Xây dựng mô hình dự đoán giá nhà đất

Tuấn Kiệt Trịnh[1*], Gia Bảo Ngô[1], Ngọc Thành Minh Phan[1], Phùng Nhật Khanh Nguyễn[1], Hoàng Long Nguyễn[1†]

[1]Khoa Công nghệ thông tin, Trường Đại học Khoa học tự nhiên, ĐHQG-HCM, Thành phố Hồ Chí Minh, Việt Nam .

*Corresponding author(s). E-mail(s): 2512026335@student.hcmus.edu.vn;
Contributing authors: 2512047728@student.hcmus.edu.vn;
2512048368@student.hcmus.edu.vn; 2512048079@student.hcmus.edu.vn;
2512037752@student.hcmus.edu.vn;
†Các tác giả đóng góp công sức vào bài làm là như nhau.

Tóm tắt nội dung

Dự án này nhằm mục tiêu xây dựng mô hình dự đoán giá nhà dựa trên các yếu tố ảnh hưởng đến giá trị bất động sản, bao gồm tuổi nhà, khoảng cách đến trạm MRT, số lượng cửa hàng tiện lợi lân cận và tọa độ địa lý. Việc dự đoán chính xác giá nhà giúp người mua, người bán và các nhà phân tích bất động sản đưa ra quyết định hợp lý, đồng thời hỗ trợ nghiên cứu thị trường và định giá tài sản. Trong nghiên cứu này, hai phương pháp chính được áp dụng: *K-Nearest Neighbors Regression* (KNN) và *K-Means clustering* kết hợp với KNN. KNN dựa trên nguyên lý khoảng cách để dự đoán giá trị dựa trên những ngôi nhà tương tự, trong khi phương pháp kết hợp với K-Means giúp phân nhóm dữ liệu thành các cụm có đặc điểm tương đồng, từ đó cải thiện khả năng dự đoán của KNN bằng cách học riêng biệt trong từng cụm.

Dữ liệu được tiền xử lý kỹ lưỡng, bao gồm việc loại bỏ các thuộc tính không cần thiết, biến đổi log để giảm thiểu phân bố lệch, loại bỏ ngoại lai và chuẩn hóa dữ liệu. Các thí nghiệm đánh giá hiệu quả mô hình dựa trên các chỉ số như $\boldsymbol{R^2}$, MAE, RMSE và MAPE. Kết quả thực nghiệm cho thấy mô hình kết hợp K-Means và KNN cho dự đoán ổn định hơn so với KNN thông thường, nhờ khả năng nắm bắt các đặc điểm riêng của từng nhóm nhà khác nhau.

Kết quả này góp phần cung cấp một hướng tiếp cận hiệu quả để phân tích và dự đoán giá nhà, đồng thời mở ra cơ hội ứng dụng trong các hệ thống tư vấn bất động sản thông minh.

Keywords: House price prediction, KNN, K-means, Regression

1 Giới thiệu

1.1 Tổng quan

Giá bất động sản là một trong những yếu tố kinh tế quan trọng, chịu ảnh hưởng bởi nhiều đặc trưng khác nhau như vị trí địa lý, cơ sở hạ tầng, tiện ích xung quanh và các đặc điểm của công trình. Việc dự đoán chính xác giá nhà đất không chỉ có ý nghĩa đối với người mua và người bán mà còn đóng vai trò quan trọng trong công tác quản lý, quy hoạch đô thị và hỗ trợ ra quyết định đầu tư.[1]

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của khoa học dữ liệu và học máy (machine learning), nhiều phương pháp dự đoán dựa trên dữ liệu đã được đề xuất và áp dụng nhằm mô hình hóa mối quan hệ phức tạp giữa các đặc trưng đầu vào và giá nhà[2]. Có thể kể đến các phương pháp thống kê và tuyến tính như *Linear Regression*, *Elastic Net*, các thuật toán cây như *Decision Tree Regression*, *Random Forest* là các phương pháp được khai thác rộng rãi trong nhiều nghiên cứu về dự đoán giá nhà. Đặc biệt, việc sử dụng các mô hình lai *(hybrid models)* kết hợp phương pháp học không tham số như *K-Nearest Neighbors* (KNN)[3] và các kỹ thuật phân cụm như *K-means* được đánh giá cao nhờ tính trực quan, dễ triển khai và khả năng thích ứng tốt với các cấu trúc dữ liệu phi tuyến, không đồng nhất.

Tuy nhiên, hiệu quả của các phương pháp này phụ thuộc lớn vào quy trình tiền xử lý dữ liệu, chuẩn hóa đặc trưng và lựa chọn siêu tham số, đặc biệt là số lượng láng giềng trong KNN và số cụm trong K-means. Bên cạnh đó, việc đánh giá mô hình không đúng cách có thể dẫn đến hiện tượng rò rỉ dữ liệu (data leakage)[4], làm sai lệch kết quả và giảm độ tin cậy của mô hình.

1.2 Mục tiêu và các đóng góp chính

Mục tiêu chính của báo cáo này là xây dựng và đánh giá một mô hình dự đoán giá nhà đất dựa trên dữ liệu thực tế bằng cách kết hợp phương pháp hồi quy KNN với kỹ thuật phân cụm K-means. Cách tiếp cận này nhằm khai thác cấu trúc cục bộ của dữ liệu để nâng cao độ chính xác dự đoán so với việc sử dụng mô hình KNN đơn lẻ.

Các đóng góp chính của bài báo cáo bao gồm:

- Thực hiện phân tích khám phá dữ liệu và tiền xử lý dữ liệu bất động sản, bao gồm biến đổi logarit đối với các biến có phân phối lệch và loại bỏ ngoại lai sau biến đổi nhằm giảm ảnh hưởng của các giá trị cực đoan.
- Xây dựng mô hình hồi quy KNN cho bài toán dự đoán giá nhà và phân tích ảnh hưởng của tham số số láng giềng đến hiệu năng mô hình.
- Đề xuất mô hình kết hợp *K-means + KNN*, trong đó dữ liệu được phân cụm trước bằng K-means, sau đó huấn luyện một mô hình KNN riêng với tham số tối ưu cho từng cụm.
- Sử dụng các thước đo đánh giá phù hợp cho bài toán hồi quy, bao gồm MAE, RMSE, MAPE và hệ số xác định R^2[5]. Đối với bài toán phân cụm, Inertia được sử dụng như một chỉ báo hỗ trợ nhằm phân tích xu hướng khi thay đổi số cụm thông qua phương pháp elbow, trong khi Silhouette Score[6] được áp dụng để đánh giá chất lượng phân cụm và lựa chọn số cụm tối ưu.

- Xây dựng quy trình huấn luyện, xác thực và kiểm tra chặt chẽ nhằm đảm bảo tính khách quan và tránh rò rỉ dữ liệu trong quá trình lựa chọn siêu tham số.

1.3 Cấu trúc bài báo cáo

Bài báo cáo được tổ chức thành các phần như sau:

- **Phần 1: Giới thiệu**
  Phần này trình bày bối cảnh và tầm quan trọng của việc dự đoán giá nhà, đồng thời nêu ra một số phương pháp dự đoán đã được áp dụng. Ngoài ra, chương này cũng trình bày mục tiêu của dự án, phương pháp tiếp cận tổng quát cũng như những đóng góp chính của dự án.
- **Phần 2: Kiến thức nền tảng**
  Phần này trình bày các kiến thức nền tảng liên quan đến các phương pháp KNN, K-means và các thước đo đánh giá sai số.
- **Phần 3: Mô hình dự đoán giá nhà đất**
  Phần này là phần trọng tâm của bài báo cáo, mô tả chi tiết mô hình dự đoán giá nhà đất, phương pháp thực hiện, quy trình thực nghiệm và kết quả đạt được.
- **Phần 4: Thảo luận**
  Phần này thảo luận về ý nghĩa thực tiễn và các hạn chế còn tồn tại của mô hình.
- **Phần 5: Đề xuất mô hình dự đoán khác (Linear Regression)**
  Phần này đề xuất việc sử dụng mô hình *Linear Regression* để dự đoán giá nhà, từ đó đưa ra nhận xét về hiệu quả dự đoán, và so sánh với mô hình lai K-means + KNN.
- **Phần 6: Kết luận**
  Phần này tổng kết các kết quả chính và đưa ra kết luận chung của dự án.

1.4 Phân công nhiệm vụ

Nhiệm vụ của các thành viên trong nhóm chúng em phân công theo kế hoạch như sau

Bảng 1 Bảng phân công nhiệm vụ của các thành viên trong nhóm theo từng tuần

Tuần	MSSV	Họ tên	Công việc đã thực hiện
Tuần 1 (18/11–25/11)	25120377	Nguyễn Hoàng Long	Import dữ liệu, xuất file cho tập train và test
	25120480	Nguyễn Phùng Nhật Khanh	Phân tích dữ liệu ban đầu
	25120263	Trịnh Tuấn Kiệt	Loại bỏ các giá trị ngoại lai (outliers)
	25120477	Ngô Gia Bảo	Chuẩn hóa dữ liệu
	25120483	Phan Ngọc Thành Minh	Chia dữ liệu thành hai tập train và test
Tuần 2 (02/12–09/12)	25120377	Nguyễn Hoàng Long	Tính giá trị trung bình, xây dựng hàm tính R^2
	25120480	Nguyễn Phùng Nhật Khanh	Tính khoảng cách trên tập huấn luyện
	25120263	Trịnh Tuấn Kiệt	So sánh kết quả, kiểm tra độ chính xác trên tập test
	25120477	Ngô Gia Bảo	Lựa chọn giá trị k tối ưu
	25120483	Phan Ngọc Thành Minh	Vẽ biểu đồ và hình ảnh minh họa
Tuần 3 (9/12–16/12)	25120377	Nguyễn Hoàng Long	Xây dựng hàm phân cụm (KMeans)
	25120480	Nguyễn Phùng Nhật Khanh	Tính toán lại trên tập test, đánh giá độ chính xác
	25120263	Trịnh Tuấn Kiệt	Xác định số cụm tối ưu
	25120477	Ngô Gia Bảo	Vẽ biểu đồ và hình ảnh minh họa
	25120483	Phan Ngọc Thành Minh	Tìm giá trị k tối ưu cho từng tập dữ liệu
Tuần 4 (16/12–23/12)	25120377	Nguyễn Hoàng Long	Viết báo cáo (Phần phương pháp nghiên cứu)
	25120480	Nguyễn Phùng Nhật Khanh	Viết báo cáo (Phần giới thiệu và kết luận)
	25120263	Trịnh Tuấn Kiệt	Viết báo cáo (Các mô hình khác và thảo luận)
	25120477	Ngô Gia Bảo	Viết báo cáo (Phần kiến thức nền tảng)
	25120483	Phan Ngọc Thành Minh	Viết báo cáo (Phần thực nghiệm, kết quả và đánh giá)

Bảng trình bày quá trình phân công và thực hiện nhiệm vụ của các thành viên trong nhóm suốt thời gian thực hiện đồ án.

