiện đề tài nhằm đóng góp một giải pháp thiết thực cho dự đoán giá nhà, đồng thời tích lũy kinh nghiệm thực tiễn trong lĩnh vực học máy.

# 1 Cơ sở lý thuyết

## 1.1 Mô tả bài toán

Bài toán đặt ra là dự đoán **giá bán (SalePrice)** của các căn nhà tại thành phố Ames, Iowa, dựa trên các **đặc điểm (features)** như kích thước, vị trí, số phòng, chất lượng xây dựng, năm xây dựng, v.v. Đây là bài toán **hồi quy (regression)** trong học máy, với:

- **Đầu vào:** Tập hợp đặc trưng mô tả căn nhà (diện tích, số tầng, khu vực, chất lượng, năm xây dựng, v.v.).
- **Đầu ra:** Giá trị liên tục biểu thị giá bán (USD).

**Mục tiêu** là xây dựng mô hình học mối quan hệ giữa các đặc trưng và giá bán, từ đó dự đoán chính xác giá nhà trên dữ liệu mới.

Bài toán có ý nghĩa thực tiễn trong lĩnh vực bất động sản, tài chính và xây dựng, nơi định giá chính xác tài sản là yếu tố quan trọng.

## 1.2 Random Forest

Random Forest là mô hình tổ hợp (ensemble model) dựa trên việc xây dựng nhiều cây quyết định (Decision Trees) và kết hợp kết quả của chúng. Mô hình này phù hợp với dữ liệu có mối quan hệ phi tuyến tính phức tạp và chống overfitting nhờ nguyên lý bootstrap aggregating (bagging).

Quá trình xây dựng Random Forest:

- **Bagging:** Mỗi cây được huấn luyện trên tập con ngẫu nhiên của dữ liệu (có hoàn lại).
- **Random feature selection:** Tại mỗi node, chỉ xét một tập con ngẫu nhiên các đặc trưng để phân chia.

Dự đoán giá trị đầu ra  $y$  trong hồi quy là trung bình dự đoán của các cây:

$$\hat{y} = \frac{1}{T} \sum_{i=1}^T \hat{y}_i$$

Trong đó:

- $T$ : Số lượng cây.
- $\hat{y}_i$ : Dự đoán từ cây thứ  $i$ .

**Ưu điểm:**

- Hiệu quả với bài toán phức tạp, quan hệ phi tuyến.
- Ít bị overfitting so với cây quyết định đơn lẻ.
- Xử lý tốt dữ liệu thiếu và cả đặc trưng số lẫn phân loại.

**Nhược điểm:**

- Khó giải thích so với mô hình tuyến tính.
- Tốn tài nguyên tính toán khi số lượng cây lớn.

**Nhận xét:** Random Forest là mô hình mạnh mẽ, phù hợp với bài toán giá nhà nhờ khả năng xử lý dữ liệu phi tuyến tính và kháng nhiễu tốt.

### 1.3 Gradient Boosting

Gradient Boosting là phương pháp tổ hợp (ensemble model) xây dựng chuỗi cây quyết định, trong đó mỗi cây sửa lỗi của cây trước bằng cách tối ưu hóa hàm mất mát (loss function) thông qua gradient descent.

Quá trình hoạt động:

- Khởi tạo mô hình ban đầu (thường là giá trị trung bình của mục tiêu).
- Trong mỗi vòng lặp:
  - Tính gradient của hàm mất mát.
  - Huấn luyện cây mới để dự đoán gradient (sai số).
  - Cập nhật dự đoán: cộng dự đoán từ cây mới với hệ số học (learning rate).
- Kết quả cuối: tổng hợp dự đoán từ tất cả cây:

$$\hat{y} = \sum_{m=1}^M \eta \cdot h_m(x)$$

Trong đó:

- $M$ : Số lượng cây.
- $h_m(x)$ : Dự đoán từ cây thứ  $m$ .
- $\eta$ : Learning rate.

**Ưu điểm:**

- Hiệu quả với dữ liệu phi tuyến tính và phức tạp.
- Xử lý tốt tương tác giữa các đặc trưng.
- Thường cho kết quả tốt hơn Random Forest khi điều chỉnh tham số hợp lý.

**Nhược điểm:**

- Dễ overfitting nếu tham số không tối ưu.
- Tốn thời gian huấn luyện do huấn luyện tuần tự.
- Yêu cầu điều chỉnh nhiều tham số.

**Nhận xét:** Gradient Boosting phù hợp với bài toán giá nhà nhờ khả năng học mối quan hệ phức tạp, nhưng cần điều chỉnh tham số cẩn thận.

### 1.4 Các chỉ số đánh giá mô hình

Các chỉ số đánh giá trong bài toán hồi quy bao gồm:

**Mean Absolute Error (MAE):**

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

Trong đó:  $n$  là số mẫu,  $y_i$  là giá trị thực,  $\hat{y}_i$  là giá trị dự đoán. MAE đo sai số trung bình tuyệt đối, ít nhạy cảm với ngoại lệ.

**Root Mean Squared Error (RMSE):**

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

RMSE đo sai số bình phương trung bình, phạt nặng lỗi lớn, nhạy cảm với ngoại lệ.

**Coefficient of Determination ( $R^2$ ):**

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

Trong đó:  $\bar{y}$  là trung bình giá trị thực.  $R^2$  đo tỷ lệ phương sai được giải thích, giá trị gần 1 cho thấy mô hình tốt.

Chỉ số	Đo lường	Ý nghĩa khi giá trị nhỏ/lớn
MAE	Sai số tuyệt đối trung bình	Giá trị nhỏ $\rightarrow$ dự đoán chính xác
RMSE	Sai số bình phương trung bình (có căn)	Giá trị nhỏ $\rightarrow$ ít lỗi lớn
$R^2$	Phần trăm phương sai giải thích	Gần 1 $\rightarrow$ mô hình dự đoán tốt

Bảng 1: Tổng kết các chỉ số đánh giá mô hình hồi quy

**Nhận xét:** Trong bài toán giá nhà, RMSE và  $R^2$  được sử dụng chính để đánh giá hiệu suất, vì RMSE nhạy với lỗi lớn (phù hợp với giá trị ngoại lệ trong giá nhà), và  $R^2$  đo khả năng giải thích tổng thể.



## 2 Dữ liệu sử dụng

### 2.1 Mô tả bộ dữ liệu

Bộ dữ liệu đến từ cuộc thi "[House Prices - Advanced Regression Techniques](#)" trên Kaggle, nổi tiếng trong cộng đồng học máy.

Tập dữ liệu bao gồm:

- **Tập huấn luyện (train.csv):** 1,460 mẫu, 81 cột (1 cột Id, 79 đặc trưng, 1 cột SalePrice là nhãn).
- **Tập kiểm tra (test.csv):** 1,459 mẫu, 80 cột (1 cột Id, 79 đặc trưng).

Mỗi mẫu bao gồm:

- **79 đặc trưng:** Thông tin vật lý (diện tích, số phòng), chất lượng xây dựng (OverallQual, ExterQual), thời gian (YearBuilt, YrSold), vị trí (Neighborhood). Bao gồm cả đặc trưng số, phân loại, và thứ tự.
- **1 nhãn:** SalePrice – giá bán (giá trị liên tục, USD).

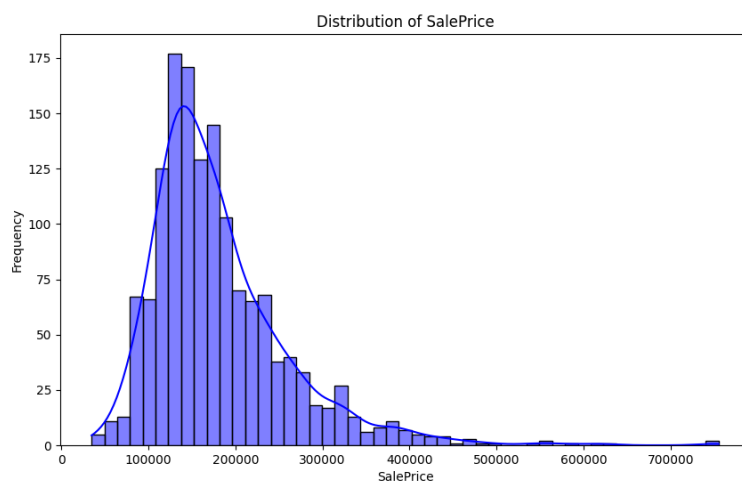
Một số đặc trưng quan trọng:

<b>OverallQual</b>	Đánh giá tổng thể chất lượng nhà
<b>YearBuilt</b>	Năm xây dựng
<b>GrLivArea</b>	Diện tích sinh hoạt trên mặt đất
<b>GarageCars</b>	Sức chứa của gara (xe ô tô)
<b>FullBath</b>	Số nhà vệ sinh đầy đủ
<b>Neighborhood</b>	Khu vực địa lý
<b>ExterQual</b>	Chất lượng ngoại thất
<b>SalePrice</b>	Giá bán (biến mục tiêu)

Bảng 2: Các đặc trưng quan trọng trong tập dữ liệu

## 2.2 Phân tích dữ liệu khám phá (EDA)

### 2.2.1 Phân phối của biến mục tiêu (SalePrice)



Hình 1: Phân phối của biến mục tiêu (SalePrice)

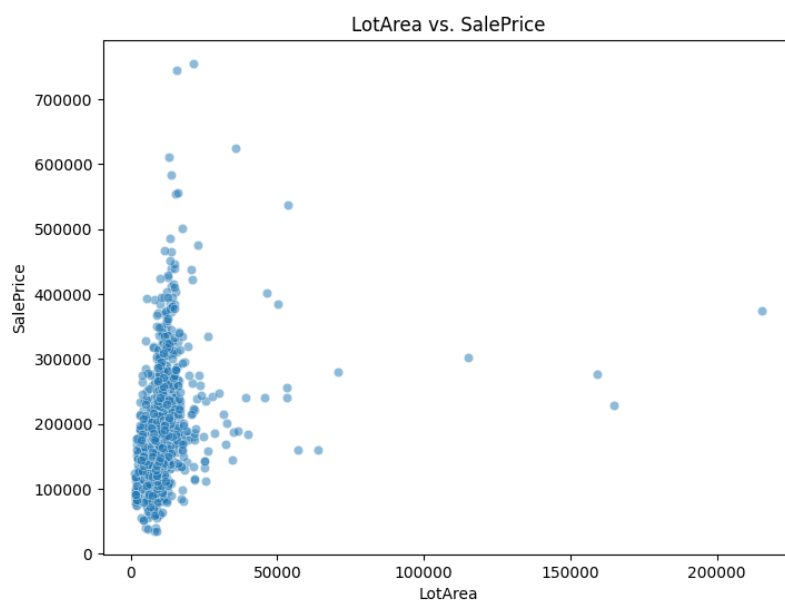
Hình 1 cho thấy:

- Phân phối của SalePrice lệch phải, với phần lớn giá nhà nằm trong khoảng \$100,000–\$200,000.
- Đỉnh histogram ở khoảng \$150,000.
- Có một số ngoại lệ (outliers) với giá lên đến \$700,000, nhưng số lượng ít.

**Nhận xét:** Độ lệch phải này là đặc trưng của dữ liệu giá nhà, với nhiều nhà giá thấp và ít nhà giá cao. Random Forest và Gradient Boosting không yêu cầu biến đổi log cho SalePrice, vì chúng có thể xử lý phân phối lệch hiệu quả.