2 Kiến thức nền tảng

2.1 K-nearest Neighbour

K-Nearest Neighbour (KNN) là một trong những thuật toán supervised-learning đơn giản nhất. KNN được xếp vào loại *lazy learning*. Khác với các thuật toán *eager learning*, KNN không xây dựng mô hình hay rút trích quy luật từ dữ liệu ngay từ đầu, mà chỉ thực hiện quá trình suy luận khi có yêu cầu dự đoán.[7]. KNN đều có thể được sử dụng cho cả hai bài toán Classification và Regression. Tuy nhiên đối với bài toán dự đoán giá nhà chúng tôi chỉ bàn đến bài toán Regression.

Cơ chế hoạt động của KNN là đầu ra của mô hình sẽ được tính toán bằng dựa vào K điểm dữ liệu gần nhất đối với dữ liệu đầu vào. Các bước để làm một bài toán Regression dùng KNN như sau:

1. Tính khoảng cách từ điểm đầu vào tới K điểm gần nhất
2. Sắp xếp khoảng cách của các điểm này theo thứ tự tăng dần
3. Tìm label của điểm mới bằng cách lấy giá trị trung bình cộng của K điểm gần nhất
4. Tìm K sao cho mô hình có Root mean square error (RMSE) nhỏ nhất

Khoảng cách giữa các điểm được tính bằng khoảng cách Euclid với công thức như sau:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

với p và q là các điểm trên không gian n-chiều

Ta cũng có thể viết lại dưới dạng vector:

$$d(p,q) = \|p - q\|$$

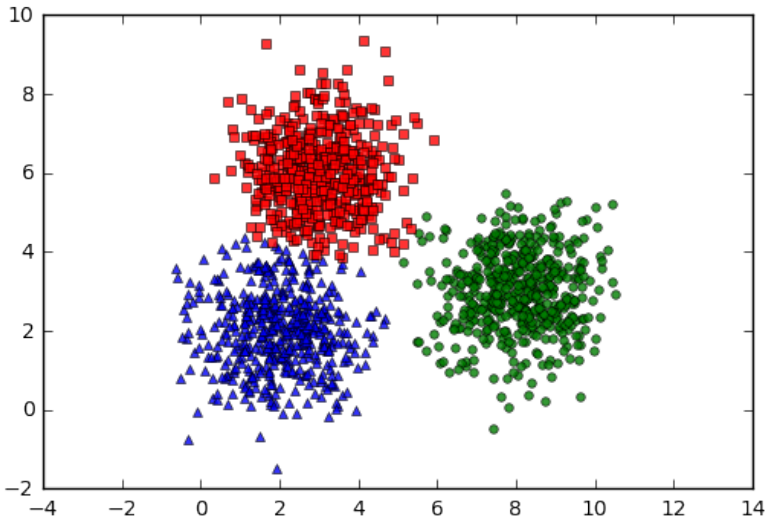
2.2 K-means Clustering

K-means Clustering (K-means) là thuật toán dùng để chia tập dữ liệu thành K tập sao cho mọi dữ liệu trong tập có tính chất gần giống nhau nhất có thể. Thuật toán K-means có thể được thực hiện như sau bởi thuật toán Lloyd[8] như sau:

1. Chọn K điểm ngẫu nhiên để làm tâm của tập
2. Với mỗi điểm, ta tính bình phương khoảng cách Euclid đến K tâm và phân điểm đó vào tập có tâm gần nhất

$$d^2(p,q) = (p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2$$

3. Tìm tâm mới bằng cách tính trung bình cộng các điểm trong tập đã chia
4. Lặp lại bước 2 và 3 cho đến khi giá trị thay đổi của các tâm bé hơn ϵ cho trước hoặc sau n lần lặp cho trước.



Hình 1 Dữ liệu được chia thành 3 tập nhờ thuật toán K-means

2.3 Các phương pháp đánh giá sai số

2.3.1 Mean absolute percentage error

Mean absolute percentage error (MAPE) là phần trăm chênh lệch giữa kết quả do mô hình dự đoán được và kết quả thật từ dữ liệu. MAPE được định nghĩa như sau:

$$\text{MAPE} = 100\frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

trong đó A_t là giá trị thực tế và F_t là giá trị dự đoán[5]

MAPE được dùng rất phổ biến trong các bài toán Regression bởi nó cho ta dễ hình dung về độ chính xác của mô hình hơn. Chỉ số MAPE nhỏ chứng tỏ độ chênh lệch nhỏ và mô hình chuẩn xác hơn. Tuy nhiên ta sẽ không tính được MAPE khi $A_t = 0$

2.3.2 Root mean squared error

Root mean square error (RMSE) cũng là một cách đo độ chính xác của mô hình dự đoán. Công thức tính RMSE như sau:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - x_i)^2}$$

trong đó $[X_1, X_2, ..., X_n]$ là các giá trị được dự đoán và $[x_1, x_2, ..., x_n]$ là các giá trị thực[5]

Ta có thể thấy RMSE cải thiện so với MAPE do có thể trả về kết quả khi $x_i = 0$. Cũng giống như MAPE, chỉ số RMSE càng nhỏ thì mô hình dự đoán càng chính xác. Tuy nhiên RMSE trừng phạt mạnh với sai số lớn hơn. Nhược điểm của RMSE là bị ảnh hưởng mạnh bởi các outlier.

2.3.3 Mean absolute error

Mean absolute error (MAE) là một cách khác để đo độ chênh lệch trung bình giữa giá trị dự đoán và giá trị thực tế. MAE có thể được tính như sau:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |A_i - F_i|$$

với A_i là giá trị thực tế và F_i là giá trị dự đoán[5]

Khác với RMSE, MAE không trừng phạt lỗi sai quá nhiều. Do đó, MAE ít bị ảnh hưởng bởi outlier

2.3.4 Coefficient of determination

Coefficient of determination ($\mathbf{R^2}$ score) là tỷ lệ biến thiên của biến phụ thuộc có thể được giải thích (hoặc dự đoán) từ biến độc lập (hoặc các biến độc lập). R^2 score có thể được tính như sau[5][9]:

Một dữ liệu có n giá trị $[y_1, y_2, ..., y_n]$ và tương ứng với mỗi giá trị là giá trị dự đoán từ mô hình $[f_1, f_2, ..., f_n]$

Ta có giá trị trung bình của giá trị thực như sau

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Ta có tổng phần dư được tính như sau:

$$\text{SS}_{\text{res}} = \sum_i (y_i - f_i)^2$$

Ta có tổng biến thiên dữ liệu:

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

Cuối cùng ta có:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

R^2 score dùng để cho ta thấy mô hình dự đoán tốt như thế nào dựa vào tổng biến thiên của dữ liệu. R^2 cho biết mô hình dự đoán tốt hơn bao nhiêu so với việc không sử dụng biến đầu vào.

2.3.5 Silhouette

Silhouette là giá trị để cho thấy sự tương đồng của một điểm dữ liệu khi so sánh giữa tập chứa nó và các tập khác[10]. Giá trị silhouette nằm trong khoảng (-1,1). Nếu đa số các dữ liệu có chỉ số silhouette cao thì số lượng tập chia là phù hợp. Ngược lại, nếu chỉ số của toàn tập thấp thì có nghĩa là chỉ số K quá cao hoặc quá thấp. Giá trị silhouette được tính như sau:

Với mọi điểm dữ liệu $i \in C_i$ (C_i là cụm dữ liệu chứa i), ta có:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j)$$

$|C_i|$ là số phần tử có trong cụm dữ liệu và $d(i, j)$ là khoảng cách Euclid giữa hai điểm dữ liệu

Với mọi $i \in C_i$, ta có:

$$b(i) = \min_{j \neq i} \frac{1}{C_j} \sum_{j \in C_j} d(i, j)$$

Ở đây $a(i)$ tính tổng khoảng cách của điểm dữ liệu i và các điểm dữ liệu chung cụm với nó. Với $b(i)$, nó là tổng khoảng cách nhỏ nhất của điểm dữ liệu i với các điểm dữ liệu của từng cụm khác. Hai chỉ số trên càng nhỏ thì mức độ tương đồng của i với cụm dữ liệu đó càng nhau.

Khi đó diểm Silhouette được tính như sau:

$$s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \dfrac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases}$$

2.3.6 Elbow method

Elbow method là một cách heuristic để tìm số cụm trong một tập dữ liệu[11]. Phương pháp làm như sau:

1. Tính Inertia của dữ liệu:

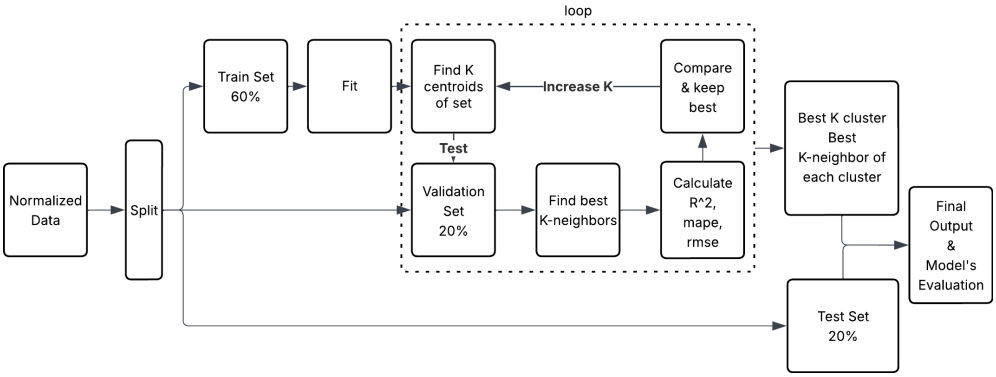
$$I = \sum_{i=1}^{n} \sum_{j \in C_j} d(j, A_{C_j})^2$$

   với C_j là cụm dữ liệu và A_{C_j} là tâm của cụm
2. Tìm điểm mà tốc độ giạm của inertia giảm đột ngột và gọi điểm đó là điểm Elow

Tuy nhiên phương pháp này gây tranh cãi vì định nghĩa của điểm Elbow tùy thuộc và người sử dụng.