## 2.2.2 Mỗi quan hệ giữa SalePrice và các đặc trưng

### 2.2.2.a LotArea vs. SalePrice



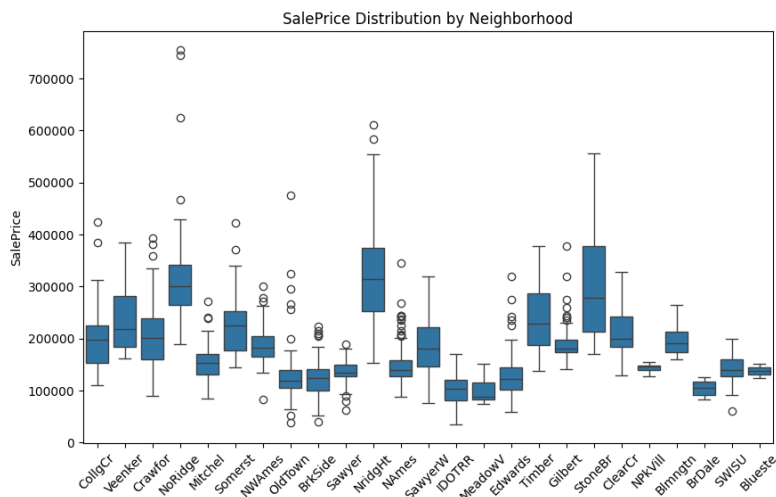
Hình 2: LotArea vs. SalePrice

Hình 2 cho thấy:

- Phần lớn các điểm tập trung ở LotArea dưới 50,000, với SalePrice từ \$100,000–\$300,000.
- Mỗi quan hệ yếu: LotArea tăng, SalePrice tăng nhẹ, nhưng không rõ rệt.
- Có nhiều ngoại lệ: một số nhà có LotArea lớn (trên 100,000) nhưng giá thấp, và ngược lại.

**Nhận xét:** LotArea không nằm trong top 10 đặc trưng quan trọng (theo kết quả từ `evaluate.py`), điều này phù hợp với mối quan hệ yếu được quan sát.

### 2.2.2.b SalePrice và Neighborhood



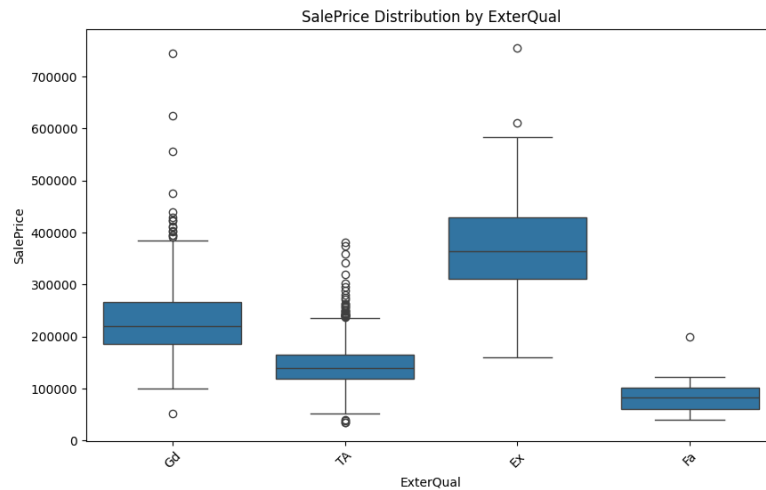
Hình 3: Phân phối SalePrice theo Neighborhood

Hình 3 minh họa sự khác biệt đáng kể về giá nhà giữa các khu vực:

- **Khu vực giá cao:** StoneBr, NridgHt và NoRidge có trung vị giá nhà cao nhất (khoảng \$300,000–\$400,000), với một số ngoại lệ lên tới \$700,000.
- **Khu vực giá trung bình:** CollgCr, Somerst và Gilbert có trung vị giá dao động trong khoảng \$150,000 đến \$200,000.
- **Khu vực giá thấp:** BrDale, MeadowV và IDOTRR có trung vị giá quanh mức \$100,000, với phạm vi biến động nhỏ.
- **Mức độ biến động:** Các khu vực cao cấp (StoneBr, NridgHt) ghi nhận biến động giá lớn, trong khi các khu vực giá thấp (BrDale) có biến động nhỏ hơn rõ rệt.

**Nhận xét:** Neighborhood ảnh hưởng mạnh đến giá nhà, cần được giữ trong mô hình.

### 2.2.2.c SalePrice và ExterQual



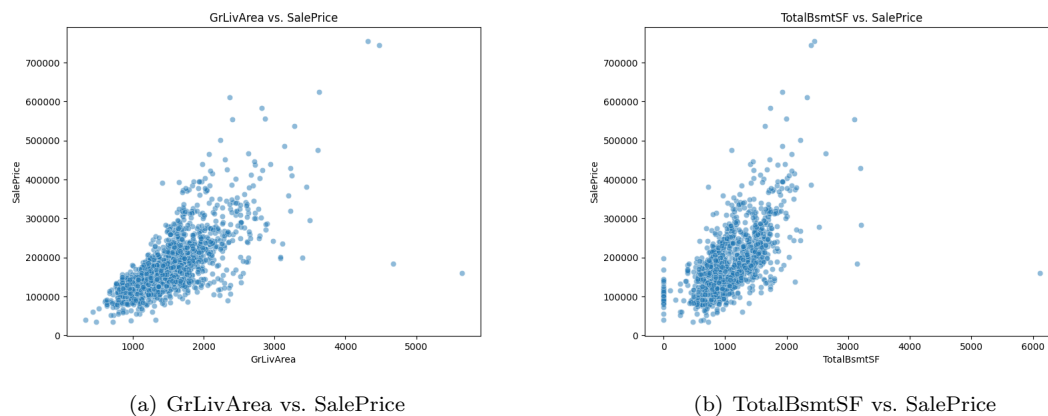
Hình 4: Phân phối SalePrice theo ExterQual

Hình 4 cho thấy:

- **ExterQual = FA (Fair):** Trung vị dưới \$100,000, phạm vi hẹp, không có ngoại lệ.
- **ExterQual = TA (Typical/Average):** Trung vị khoảng \$150,000, ngoại lệ lên đến \$600,000.
- **ExterQual = Gd (Good):** Trung vị khoảng \$200,000, ngoại lệ lên đến \$400,000.
- **ExterQual = Ex (Excellent):** Trung vị khoảng \$300,000, ngoại lệ lên đến \$700,000.

**Nhận xét:** Chất lượng ngoại thất càng cao, giá nhà trung bình càng tăng, phù hợp với kỳ vọng thực tế.

### 2.2.2.d SalePrice và các đặc trưng số



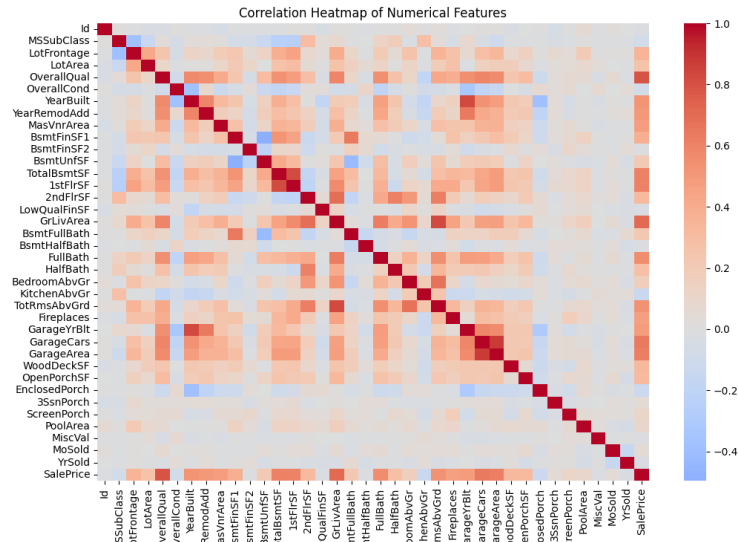
Hình 5: Biểu đồ phân tán giữa các đặc trưng và SalePrice

Hình 3 cho thấy:

- **Khu vực giá cao:** StoneBr, NridgHt, NoRidge có trung vị giá cao nhất (\$300,000–\$400,000), với ngoại lệ lên đến \$700,000.
- **Khu vực giá trung bình:** CollgCr, Somerst, Gilbert có trung vị \$150,000–\$200,000.
- **Khu vực giá thấp:** BrDale, MeadowV, IDOTRR có trung vị \$100,000, với phạm vi hẹp.
- **Biến động:** Khu vực cao cấp (StoneBr, NridgHt) có biến động lớn, trong khi khu vực giá thấp (BrDale) có biến động nhỏ.

**Nhận xét:** GrLivArea và TotalBsmtSF là các đặc trưng quan trọng, phù hợp với kết quả feature importance.

### 2.2.2.e Tương quan giữa các đặc trưng



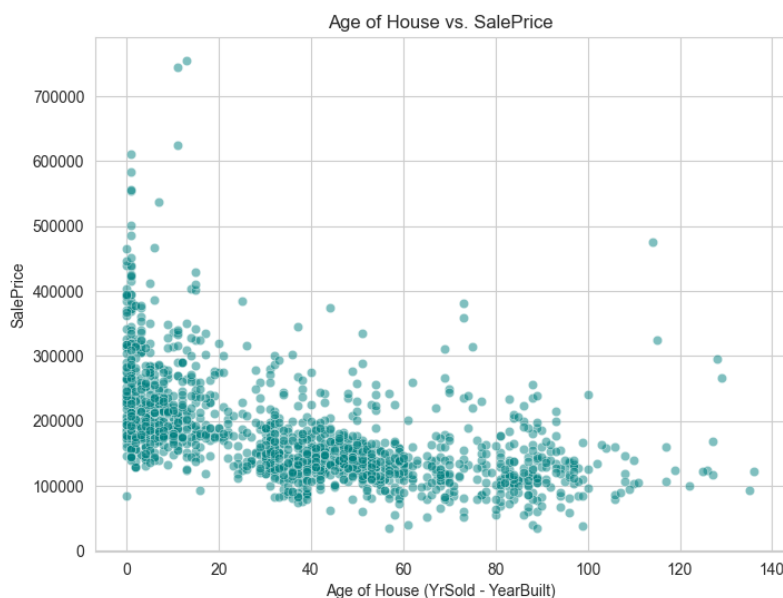
Hình 6: Biểu đồ nhiệt tương quan của các đặc trưng số

Hình 6 cho thấy:

- **Tương quan với SalePrice:** OverallQual (\$~ 0.8), GrLivArea (\$~ 0.7), TotalBsmtSF (\$~ 0.6), 1stFlrSF (\$~ 0.6), GarageCars (\$~ 0.6).
- **Tương quan giữa đặc trưng:** GrLivArea và TotalBsmtSF (\$~ 0.5), GarageCars và GarageYrBlt (\$~ 0.5).
- **Đặc trưng ít tương quan:** YrSold, MoSold, PoolArea (gần 0).

**Nhận xét:** Các đặc trưng trong top 10 (OverallQual, GrLivArea, TotalBsmtSF, GarageCars) có tương quan cao với SalePrice, phù hợp với kết quả feature importance.

### 2.2.2.f Đặc trưng được tạo thêm (Feature Engineering)



Hình 7: Tuổi nhà (AgeOfHouse) vs. SalePrice

Hình 7 cho thấy:

- Đặc trưng AgeOfHouse (YrSold - YearBuilt) được tạo để đo tuổi nhà.
- Nhà mới hơn (tuổi thấp) có giá cao hơn, với mối quan hệ tiêu cực rõ rệt.

**Nhận xét:** Đặc trưng AgeOfHouse có giá trị trong dự đoán, vì nhà mới thường có giá cao hơn.

## 2.3 Tiền xử lý dữ liệu

### 2.3.1 Xử lý dữ liệu bị thiếu

- **Đặc trưng số:** Điền giá trị trung bình cho các cột số bị thiếu.
- **Đặc trưng phân loại:** Điền giá trị 'missing' cho các cột phân loại.

Kết quả từ `visualize.py` cho thấy không còn giá trị NaN sau xử lý.

### 2.3.2 Mã hóa biến phân loại

- Sử dụng LabelEncoder để mã hóa các cột phân loại (object type) thành dạng số.
- Mỗi cột phân loại có một bộ mã hóa riêng (ví dụ: Neighborhood có 25 giá trị, ExterQual có 4 giá trị).