3 Mô hình dự đoán giá nhà đất

3.1 Phương pháp



Hình 2 Sơ đồ tổng quát kiến trúc mô hình

Hình 2 minh họa quy trình công tác (pipeline) vận hành của kiến trúc mô hình dự đoán giá nhà đất, bao gồm: tìm trung tâm và phân cụm các mẫu dữ liệu nhà đất trong tập huấn luyện, thực hiện vòng lặp trên tập xác thực để tìm số cụm tối ưu, đồng thời tìm số K-nearest neighbor tối ưu cho mỗi cụm, thông qua các phương pháp đánh giá như $\mathbf{R^2}$, **MAPE**, **RMSE**, **MAE**, **Silhouette**, sử dụng giá trị K tìm được ở bước cuối làm tham số cho mô hình dự đoán giá nhà đất hoàn chỉnh và đánh giá mô hình máy học hoàn chỉnh trên tập kiểm tra.

3.1.1 K-means

K-means là một phương pháp *học máy không giám sát*, được sử dụng trong nghiên cứu này nhằm hỗ trợ và nâng cao hiệu quả của mô hình KNN, đồng thời giảm ảnh hưởng của các điểm dữ liệu ngoại lệ trong quá trình tìm kiếm láng giềng. Thuật toán hoạt động bằng cách nhóm và chia các mẫu dữ liệu nhà đất gần nhau thành một số hữu hạn các cụm, và chỉ thực hiện thuật toán KNN để dự đoán một mẫu dữ liệu dựa trên những mẫu dữ liệu thuộc cùng cụm với nó.

Quá trình phân cụm được thực hiện bằng cách khởi tạo K tâm cụm ban đầu, sau đó tiến hành các vòng lặp cập nhật để các tâm cụm dần hội tụ về một cấu trúc phân cụm ổn định, qua đó hỗ trợ hiệu quả cho các bước phân tích và mô hình hóa dữ liệu giá nhà đất tiếp theo.

3.1.2 KNN

Thuật toán KNN là một phương pháp học máy thuộc nhóm *lazy learning*, được biết đến rộng rãi nhờ cơ chế suy luận đơn giản và thời gian huấn luyện gần như bằng 0.

Phương pháp này hoạt động bằng cách lựa chọn K láng giềng gần nhất trong tập huấn luyện đối với điểm cần dự đoán, sau đó tổng hợp nhãn của điểm đó dựa trên các mẫu lân cận. Trong nghiên cứu này, bài toán được xét dưới dạng hồi quy, theo đó thuật toán KNN được triển khai ở chế độ hồi quy, với giá trị đầu ra được ước lượng thông qua trung bình cộng giá nhà đất của các mẫu lân cận.

Tuy nhiên, KNN cũng tồn tại một số hạn chế. Mặc dù chi phí huấn luyện không đáng kể, thời gian suy luận lại tăng lên đáng kể khi kích thước tập dữ liệu hoặc số K lớn. Ngoài ra, khi K nhỏ, mô hình trở nên nhạy cảm với nhiễu, dễ dẫn đến các dự đoán sai lệch.

Do đó, mô hình đề xuất trong nghiên cứu này kết hợp K-means và KNN nhằm phân nhóm dữ liệu giá nhà đất trước, sau đó chỉ thực hiện dự đoán KNN trong phạm vi từng cụm, qua đó hạn chế ảnh hưởng của nhiễu và cải thiện hiệu quả dự báo.

3.1.3 Phương pháp tìm tham số K cho số cụm

Trong bài toán phân cụm, việc lựa chọn số cụm K có ý nghĩa quan trọng đến tính chính xác của những suy luận của mô hình. Đặc biệt với những tập dữ liệu phức tạp và có kích thước không xác định, việc chọn một số cụm K cố định có thể dẫn đến sự thiếu cơ sở trong nhiều trường hợp. Vì vậy, nghiên cứu nên có phương pháp rõ ràng và khoa học để có thể tìm được cách chia cụm cho ra kết quả với nền tảng rõ ràng.

Để giải quyết vấn đề này, nhóm chúng tôi đề xuất một phương pháp tìm số cụm K, nhằm mục đích phản ánh hợp lý mẫu dữ liệu nhà đất trên phương diện suy luận và dự đoán. Thay vì giả định một số K từ trước, phương pháp tìm cụm của chúng tôi sẽ trực tiếp tính kết quả giá nhà đất đối với các giá trị K khác nhau trong một khoảng cho sẵn, từ đó tìm được lựa chọn phản ánh mức độ phân cụm thỏa mãn các tiêu chí đánh giá nhất quán.

Về cơ bản, phương pháp tìm cấu hình phù hợp có ưu điểm là sẽ thể hiện được các cấu trúc tiềm ẩn của dữ liệu nhà đất, đồng thời làm giảm tính chủ quan trong việc chọn giá trị của K. Nhưng mặt khác, cách tiếp cận này còn phụ thuộc nhiều vào tiêu chí đánh giá, dẫn đến sự không nhất quán về kết quả nếu dựa trên các tiêu chí khác nhau. Bên cạnh đó, việc thử với nhiều giá trị K khác nhau sẽ làm tăng thời gian tính toán lên nhiều lần, đặc biệt là với các tập dữ liệu kích thước lớn.

3.1.4 Phương pháp tìm tham số K láng giềng của từng cụm

Sau khi dữ liệu nhà đất được phân chia thành các cụm riêng biệt, mỗi cụm được xem như một không gian con có đặc trưng phân bố khác nhau về mật độ, độ nhiễu và quy mô mẫu. Do đó, việc sử dụng cùng một giá trị K cho thuật toán KNN trên toàn bộ dữ liệu có thể không còn phù hợp trong bối cảnh này.

Đối với mỗi cụm, thuật toán KNN được áp dụng độc lập với các giá trị K khác nhau trong một khoảng xác định trước. Quá trình đánh giá được thực hiện nội bộ trong từng cụm, dựa trên khả năng dự đoán của mô hình đối với các mẫu thuộc chính

cụm đó. Việc so sánh các giá trị K cho phép quan sát ảnh hưởng của số láng giềng đến độ ổn định và độ chính xác của kết quả dự đoán trong từng không gian cụm cụ thể.

Giá trị K được lựa chọn cho mỗi cụm là giá trị thể hiện hiệu năng dự đoán phù hợp và ổn định nhất trong phạm vi cụm tương ứng. Cách tiếp cận này cho phép mô hình KNN thích nghi linh hoạt với đặc điểm cục bộ của từng cụm, đồng thời khai thác hiệu quả hơn thông tin xu hướng giá nhà đất đã được tạo ra từ bước phân cụm trước đó.

3.2 Thực nghiệm

Phần này trình bày chi tiết quy trình xây dựng mô hình, bắt đầu từ việc phân tích đặc điểm tập dữ liệu, thực hiện các bước tiền xử lý cần thiết để đảm bảo tính nhất quán, và cuối cùng là thiết lập các tham số cho mô hình dự đoán.

3.2.1 Phân tích dữ liệu và tiền xử lý

Nghiên cứu sử dụng tập dữ liệu bao gồm 414 mẫu quan sát được trích xuất từ các giao dịch bất động sản thực tế. Mỗi mẫu dữ liệu ban đầu bao gồm một thuộc tính định danh (No) và 7 đặc trưng, trong đó có 6 đặc trưng đầu vào và 1 biến mục tiêu (giá nhà). Chi tiết các biến được trình bày trong Bảng 2.

Bảng 2 Mô tả các đặc trưng trong tập dữ liệu

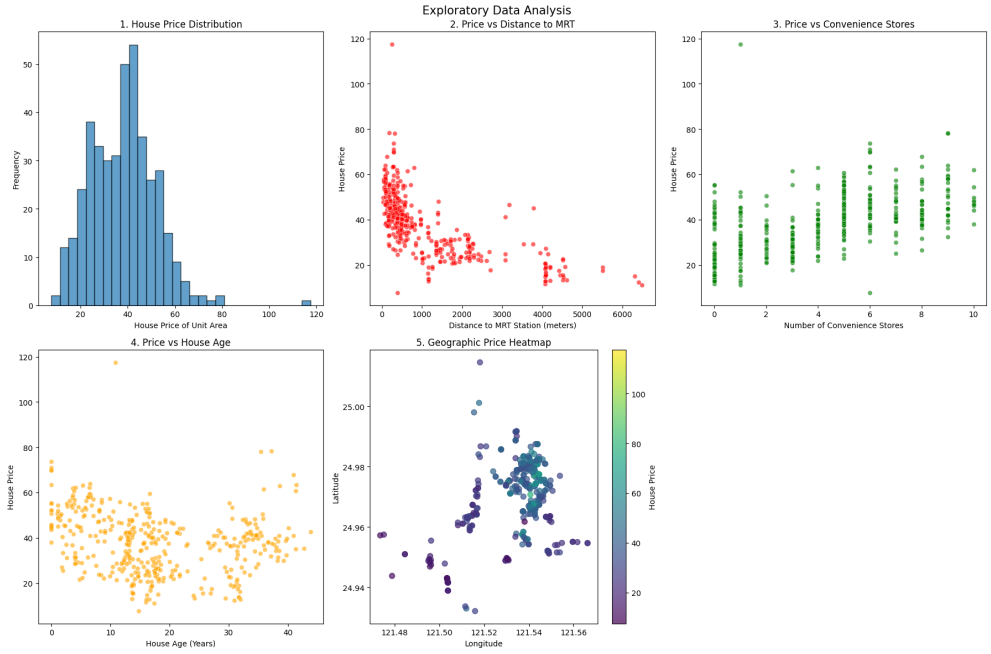
Ký hiệu	Tên đặc trưng	Ý nghĩa	Kiểu dữ liệu
X1	transaction_date	Ngày giao dịch	Float
X2	house_age	Tuổi đời của căn nhà (năm)	Float
X3	distance_to_MRT	Khoảng cách tới trạm MRT gần nhất (m)	Float
X4	convenience_stores	Số lượng cửa hàng tiện ích xung quanh	Integer
X5	latitude	Vĩ độ địa lý	Float
X6	longitude	Kinh độ địa lý	Float
Y	house_price	Giá trị căn nhà trên một đơn vị diện tích	Float

Để có cái nhìn sâu sắc về cấu trúc dữ liệu và mối quan hệ giữa các đặc trưng, kỹ thuật "Phân tích dữ liệu khám phá" (Exploratory Data Analysis - EDA) được áp dụng thông qua các phương pháp trực quan hóa sau:

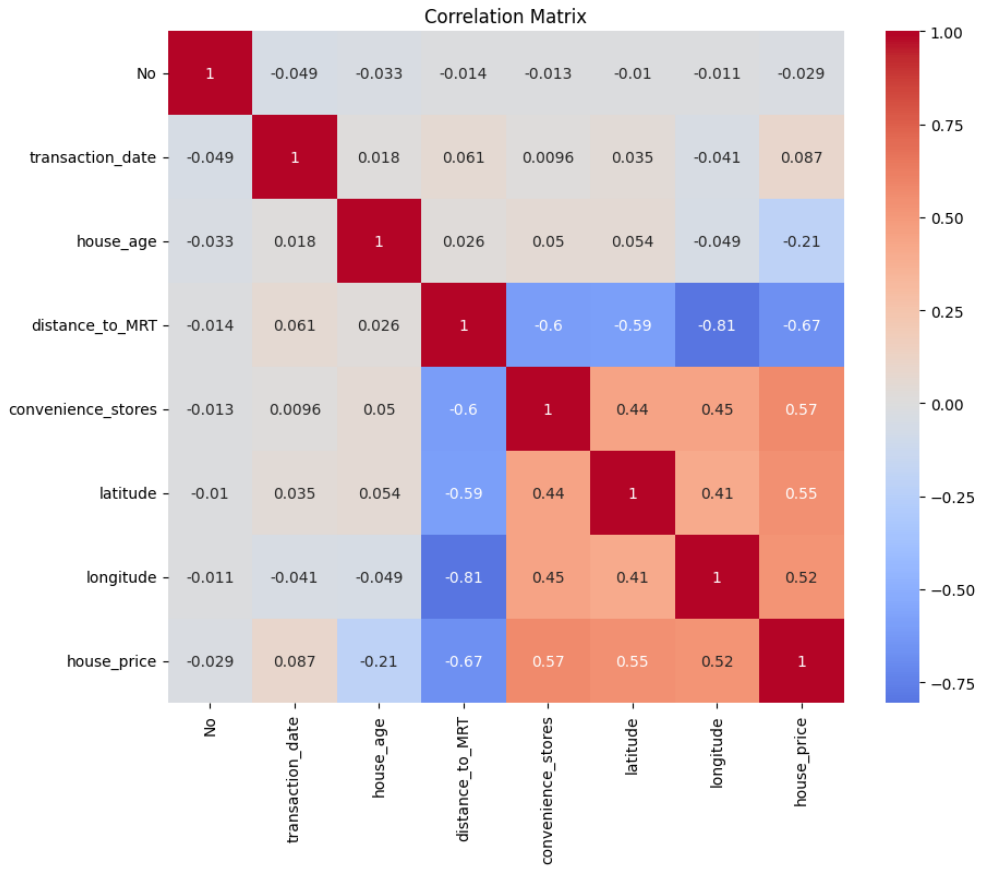
- **Biểu đồ phân phối (Histogram):** Quan sát tần suất xuất hiện của các giá trị trong biến mục tiêu nhằm nhận diện dạng phân phối (chuẩn hay lệch) và phát hiện các giá trị ngoại lai (outliers).
- **Biểu đồ tán xạ (Scatter Plot):** Biểu diễn mối tương quan giữa từng đặc trưng đầu vào (như khoảng cách đến MRT, tuổi đời căn nhà) với biến mục tiêu, qua đó nhận định sơ bộ về xu hướng tác động và độ mạnh yếu của mối quan hệ.
- **Bản đồ nhiệt địa lý (Geographic Heatmap):** Tận dụng hai đặc trưng Vĩ độ và Kinh độ để mô phỏng sự phân bố không gian của giá nhà, giúp phát hiện các cụm khu vực có giá trị bất động sản cao/thấp đặc thù.

- **Ma trận tương quan (Correlation Matrix):** Sử dụng hệ số tương quan Pearson để định lượng mức độ phụ thuộc tuyến tính giữa các cặp đặc trưng, hỗ trợ việc lựa chọn đặc trưng quan trọng và loại bỏ hiện tượng đa cộng tuyến.

Dưới đây là một số đồ thị được chúng tôi vẽ ra trong quá trình khảo sát tính chất của tập dữ liệu:



Hình 3 Biểu đồ phân tích dữ liệu khám phá (EDA)



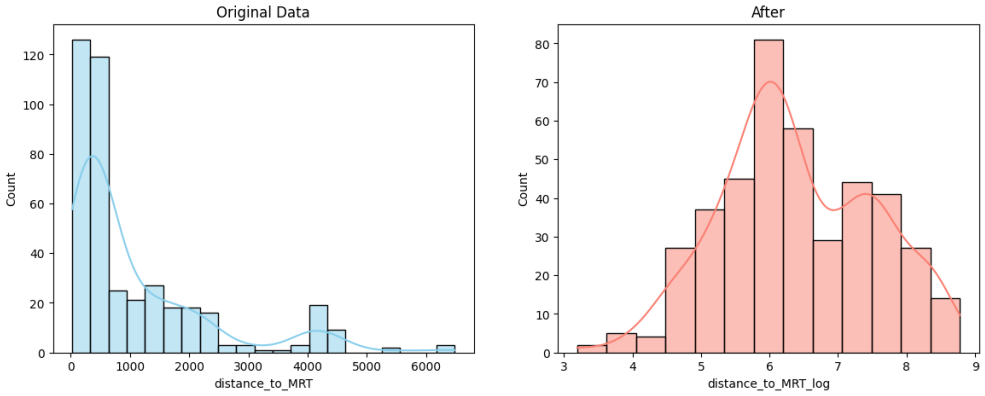
Hình 4 Ma trận tương quan (Correlation Matrix)

Dựa trên các phân tích trực quan, một số nhận định quan trọng được rút ra như sau:

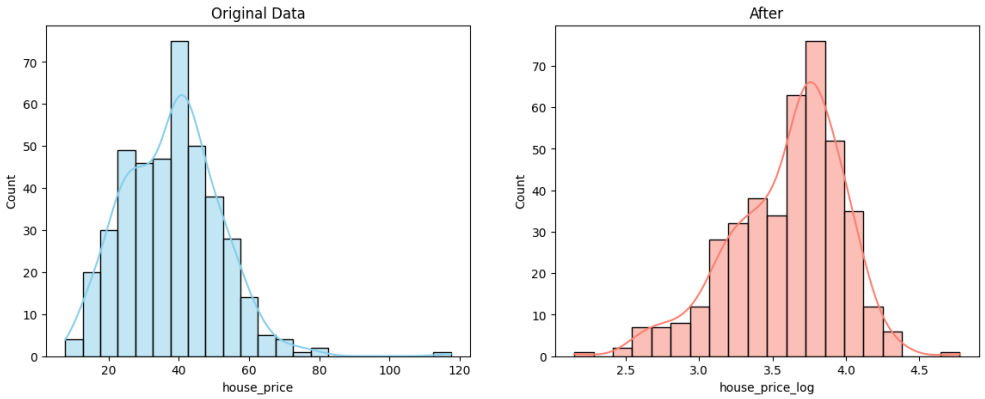
- Thuộc tính số thứ tự (No) và ngày giao dịch (transaction_date) có hệ số tương quan Pearson gần bằng 0 đối với biến mục tiêu, cho thấy chúng không mang nhiều ý nghĩa dự báo.
- Biểu đồ phân phối của biến mục tiêu cho thấy hiện tượng lệch phải (right-skewed) với sự xuất hiện của các điểm dữ liệu cực cao.
- Đặc trưng *distance_to_MRT* có tương quan nghịch mạnh nhất (-0.67) với giá nhà theo quy luật phi tuyến tính (giá giảm nhanh ở khoảng cách gần và bão hòa ở khoảng cách xa). Ngược lại, *convenience_stores* thể hiện tương quan thuận khá mạnh (0.57).
- Đặc trưng vị trí địa lý (Kinh độ, Vĩ độ) cho thấy sự phân cụm rõ rệt của các khu vực giá cao, gợi ý vai trò quan trọng của yếu tố không gian.

Từ những nhận định trên, quy trình tiền xử lý được thực hiện bao gồm các bước:

- **Lựa chọn đặc trưng:** Loại bỏ thuộc tính số thứ tự (No) và đặc trưng ngày giao dịch (transaction_date) do không đóng góp vào giá nhà.
- **Biến đổi dữ liệu:** Áp dụng hàm $\log(1+x)$ cho đặc trưng khoảng cách đến trạm MRT gần nhất (distance_to_MRT) để tuyến tính hóa biểu đồ tán xạ, cũng như áp dụng hàm $\log(1+x)$ cho biến mục tiêu (giá nhà) để đưa phân phối về dạng gần chuẩn. Điều này giúp ổn định phương sai, thu hẹp khoảng cách giữa các giá trị cực đại, hỗ trợ các thuật toán hoạt động hiệu quả hơn.



Hình 5 Phân phối khoảng cách đến MRT trước và sau khi biến đổi Log



Hình 6 Phân phối giá nhà trước và sau khi biến đổi Log

- **Xử lý ngoại lai:** Loại bỏ các điểm dữ liệu bất thường sử dụng phương pháp IQR (Interquartile Range).