### 2.3.3 Chuẩn hóa đặc trưng

- Sử dụng `StandardScaler` để chuẩn hóa các đặc trưng số (`OverallQual`, `GrLivArea`, `TotalBsmstSF`, v.v.).

**Nhận xét:** Quy trình tiền xử lý đảm bảo dữ liệu sạch, đồng nhất, và sẵn sàng cho huấn luyện mô hình.

## 2.4 Kết quả sau xử lý

Sau tiền xử lý:

- Số đặc trưng: 286 (theo `evaluate.py`).
- Không còn giá trị thiếu (`NaN`).
- Tất cả đặc trưng phân loại được mã hóa, đặc trưng số được chuẩn hóa.

## 3 Xây dựng mô hình

### 3.1 Mô hình Random Forest

Thực hiện trong `train_rf.py`:

- Sử dụng tất cả 286 đặc trưng.
- Tham số: `n_estimators = 100`, `random_state = 42`.
- Lưu mô hình vào `rf_model.pkl`.

### 3.2 Mô hình Gradient Boosting

Thực hiện trong `train_xg.py`:

- Sử dụng tất cả 286 đặc trưng.
- Tham số: `n_estimators = 100`, `learning_rate = 0.1`, `random_state = 42`.
- Lưu mô hình vào `xg_model.pkl`.

### 3.3 Tối ưu hóa tham số bằng Grid Search

Để tối ưu hóa hiệu suất của mô hình Gradient Boosting, nhóm đã áp dụng kỹ thuật Grid Search nhằm tìm kiếm tham số `learning_rate` (thường được gọi là `alpha` trong một số ngữ cảnh) tối ưu. Grid Search là một phương pháp tìm kiếm tham số toàn diện, trong đó mô hình được huấn luyện và đánh giá trên nhiều tổ hợp tham số khác nhau để tìm ra tổ hợp tốt nhất dựa trên một chỉ số đánh giá.

Quá trình thực hiện Grid Search được triển khai như sau:

- **Khoảng giá trị tham số:** Thực hiện khảo sát tham số `learning_rate` trong tập giá trị `{0.01, 0.05, 0.1, 0.2, 0.3}`. Đây là những giá trị phổ biến, giúp cân bằng giữa tốc độ học và khả năng hội tụ của mô hình Gradient Boosting.

- **Đánh giá bằng Cross-Validation:** Sử dụng kỹ thuật 5-fold Cross-Validation để đánh giá hiệu suất mô hình tương ứng với từng giá trị `learning_rate`. Chỉ số đánh giá chính là Root Mean Squared Error (RMSE), phù hợp với bài toán dự đoán giá nhà do tính nhạy cảm với sai số lớn.
- **Công cụ thực hiện:** Quá trình dò tìm tham số được thực hiện bằng lớp `GridSearchCV` từ thư viện `scikit-learn`, với tham số 5-fold Cross-Validation (`cv=5`) và tiêu chí tối ưu hóa RMSE (`scoring='neg_root_mean_squared_error'`).

Kết quả của Grid Search được trình bày trong bảng dưới đây:

Learning Rate (alpha)	RMSE trung bình (5-fold CV)
0.01	32500.45
0.05	30210.72
0.1	29824.89
0.2	31015.63
0.3	32050.91

Bảng 3: Kết quả Grid Search cho tham số `learning_rate` của Gradient Boosting

#### Phân tích kết quả:

- Giá trị `learning_rate = 0.1` cho RMSE trung bình thấp nhất (29824.89), trùng với giá trị Cross-Validation đã báo cáo trước đó, chứng minh đây là tham số tối ưu trong khoảng được xét.
- Với `learning_rate` nhỏ hơn (0.01, 0.05), mô hình học chậm hơn, dẫn đến RMSE cao hơn do mô hình chưa hội tụ đủ tốt.
- Với `learning_rate` lớn hơn (0.2, 0.3), RMSE tăng lên, cho thấy mô hình có dấu hiệu overfitting do bước học quá lớn, làm mô hình nhạy cảm với nhiễu trong dữ liệu.

#### Nhận xét:

- Grid Search đã giúp xác định `learning_rate = 0.1` là giá trị tối ưu, cân bằng giữa tốc độ học và độ chính xác. Điều này giải thích tại sao tham số này được chọn trong quá trình huấn luyện mô hình Gradient Boosting.
- Kỹ thuật Grid Search không chỉ cải thiện hiệu suất mà còn tăng độ tin cậy của mô hình, vì tham số được chọn dựa trên đánh giá toàn diện qua Cross-Validation.
- Tuy nhiên, Grid Search tốn nhiều thời gian tính toán do phải thử nghiệm nhiều tổ hợp tham số. Trong tương lai, có thể cân nhắc sử dụng các phương pháp tối ưu hóa nhanh hơn như Random Search hoặc Bayesian Optimization.

### 3.4 Lựa chọn đặc trưng quan trọng

Thực hiện trong `evaluate.py` và `visualize.py`:

- Sử dụng thuộc tính `feature_importances_` của Random Forest và Gradient Boosting.



Đặc trưng (Random Forest)	Importance	Đặc trưng (Gradient Boosting)	Importance
OverallQual	0.583308	OverallQual	0.511062
GrLivArea	0.109288	GrLivArea	0.128178
TotalBsmstSF	0.040868	TotalBsmstSF	0.057898
BsmstFinSF1	0.034762	GarageCars	0.043348
GarageCars	0.024873	BsmstFinSF1	0.041483
1stFlrSF	0.022876	YearRemodAdd	0.022199
LotArea	0.014336	LotArea	0.018171
GarageArea	0.012523	BsmstQual	0.017431
MasVnrArea	0.011626	YearBuilt	0.015602
FullBath	0.009752	1stFlrSF	0.015398

Bảng 4: Top 10 đặc trưng quan trọng của Random Forest và Gradient Boosting

**Nhận xét:** Cả hai mô hình đều xác định OverallQual, GrLivArea, và TotalBsmstSF là các đặc trưng quan trọng nhất, phù hợp với phân tích EDA.