- **Phân chia dữ liệu:** Tập dữ liệu được chia ngẫu nhiên theo tỷ lệ 80% cho huấn luyện (training set) và 20% cho kiểm tra (test set). Việc chia tập diễn ra ngẫu nhiên để đảm bảo phân bố dữ liệu đồng đều.
- **Chuẩn hóa:** Sử dụng phương pháp Min-Max Scaling để đưa các đặc trưng về cùng khoảng giá trị $[0, 1]$, triệt tiêu sự chênh lệch về đơn vị đo lường. Các tham số chuẩn hóa được tính toán trên tập huấn luyện và áp dụng cho tập kiểm tra.

Bảng 3 Tóm tắt các thông số sau tiền xử lý

Thông số	**Giá trị**
Tổng số mẫu sau khi lọc nhiễu	409
Số đặc trưng đầu vào	5
Tỷ lệ chia Tập huấn luyện/Kiểm tra	80/20
Phương pháp chuẩn hóa	Min-Max Scaling

3.2.2 Xây dựng mô hình dự đoán

Nghiên cứu triển khai thực nghiệm trên hai phương pháp tiếp cận chính dựa trên thuật toán láng giềng gần nhất. Chi tiết mã nguồn thực hiện được cung cấp trong các tệp `source.ipynb` và `source.py` đính kèm.

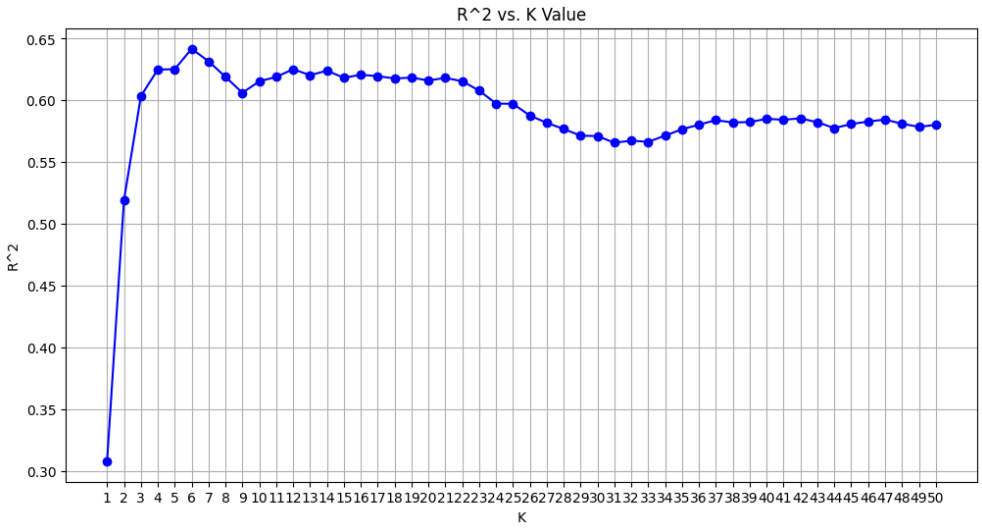
Phương pháp A: KNN Regressor (K-Nearest Neighbors)

Mô hình hồi quy KNN được xây dựng thông qua lớp `KNN_regressor`. Dưới đây là đoạn mã mô tả quá trình dự đoán dựa trên K láng giềng gần nhất sử dụng khoảng cách Euclid:

```
def predict(self, X):
    X = np.array(X)
    predictions = []
    for x in X:
        distances = euclidean_distance(self.X, x)
        k_idx = np.argsort(distances)[:min(self.k, len(self.X))]
        predictions.append(np.mean(self.y[k_idx]))
    return np.array(predictions)
```

Listing 1 Đoạn mã dự đoán của lớp KNN Regressor

Yếu tố cốt lõi quyết định hiệu suất của phương pháp này là việc xác định tham số K (số lượng láng giềng) tối ưu. Quy trình tìm kiếm tham số (grid search) được thực hiện với K chạy từ 1 đến 50 trên tập xác thực (validation set - được tách ra từ 20% tập huấn luyện).



Hình 7 Biểu đồ thể hiện sự thay đổi của R^2 theo giá trị K

Dựa vào biểu đồ ở Hình 7, giá trị $\mathbf{K = 6}$ được lựa chọn cho mô hình cuối cùng do đạt hiệu suất tối ưu trên tập xác thực.

Phương pháp B: Mô hình lai K-Means + KNN

Đây là hướng tiếp cận nâng cao nhằm khai thác cấu trúc cục bộ của dữ liệu theo chiến lược "chia để trị". Quy trình bao gồm hai giai đoạn:

- **Giai đoạn 1: Phân cụm dữ liệu.** Thuật toán K-Means được sử dụng để nhóm các mẫu dữ liệu có đặc tính tương đồng (vị trí, tiện ích) vào các cụm riêng biệt. Thuật toán hoạt động bằng cách khởi tạo ngẫu nhiên các tâm cụm và lặp lại quá trình gán nhãn - cập nhật tâm:

```
def fit_loop(self, minval=1e-6, max_iter=300):
    self.choose_random()
    for _ in range(max_iter):
        old_centers = self.centers.copy()
        self.update_centers()
        shift = euclidean_distance(
            old_centers.flatten()[None, :],
            self.centers.flatten()
        )[0]
        if shift < minval:
            break
```

Listing 2 Cài đặt vòng lặp huấn luyện K-Means

Số lượng cụm tối ưu ($K_{cluster}$) được xác định dựa trên kết quả tính toán thực nghiệm của chỉ số quán tính (Inertia) và chỉ số dáng điệu (Silhouette Score).
Giá trị Inertia được tính toán trực tiếp thông qua phương thức nội bộ trong lớp K-Means:

```
def calculate_inertia(self):
    inertia = 0.0
    for i, x in enumerate(self.X):
        c = self.labels[i]
        inertia += np.sum((x - self.centers[c]) ** 2)
    return inertia
```

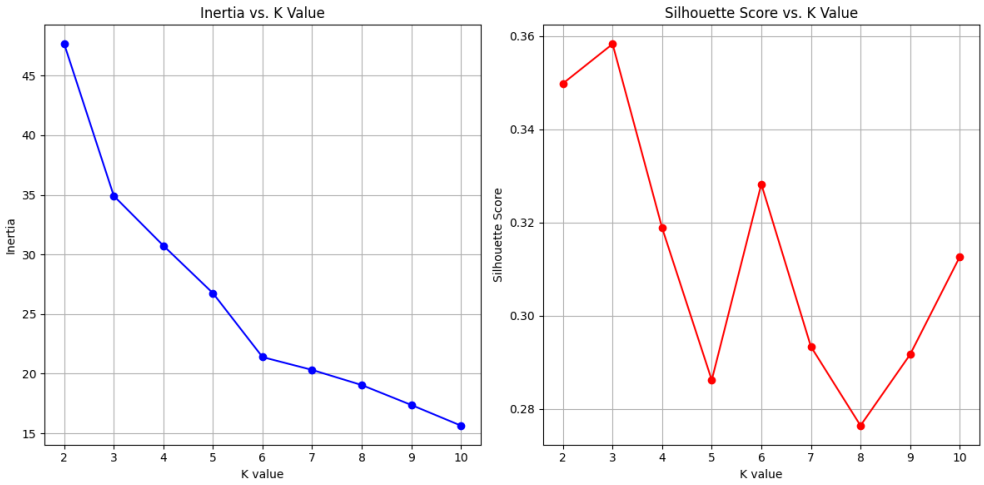
Listing 3 Phương thức tính Inertia trong lớp K-Means

Đồng thời, chỉ số Silhouette được tính toán trên toàn bộ tập dữ liệu để đánh giá chất lượng phân tách của các cụm:

```
def silhouette_score(X, labels, metric='euclidean'):
    max_ab = np.maximum(A, B)
    s_scores = np.zeros(n_samples)
    valid_mask = max_ab > 0
    s_scores[valid_mask] = (B[valid_mask] - A[valid_mask]) / max_ab[valid_mask]
    return np.mean(s_scores)
```

Listing 4 Hàm tính chỉ số Silhouette

Kết quả thực nghiệm với K chạy từ 2 đến 10 được trực quan hóa tại Hình 8.



Hình 8 Biểu đồ quán tính (Inertia) và chỉ số Silhouette theo số lượng cụm K

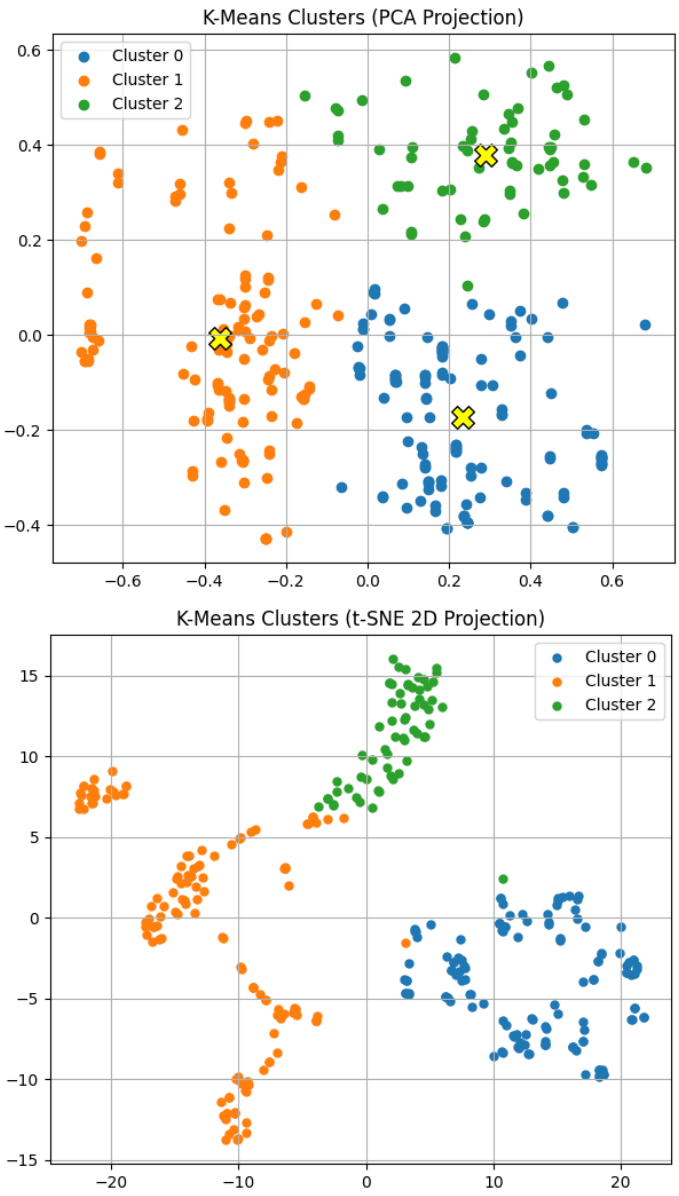
Dựa trên biểu đồ, các phân tích sau được đưa ra:

– **Phương pháp Elbow (Biểu đồ trái):** Một sự sụt giảm mạnh về giá trị Inertia được ghi nhận trong khoảng K từ 2 đến 3. Tại $K = 3$, độ dốc của đường cong

bắt đầu giảm đáng kể, hình thành điểm uốn ("khuỷu tay"), báo hiệu sự bão hòa về độ nén của cụm.

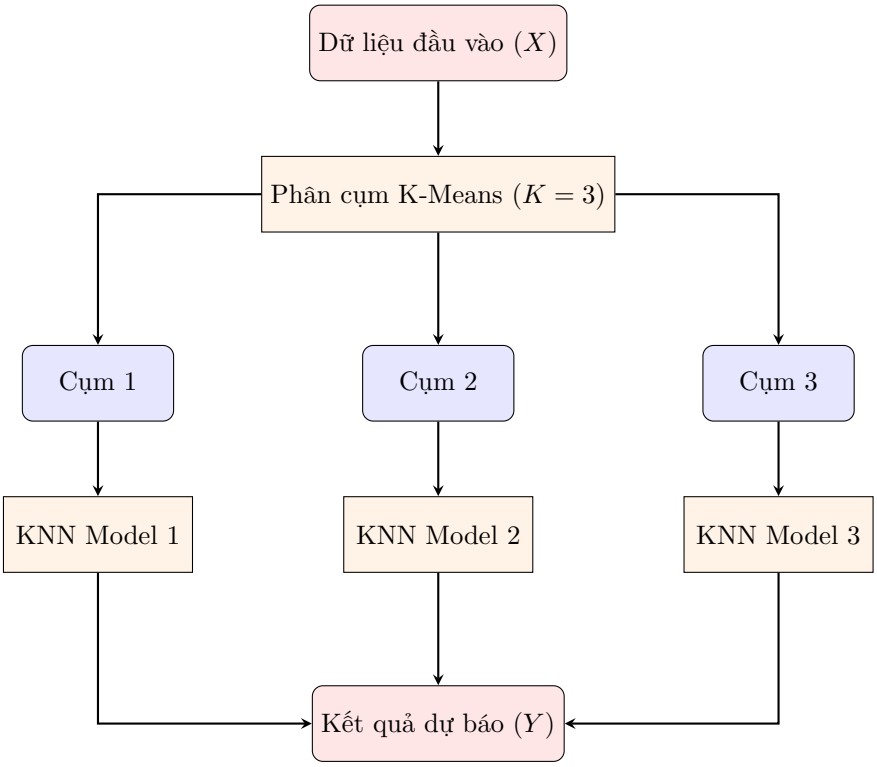
– **Chỉ số Silhouette (Biểu đồ phải):** Giá trị Silhouette đạt cực đại (xấp xỉ 0.36) tại $K = 3$. Các giá trị K khác đều cho kết quả thấp hơn, cho thấy tại $K = 3$, cấu trúc phân cụm đạt được sự tách biệt rõ ràng nhất.

Từ sự đồng thuận giữa hai phương pháp đánh giá, giá trị $\mathbf{K = 3}$ được lựa chọn làm tham số tối ưu cho mô hình. Kết quả phân cụm cuối cùng được minh họa trực quan tại Hình 9.



Hình 9 Trực quan hóa các cụm dữ liệu sau khi áp dụng K-Means

- **Giai đoạn 2: Dự báo cục bộ.** Với mỗi cụm được hình thành, một mô hình KNN riêng biệt được xây dựng và huấn luyện. Cách tiếp cận này cho phép mô hình thích nghi tốt hơn với quy luật giá nhà tại từng phân khúc thị trường đặc thù (ví dụ: khu vực trung tâm sầm uất so với khu vực ngoại ô).



Hình 10 Sơ đồ quy trình hoạt động của mô hình lai K-Means + KNN

3.3 Kết quả và đánh giá chung

3.3.1 Kết quả thực nghiệm định lượng

Sau khi hoàn tất quá trình huấn luyện và tối ưu tham số, hiệu suất của các mô hình được đánh giá trên tập kiểm tra độc lập (test set). Các chỉ số đo lường được tổng hợp trong Bảng 4.

Bảng 4 So sánh các chỉ số hiệu suất của hai phương pháp

Chỉ số đánh giá	**KNN Regressor**	**K-Means + KNN**
R2 Score (càng cao càng tốt)	0.7902	**0.7957**
MAE (càng thấp càng tốt)	4.2213	**4.2193**
RMSE (càng thấp càng tốt)	5.6574	**5.5819**
MAPE (càng thấp càng tốt)	**0.1328**	0.1358

3.3.2 Phân tích kết quả

Dựa trên số liệu thực nghiệm, một số phân tích và đánh giá được đưa ra như sau:

- **Về độ chính xác tổng thể:** Cả hai phương pháp đều thể hiện khả năng dự báo khả quan với hệ số xác định R^2 đạt xấp xỉ 0.8. Điều này chứng minh rằng quy trình tiền xử lý dữ liệu (đặc biệt là biến đổi logarit và loại bỏ ngoại lai) đã giúp cải thiện đáng kể chất lượng dữ liệu đầu vào.
- **So sánh hiệu quả giữa hai phương pháp:** Mô hình lai **K-Means + KNN** cho thấy ưu thế vượt trội hơn ở hầu hết các chỉ số quan trọng. Cụ thể, R^2 tăng lên mức 0.7957 và RMSE giảm xuống còn 5.5819 so với mô hình KNN đơn lẻ. Việc chỉ số RMSE giảm phản ánh rằng phương pháp lai ít gặp phải các sai số dự báo lớn, nhờ vào khả năng phân tách và xử lý dữ liệu theo từng đặc thù nhóm riêng biệt.
- **Về tính ổn định:** Mặc dù mô hình lai có độ chính xác cao hơn, mô hình KNN thuần túy vẫn duy trì ưu thế về chỉ số MAPE thấp nhất (0.1328). Điều này cho thấy trong một số trường hợp, sự đơn giản của mô hình toàn cục vẫn mang lại độ ổn định tương đối tốt trên toàn bộ tập dữ liệu.

Tóm lại, trong phạm vi thực nghiệm của nghiên cứu này, phương pháp kết hợp **K-Means + KNN** đã chứng minh được tính hiệu quả cao hơn trong việc giải quyết bài toán dự đoán giá nhà đất phức tạp.

4 Thảo luận

4.1 Tác động

Việc triển khai các mô hình dự báo giá bất động sản trong bài làm này không chỉ dừng lại ở bài toán hồi quy đơn thuần mà còn mở ra cái nhìn sâu sắc về cấu trúc dữ liệu thị trường.

- **Về mặt kỹ thuật:** Bài làm cho thấy việc ứng dụng các mô hình cơ bản như **K-Nearest Neighbors (KNN)** có thể tạo ra nền tảng dự báo ổn định nhờ tận dụng tính tương đồng về đặc điểm địa lý và thuộc tính nhà ở.
- **Về cải tiến:** Việc ứng dụng thêm **K-Means** vào mô hình **KNN** cho ta một kết quả tốt hơn, thấy được tác động tích cực của việc phân cụm dữ liệu trước tính toán. Việc dùng K-Means để phân các "clusters" trong đó các dữ liệu có tính tương đồng với nhau, sau đó sử dụng mô hình KNN để tiến hành dự đoán trên khoảng không gian tập trung hơn, điều này giúp giảm thiểu việc nhiễu, phản ánh chính xác hơn các biến động cục bộ của thị trường.
- **Ứng dụng:** Mô hình cung cấp công cụ hỗ trợ đưa ra một cái nhìn khách quan cho các nhà đầu tư và người mua nhà. Việc ứng dụng trên tập dữ liệu trước đó để đưa ra dự đoán cho người dùng một cách có căn cứ hơn, dựa vào đây người dùng có thể có được cái nhìn khái quát về biến động của thị trường và xây dựng được chiến lược cụ thể.

4.2 Hạn chế

Mặc dù việc kết hợp 2 mô hình **KNN** và **K-Means** mang lại kết quả khả quan, hệ thống vẫn tồn tại những rào cản kỹ thuật cần được khắc phục trong tương lai:

- **Độ chính xác:** Do thị trường bất động sản chịu ảnh hưởng mạnh mẽ bởi các yếu tố phi cấu trúc (chính sách kinh tế, tâm lý đám đông, quy hoạch đô thị), các mô hình dựa thuần túy trên dữ liệu lịch sử như KNN vẫn chưa thể đạt được độ chính xác tuyệt đối trong các giai đoạn thị trường biến động mạnh.
- **Tính ổn định:** Cả hai phương pháp đều phụ thuộc lớn vào việc lựa chọn tham số K (số láng giềng) và số lượng cụm trong K-Means. Việc chọn sai tham số có thể dẫn đến hiện tượng quá khớp (overfitting) hoặc dưới khớp (underfitting).
- **Dữ liệu đầu vào:** Mô hình hiện tại chủ yếu tập trung vào các biến định lượng. Việc thiếu vắng các dữ liệu định tính (như uy tín chủ đầu tư, phong thủy, hoặc tiện ích hạ tầng ngầm) phần nào hạn chế khả năng tiệm cận giá trị thực của mô hình.

5 Đề xuất mô hình dự đoán khác

Qua phần trình bày trên đã trình bày về phương pháp dự đoán giá nhà đất mà chúng tôi đã thực hiện công việc này thông qua việc sử dụng mô hình K-nearest neighbor (KNN) và mô hình phân cụm K-means, bên cạnh đó chúng tôi có tìm hiểu thêm được nhiều mô hình phù hợp khác đối với bài toán dự đoán. Chúng tôi qua việc tìm hiểu các mô hình ngoài hai mô hình đã được ứng dụng chính trong bài làm, đã có đề xuất thêm mô hình dự đoán đối với bài toán dự đoán giá nhà trên là dùng mô hình **Linear Regression**.