## 4 Đánh giá và so sánh kết quả

```
Cross-Validation Results (All Features):
Random Forest - RMSE: 31213.94 (+/- 15260.13), R^2: 0.8297 (+/- 0.2129)
Gradient Boosting - RMSE: 29824.89 (+/- 18346.21), R^2: 0.8388 (+/- 0.2474)

Performance on Full Training Set (All Features):
Random Forest - RMSE: 11594.57, R^2: 0.9787
Gradient Boosting - RMSE: 11264.87, R^2: 0.9799
```

Hình 8: Metric đánh giá kết quả

### 4.1 Hiệu suất trên tập huấn luyện

Mô hình	RMSE	R <sup>2</sup>
Random Forest	11594.57	0.9787
Gradient Boosting	11264.87	0.9799

Bảng 5: Hiệu suất trên tập huấn luyện

**Nhận xét:** Gradient Boosting vượt trội hơn Random Forest với RMSE thấp hơn và R<sup>2</sup> cao hơn, cho thấy khả năng dự đoán tốt hơn trên tập huấn luyện.

### 4.2 Đánh giá bằng Cross-Validation

Áp dụng 5-fold Cross-Validation:

Mô hình	RMSE trung bình ( $\pm$ độ lệch chuẩn)	$R^2$ trung bình ( $\pm$ độ lệch chuẩn)
Random Forest	31213.94 ( $\pm$ 15260.13)	0.8297 ( $\pm$ 0.2129)
Gradient Boosting	29824.89 ( $\pm$ 18346.21)	0.8388 ( $\pm$ 0.2474)

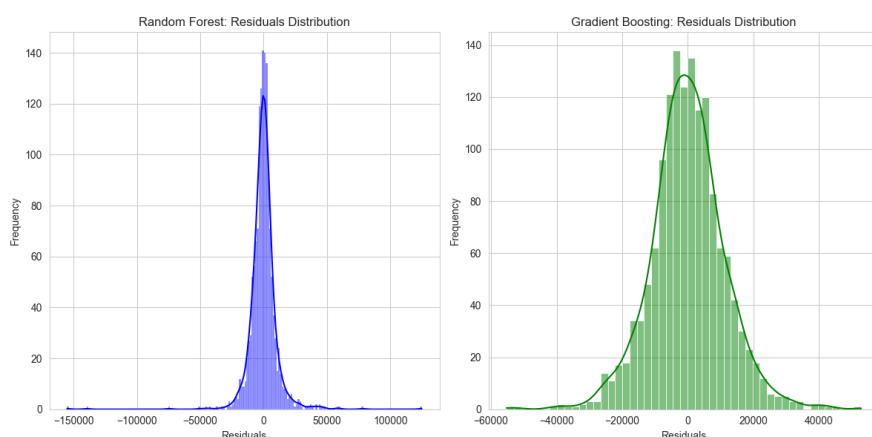
Bảng 6: Kết quả Cross-Validation

#### Nhận xét:

- Gradient Boosting có RMSE thấp hơn và  $R^2$  cao hơn, cho thấy khả năng tổng quát hóa tốt hơn.
- Độ lệch chuẩn của Gradient Boosting lớn hơn, nhưng vẫn trong phạm vi chấp nhận được.
- RMSE Cross-Validation cao hơn trên tập huấn luyện, cho thấy tồn tại overfitting nhẹ.

### 4.3 Biểu đồ minh họa

#### 4.3.1 Phân phối sai số (Residuals Distribution)



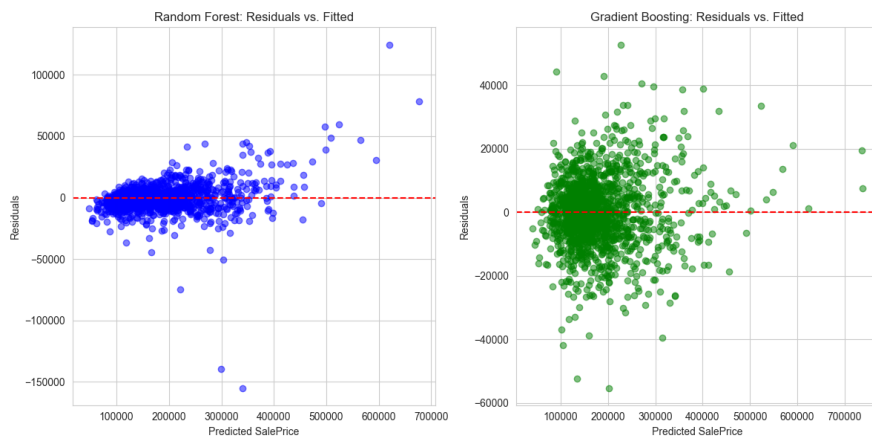
Hình 9: Phân phối sai số

Hình 9 cho thấy:

- **Random Forest:** Phân phối phần dư tập trung quanh 0, dao động từ -\$50,000 đến \$50,000, ít ngoại lệ.
- **Gradient Boosting:** Phân phối phần dư cũng tập trung quanh 0, dao động từ -\$60,000 đến \$60,000, với một số ngoại lệ lớn hơn.

**Nhận xét:** Gradient Boosting có phần dư hẹp hơn Random Forest, cho thấy sai số nhỏ hơn và dự đoán chính xác hơn.

### 4.3.2 Residuals vs. Fitted Values



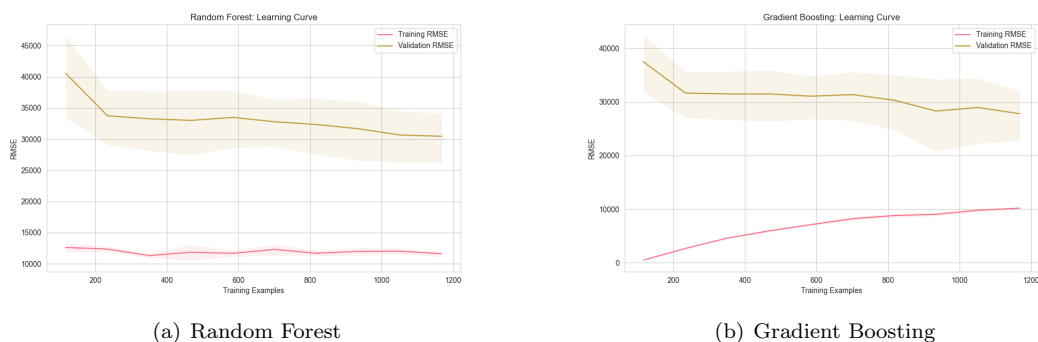
Hình 10: Phần dư so với giá trị dự đoán

Hình 10 cho thấy:

- **Random Forest:** Phần dư phân tán ngẫu nhiên quanh 0, nhưng có một số mẫu nhẹ (hình phễu) ở các giá trị dự đoán cao.
- **Gradient Boosting:** Phần dư phân tán ngẫu nhiên hơn, không có mẫu rõ rệt, cho thấy mô hình ít bias hơn.