Trước khi đi vào việc ứng dụng mô hình vào bài toán dự đoán này, chúng tôi đề cập qua kiến thức về **Linear Regression** sau đây.

5.1 Mô hình Linear Regression

Phần giới thiệu mô hình này được trích lược từ [12].

5.1.1 Giới thiệu

Hồi quy tuyến tính (Linear Regression) là thuật toán hồi quy mà đầu ra là một hàm số tuyến tính của đầu vào. Đây là thuật toán đơn giản nhất trong số thuật toán học có giám sát.

Trong phần lý thuyết này chúng tôi sẽ lấy minh hoạ một vài đặc trưng trên bài toán mà chúng tôi đang thực hiện. Ta có một căn nhà có khoảng cách đến ga tàu điện là x_1 m^2, có x_2 số cửa hàng tiện lợi gần đó và căn nhà có tuổi là x_3. Với một lượng dữ liệu về căn nhà với các đặc trưng đã nêu như trên thì ta có thể xây dựng được hàm dự đoán $y = f(\mathbf{x})$. Ta có được $\mathbf{x} = [x_1, x_2, x_3]^T$ là một vector cột chứa chứa dữ liệu đầu vào *input*, với y là một đầu ra là một số thực dương.

Dễ thấy rằng giá nhà sẽ cao nếu các đặc trưng của ngôi nhà là tối ưu nhất, ví dụ như khoảng cách đến ga tàu điện là ngắn nhất, nhiều cửa hàng tiện lợi nhất, căn nhà còn mới,...v.v. Ta có thể mô hình hoá được đầu ra đơn giản như sau

$$y \approx \hat{y} = f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 = \mathbf{x}^T \mathbf{w} \tag{1}$$

trong đó $\mathbf{w} = [w_1, w_2, w_3]^T$ là các vector trọng số cần tìm. Mối quan hệ của (1) là mối quan hệ tuyến tính.

5.1.2 Tổng quát

Với mỗi điểm dữ liệu được mô tả với d đặc trưng khác nhau, ta có thể mô tả dưới dạng tổng quát rằng điểm dữ liệu đó được đặc trưng bởi vector d chiều nằm trong không gian $\mathbb{R}^d$ hay $x \in \mathbb{R}^d$

$$y \approx \hat{y} = f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 = \mathbf{x}^T \mathbf{w} \tag{2}$$

5.1.3 Sai số dự đoán

Ta cần đánh giá phù hợp với bài toán yêu cầu sau khi hoàn thành việc xây dựng mô hình dự đoán (2). Với các bài toán hồi quy, mong muốn sự sai khác e giữa đầu ra thực tế y và đầu ra dự đoán $\hat{y}$ là giá trị nhỏ nhất.

$$\frac{1}{2} e^2 = \frac{1}{2} (y - \hat{y})^2 = \frac{1}{2} \left(y - \mathbf{x}^T \mathbf{w}\right)^2 \tag{3}$$

Ta lấy e^2 vì $e = y - \hat{y}$ có thể là một số âm, ta cũng có thể sử dụng hàm trị tuyệt đối $|e| = |y - \hat{y}|$ để mô tả nhưng vì hàm trị tuyệt đối không khả vi tại $(0, 0)$ không thuận tiện cho việc tối ưu. Hệ số $\frac{1}{2}$ sẽ bị triệt tiêu khi lấy đạo hàm của e theo tham số của mô hình $\mathbf{w}$.

5.1.4 Hàm mất mát

Với mọi cặp dữ liệu $(\mathbf{x}_i, y_i)$, $i = 1, 2, ..., N$, N là số lượng dữ liệu trong tập huấn luyện, việc tìm mô hình tốt nhất đồng nghĩa với tìm $\mathbf{w}$ sao cho hàm số dưới đây đạt giá trị là nhỏ nhất

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \mathbf{x}_i^T \mathbf{w}\right)^2 \tag{4}$$

Hàm số $\mathcal{L}(\mathbf{w})$ là hàm mất mát của mô hình hồi quy tuyến tính với tham số $\theta = \mathbf{w}$. Ta mong muốn sự mất mát là nhỏ nhất, điều này có thể đạt được bằng cách tối thiểu hàm mất mát theo $\mathbf{w}$:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}). \tag{5}$$

$\mathbf{w}^*$ là nghiệm cần phải tìm của bài toán. Dấu $*$ có thể bỏ đi và nghiệm có thể viết lại thành $\mathbf{w} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$.

Ta có thể viết gọn lại (4) dưới dạng ma trận, vector, norm như sau:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \mathbf{x}_i^T\mathbf{w}\right)^2 = \frac{1}{2N}\left\|\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}\mathbf{w}\right\|_2^2 = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}^T\mathbf{w}\|_2^2 \quad (6)$$

với $\mathbf{y} = [y_1, y_2, ..., y_N]^T$, $\mathbf{X} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N]$. Như vậy $\mathcal{L}(\mathbf{w})$ là một hàm số liên tục liên quan đến bình phương của ℓ_2 norm.

5.1.5 Nghiệm của hồi quy tuyến tính

Nhận thấy rằng $\mathcal{L}(\mathbf{w})$ luôn có gradient tại mọi $\mathbf{w}$. Giá trị tối ưu của w có thể tìm được thông qua việc giải phương trình đạo hàm của $\mathcal{L}(\mathbf{w})$ theo $\mathbf{w}$ bằng không.

$$\frac{\nabla\mathcal{L}(\mathbf{w})}{\nabla\mathbf{w}} = \frac{1}{N}\mathbf{X}\left(\mathbf{W}^T\mathbf{w} - \mathbf{y}\right) \quad (7)$$

Phương trình gradient bằng không:

$$\frac{\nabla\mathcal{L}(\mathbf{w})}{\nabla\mathbf{w}} = 0 \Leftrightarrow \mathbf{X}\mathbf{X}^T\mathbf{w} = \mathbf{X}\mathbf{y} \quad (8)$$

Nếu ma trận $\mathbf{X}\mathbf{X}^T$ khả nghịch thì phương trình (8) có nghiệm duy nhất $\mathbf{w} = \left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{y}$.

Ngược lại, nếu ma trận $\mathbf{X}\mathbf{X}^T$ không khả nghịch thì phương trình (8) vô nghiệm hoặc vô số nghiệm. Lúc này, có thể xác định được nghiệm đặc biệt của phương trình dựa vào *giả nghịch đảo*. Người ta chứng minh với mọi ma trận $\mathbf{X}$, luôn tồn tại duy nhất giá trị $\mathbf{w}$ có ℓ_2 norm nhỏ nhất giúp tối thiểu $\left\|\mathbf{X}^T\mathbf{w} - \mathbf{y}\right\|_F^2$. Cụ thể, $\left(\mathbf{X}\mathbf{X}^T\right)^\dagger\mathbf{X}\mathbf{y}$ trong đó $\left(\mathbf{X}\mathbf{X}^T\right)^\dagger$ là giả nghịch đảo của $\left(\mathbf{X}\mathbf{X}^T\right)$. Giả nghịch đảo của một ma trận luôn tồn tại. Đối với ma trận vuông và khả nghịch, giả nghịch đảo là nghịch đảo của ma trận đó. Tổng quát, nghiệm tối ưu của bài toán tối ưu (5) là

$$\mathbf{w} = \left(\mathbf{X}\mathbf{X}^T\right)^\dagger\mathbf{X}\mathbf{y} \quad (9)$$

5.1.6 Hệ số điều chỉnh

Hàm dự đoán đầu ra của hồi quy tuyến tính thường có thêm một *hệ số điều chỉnh* (bias) b:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b \quad (10)$$

Nếu $b = 0$, đường thẳng/mặt phẳng $\mathbf{y} = \mathbf{x}^T\mathbf{w} + b$ luôn đi qua gốc toạ độ. Việc thêm hệ số b khiến mô hình linh hoạt hơn. Hệ số điều chỉnh này cũng là một tham số mô hình.

Nếu như xem mỗi điểm dữ liệu có thêm một đặc trưng $x_0 = 1$, ta sẽ có

$$y = \mathbf{x}^T\mathbf{w} + b = w_1x_1 + w_2x_2 + ... + w_dx_d + bx_0 = \bar{\mathbf{x}}^T\bar{\mathbf{w}} \tag{11}$$

trong đó $\bar{\mathbf{x}} = [x_0, x_1, x_2, ..., x_N]^T$ và $\bar{\mathbf{w}} = [b, w_1, w_2, ..., w_N]$. Nếu đặt $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_N]$, ta có nghiệm của bài toán tối thiểu hàm mất mát

$$\bar{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{2N}\|\mathbf{y} - \bar{\mathbf{X}}^T\bar{\mathbf{w}}\|_2^2 = \left(\bar{\mathbf{X}}\bar{\mathbf{X}}^T\right)^\dagger \bar{\mathbf{X}}\mathbf{y} \tag{12}$$

Kỹ thuật thêm một đặc trưng $x_0 = 1$ vào vector đặc trưng và ghép hệ số điều chỉnh b vào vector trọng số $\mathbf{w}$ như trên còn gọi là *thủ thuật gộp hệ điều chỉnh* (bias trick).

5.2 Thử nghiệm đối với bài toán

Trong phần này, chúng tôi sẽ không đi sâu vào việc cài đặt thuật toán Linear Regression, chúng tôi sử dụng thư viện `scikit-learn` để thuận tiện hơn cho việc thực hiện bài toán. Chúng tôi sẽ tiếp tục sử dụng lại các tập dữ liệu đã được xử lý ở những phần trước đó để áp dụng cho mô hình này.

```
from sklearn.linear_model import LinearRegression
```

Tiếp đến, chúng tôi tiến hành cài đặt ngắn gọn thuật toán và thử dự đoán trên tập `X_test`

```
model = LinearRegression()
model.fit(X_train, y_train)

y_test_pred_log = model.predict(X_test)
y_test_pred = np.expm1(y_test_pred_log)
y_test_true = y_test_real
```

Để có thể đánh giá được kết quả, chúng tôi thực hiện việc tính toán với tập `y_test` ban đầu của tệp dữ liệu.

```
test_metrics = regression_metrics(y_test_true, y_test_pred)

print(test_metrics)
'''
Output:
{'R2': np.float64(0.7378927719768755), 'MAE': np.float64(4.556856531247161),
'RMSE': np.float64(6.322832202091564), 'MAPE': np.float64(0.14721592387175336)}
'''
```

Trực quan hơn, chúng tôi có cài đặt hàm để biểu diễn xem độ chính xác của việc áp dụng thuật toán hồi quy tuyến tính

```
def plot_refined_results(y_true, y_pred, ax, title, color):
    ax.scatter(y_true, y_pred, color=color, alpha=0.5, label='Actual Predicted
     Points')

    limits = [y_true.min(), y_true.max()]
    ax.plot(limits, limits, color='red', linestyle='--', linewidth=2, label='Ideal
     Line (y=x)')

    z = np.polyfit(y_true, y_pred, 1)
    p = np.poly1d(z)

    ax.plot(limits, p(limits), color='darkgreen', linestyle='-', linewidth=3,
            label=f'Model Trend (Weight-based)')

    ax.set_title(title, fontsize=16)
    ax.set_xlabel('Real Value', fontsize=12)
    ax.set_ylabel('Predicted Value', fontsize=12)
    ax.legend(loc='upper left')
    ax.grid(True, linestyle=':', alpha=0.6)
```

Áp dụng hàm bên trên vào kết quả mà ta đã nhận được bằng cách sử dụng code như sau

```
fig, ax = plt.subplots(1, 1, figsize=(15, 8))

plot_refined_results(y_test_true, y_test_pred, ax, 'Testing Set: Analysis of Model
     Weights', 'blue')

plt.tight_layout()
plt.show()
```

Sau đó, ta nhận được một hình như bên dưới đây:

Hình 11 Kết quả mô hình *hồi quy tuyến tính* sau dự đoán

5.3 Đánh giá kết quả

5.3.1 Phân tích định lượng (Regression Metrics)

Qua kết quả *output* của hàm `regression_metrics()` ở bên trên, ta rút ra được một vài đánh giá như sau:

- $R^2 \approx 0.738$: Con số này cho thấy mô hình dự đoán được khoảng 74% sự biến thiên của dữ liệu. So với mặt bằng chung của các mô hình tính toán tuyến tính cơ bản thì kết quả này ở mức độ **khá tốt**.
- $MAE \approx 4.56$: Điều này thể hiện được việc dự đoán có sai lệch trung bình so với giá trị thật khoảng 4.56 **đơn vị**.
- $RMSE \approx 6.32$: Dễ thấy được, $RMSE$ cao hơn MAE một mức đáng kể, cho thấy rằng tệp dữ liệu chưa đủ tốt hoặc các điểm đang bị dự đoán chênh lệnh lớn dẫn đến việc kéo sai số bình phương cao hơn; cho thấy phần nào độ nhiễu của mô hình hồi quy tuyến tính.
- $MAPE \approx 0.147$: Sai số trung bình rơi vào khoảng 15%, chứng tỏ mức độ tin cậy của mô hình vẫn còn có thể chấp nhận được.

5.3.2 Phân tích định tính qua biểu đồ

Quan sát đường **Model Trend** so với đường **Ideal**

- **Hiện tượng giao thoa:** Đường xu hướng của mô hình cắt đường lý tưởng tại khoảng giá trị thực tế từ **30 đến 40**. Tại dải này, mô hình dự đoán chính xác nhất.
- **Vùng giá trị thấp** (< 30)**:** Đường xanh nằm trên đường đỏ. Mô hình đang có xu hướng Over-predicting (dự đoán cao hơn thực tế) ở các giá trị nhỏ.

- **Vùng giá trị thấp** (> 45): Đường xanh nằm dưới đường đỏ và khoảng cách ngày càng xa. Mô hình đang bị **Under-predicting** (dự đoán thấp hơn thực tế) ở các giá trị lớn. Đây chính là lý do khiến R^2 chưa thể đạt mức tốt nhất trong mô hình.
- **Độ phân tán (Variance):** Các điểm dữ liệu xòe rộng ra ở cuối biểu đồ (hình cái phễu), cho thấy mô hình mất dần sự ổn định khi giá trị thực tế tăng cao.

5.3.3 Đánh giá với các mô hình trước

Tiến hành so sánh kết quả của `regression_metrics()` với các mô hình chúng tôi đã làm trong bài trên, thu được một bảng so sánh như sau:

Bảng 5 Bảng đánh giá hiệu quả các mô hình

	Sai số[1]			**Độ phù hợp**[2]
Mô hình	RMSE	MAE	MAPE (%)	R^2
KNN Regressor	5.6574	4.2213	**0.1328**	0.7902
KMeans + KNN	**5.5819**	**4.2193**	0.1358	**0.7957**
Linear Regression	6.3228	4.5568	0.1472	0.7378

Note: Bảng đánh giá dựa trên việc kiểm thử trong tập dữ liệu của bài toán.

[1]Các chỉ số sai số như RMSE, MAE, MAPE càng nhỏ thì độ hiệu quả của mô hình càng tốt.

[2]Đối với chỉ số R^2, giá trị càng gần đến 1 thì càng hiệu quả.

Có thể thấy được trên tập dữ liệu của đề việc ứng dụng mô hình **KMeans + KNN** sẽ cho ra một kết quả tốt hơn so với 2 mô hình còn lại. Trên tập dữ liệu của bài toán, việc ứng dụng mô hình **hồi quy tuyến tính** không cho ra một kết quả hiệu quả, nguyên nhân là do các điểm dữ liệu không nằm ở vị trí được xem là tối ưu đối với mô hình tuyến tính này, dẫn đến việc nhiễu làm mô hình bị Underfitting và Overfitting ở các giá trị cực biên.

5.4 Hạn chế

Một trong những hạn chế của Linear Regression là mô hình rất dễ nhạy cảm với nhiễu. Đồng thời hạn chế khác của hồi quy tuyến tính là việc nó không biểu diễn được các mô hình phức tạp.

Mô hình trong bài toán hiện tại đang bị Underfitting hoặc Overfitting ở các giá trị cực biên (quá cao hoặc quá thấp) do bản chất của đường thẳng không thể uốn lượn theo sự thay đổi phức tạp của dữ liệu.

6 Kết luận

Trong báo cáo này, chúng tôi đã xây dựng và đánh giá các mô hình dự đoán giá nhà đất dựa trên dữ liệu bất động sản thực tế, với trọng tâm là các phương pháp học máy không tham số, cụ thể là KNN và mô hình lai K-means + KNN. Thông qua quá trình tiền xử lý dữ liệu, bao gồm biến đổi logarit, loại bỏ ngoại lai và chuẩn hóa đặc

trưng, dữ liệu đầu vào được cải thiện đáng kể về mặt phân phối và độ ổn định, tạo điều kiện thuận lợi cho quá trình học của mô hình.

Kết quả thực nghiệm cho thấy mô hình KNN đạt hiệu quả dự đoán tốt với các thước đo đánh giá hồi quy như MAE, RMSE, MAPE và hệ số xác định R^2. Đặc biệt, mô hình lai K-means + KNN, trong đó dữ liệu được phân cụm trước khi áp dụng KNN cho từng cụm, đạt được sự cải thiện nhẹ nhưng nhất quán hơn so với việc áp dụng KNN thuần tuý trên toàn bộ dữ liệu. Mặc dù mức cải thiện không lớn, kết quả này cho thấy việc phân cụm dữ liệu trước khi áp dụng KNN có thể giúp khai thác tốt hơn cấu trúc cục bộ của dữ liệu.

Ngoài ra, trong phần khảo sát các phương pháp khác, mô hình Linear Regression cũng được triển khai để so sánh. Tuy nhiên, kết quả thực nghiệm cho thấy các chỉ số đánh giá của mô hình tuyến tính này kém hơn đáng kể so với mô hình K-means + KNN, gợi ý rằng mối quan hệ giữa các đặc trưng và giá nhà trong bộ dữ liệu không mang tính tuyến tính toàn cục. Ngược lại, KNN và mô hình K-means kết hợp KNN, vốn không giả định trước dạng hàm, có khả năng nắm bắt tốt hơn các cấu trúc cục bộ của dữ liệu, qua đó cho kết quả dự đoán chính xác hơn.

Tóm lại, dự án đã chứng minh hiệu quả của mô hình lai K-means + KNN trong bài toán dự đoán giá nhà đất, đồng thời nhấn mạnh tầm quan trọng của việc lựa chọn mô hình phù hợp với bản chất của dữ liệu. Những kết quả đạt được không chỉ có ý nghĩa về mặt học thuật mà còn có tiềm năng ứng dụng trong các hệ thống hỗ trợ định giá bất động sản trong thực tế.

Tài liệu

[1] Malpezzi, S., *et al.*: Hedonic pricing models: a selective and applied review. Housing economics and public policy **1**, 67–89 (2003)

[2] Harrison Jr, D., Rubinfeld, D.L.: Hedonic housing prices and the demand for clean air. Journal of environmental economics and management **5**(1), 81–102 (1978)

[3] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE transactions on information theory **13**(1), 21–27 (1967)

[4] Kaufman, S., Rosset, S., Perlich, C., Stitelman, O.: Leakage in data mining: Formulation, detection, and avoidance. ACM Transactions on Knowledge Discovery from Data (TKDD) **6**(4), 1–21 (2012)

[5] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer series in statistics. Springer, ??? (2009). https://books.google.com.vn/books?id=eBSgoAEACAAJ

[6] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)

[7] Aha, D.W.: Lazy Learning. Springer, ??? (2013). https://books.google.com.vn/books?id=b1CqCAAAQBAJ

[8] Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**(2), 129–137 (1982) https://doi.org/10.1109/TIT.1982.1056489

[9] Steel, R.G.D., Torrie, J.H.: Principles and Procedures of Statistics: With Special Reference to the Biological Sciences vol. tập 1. McGraw-Hill, ??? (1960). https://books.google.com.vn/books?id=o6FpAAAAMAAJ

[10] Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987) https://doi.org/10.1016/0377-0427(87)90125-7

[11] Thorndike, R.L.: Who belongs in the family? Psychometrika **18**(4), 267–276 (1953) https://doi.org/10.1007/BF02289263

[12] Tiep, V.H.: Machine Learning Co Ban, pp. 95–102. Nha Xuat Ban Khoa Hoc Va Ky Thuat, ??? (2018)