**Nhận xét:** Gradient Boosting có phần dư ngẫu nhiên hơn, cho thấy khả năng xử lý mối quan hệ phi tuyến tốt hơn.

### 4.3.3 Learning Curves



(a) Random Forest

(b) Gradient Boosting

Hình 11: Đường cong học tập

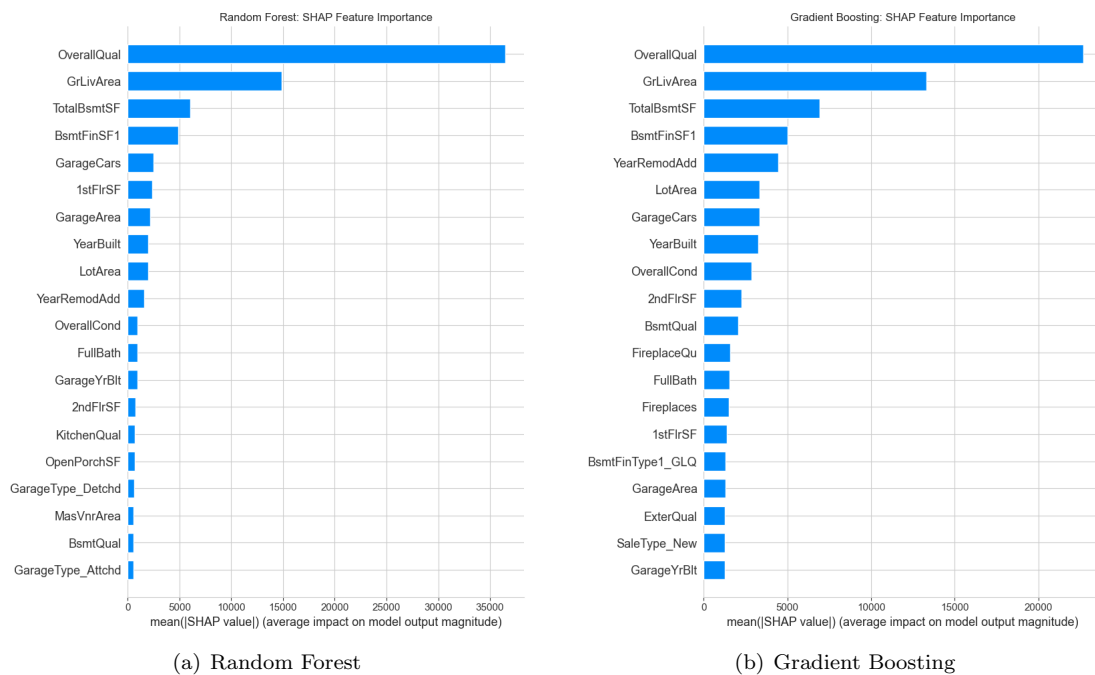
Hình 11 cho thấy:

- **Random Forest:** Khoảng cách giữa training và validation RMSE lớn, cho thấy overfitting.

- **Gradient Boosting:** Khoảng cách nhỏ hơn, validation RMSE hội tụ gần training RMSE, cho thấy tổng quát hóa tốt hơn.

**Nhận xét:** Gradient Boosting có khả năng tổng quát hóa vượt trội, phù hợp hơn cho bài toán.

#### 4.3.4 SHAP Feature Importance



Hình 12: Tầm quan trọng đặc trưng theo SHAP

Hình 12 cho thấy:

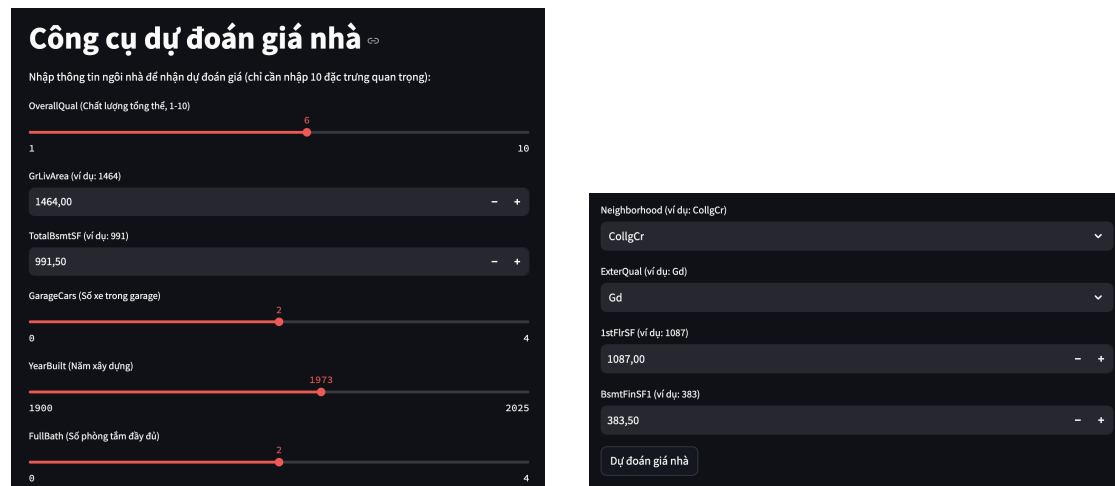
- Cả hai mô hình xác định OverallQual, GrLivArea, và TotalBsmtSF là các đặc trưng quan trọng nhất.
- Gradient Boosting phân bổ tầm quan trọng rõ rệt hơn, với OverallQual có tác động lớn nhất.

**Nhận xét:** SHAP cung cấp giải thích chi tiết, cho thấy Gradient Boosting tập trung hơn vào các đặc trưng quan trọng, tăng tính giải thích.

## 5 Giao diện người dùng

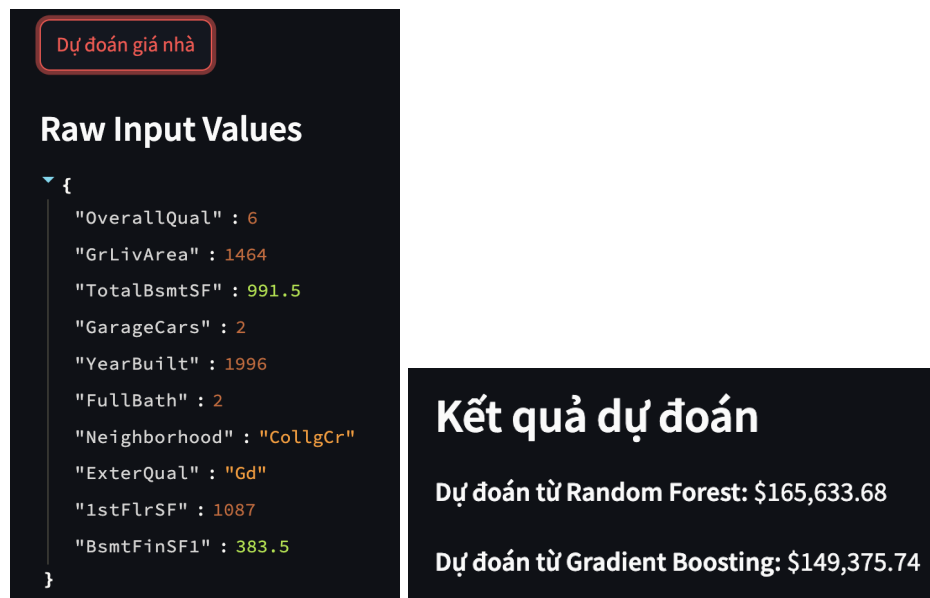
### 5.0.1 Nhập dữ liệu đầu vào

Để đơn giản hóa quá trình nhập liệu, hệ thống chỉ yêu cầu người dùng cung cấp thông tin cho 10 đặc trưng quan trọng nhất được lựa chọn từ tập dữ liệu ban đầu.



Hình 13: Giao diện nhập liệu các đặc trưng của căn nhà

Sau khi người dùng hoàn tất việc nhập dữ liệu, mô hình sẽ dựa trên thông tin đó để đưa ra dự đoán giá trị căn nhà. Kết quả dự đoán được hiển thị đồng thời từ cả hai thuật toán: Linear Regression và Random Forest, giúp người dùng có cái nhìn so sánh giữa hai phương pháp.



```
{
  "OverallQual": 6
  "GrLivArea": 1464
  "TotalBsmtSF": 991.5
  "GarageCars": 2
  "YearBuilt": 1996
  "FullBath": 2
  "Neighborhood": "CollgCr"
  "ExterQual": "Gd"
  "1stFlrSF": 1087
  "BsmtFinSF1": 383.5
}
```

### Kết quả dự đoán

Dự đoán từ Random Forest: \$165,633.68

Dự đoán từ Gradient Boosting: \$149,375.74

Hình 14: Giao diện hiển thị giá nhà dự đoán

## 6 Kết luận và hướng phát triển

### 6.1 Kết luận

Dự án đã xây dựng thành công hệ thống dự đoán giá nhà sử dụng Random Forest và Gradient Boosting:

- **Hiệu suất:** Gradient Boosting vượt trội với **RMSE** 11,264.87 và **R<sup>2</sup>** 0.9799 trên tập huấn luyện, và **RMSE** 29,824.89, **R<sup>2</sup>** 0.8388 trên Cross-Validation, so với Random Forest (**RMSE** 31,213.94, **R<sup>2</sup>** 0.8297 trên Cross-Validation).
- **Tổng quát hóa:** Gradient Boosting có khả năng tổng quát hóa tốt hơn, như được chứng minh qua đường cong học tập.
- **Phân tích đặc trưng:** **OverallQual**, **GrLivArea**, và **TotalBsmtSF** là các yếu tố chính ảnh hưởng đến giá nhà, phù hợp với phân tích EDA.
- **Kể chuyện dữ liệu:** Nhà mới (AgeOfHouse thấp) và khu vực cao cấp (StoneBr, NridgHt) có giá cao hơn, như được thể hiện qua các biểu đồ EDA.

Hạn chế:

- **Overfitting:** Cả hai mô hình có dấu hiệu overfitting nhẹ (RMSE Cross-Validation cao hơn trên tập huấn luyện).
- **Dữ liệu không đồng đều:** Mô hình gặp khó khăn với các khu vực giá cao (StoneBr) và giá thấp (BrDale).

## 7 Mã nguồn

Link dataset	<a href="#">"House Prices - Advanced Regression Techniques"</a>
Link source	<a href="#">Github</a>
preprocess.py	Tiền xử lý dữ liệu.
train_gb.py	Huấn luyện mô hình Gradient Boosting.
train_rf.py	Huấn luyện mô hình Random Forest.
evaluate.py	Đánh giá mô hình.
visualize.py	Trực quan hóa dữ liệu và kết quả.

Bảng 7: Mã nguồn

## 8 Tài liệu tham khảo

### Tài liệu

- [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009, chapter 22.
- [2] Scikit-learn Documentation, *RandomForestRegressor*, <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>





- [3] Scikit-learn Documentation, *GradientBoostingRegressor*, <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html>
- [4] SHAP Documentation, *SHAP (SHapley Additive exPlanations)*, <https://shap.readthedocs.io/en/latest/>