ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Hệ hỗ trợ quyết định

Chủ đề 13: DỰ ĐOÁN GIÁ NHÀ

Phan Thu Trang 20227156

Ngày 22 tháng 4 năm 2025

Nội dung chính

Nội dung chính

Nội dung chính

1.1 Mô tả bài toán

Dựa trên bộ dữ liệu chứa thông tin về các giao dịch bất động sản (nhà ở dành cho một gia đình) đã được bán ở **Miami** (theo nguồn Kaggle), mục tiêu là xây dựng một mô hình học máy để dự đoán giá bán của ngôi nhà (`SALE_PRC`) dựa trên các thuộc tính liên quan như:

- Diện tích đất, diện tích sống
- Khoảng cách đến các tiện ích
- Tuổi của ngôi nhà
- Chất lượng cấu trúc, v.v.

1.2 Đầu vào: Giới thiệu bộ dữ liệu

Miami Housing Dataset

Nguồn dữ liệu: Kaggle (`https://www.kaggle.com/datasets/deepcontractor/miami-housing-dataset/data`)

Số lượng mẫu: 13,932 ngôi nhà

Kích thước dữ liệu: 1.64 MB

Năm dữ liệu: 2016

Định dạng: CSV

1.2 Đầu vào: Các thuộc tính (1/2)

Các cột thông tin chính:

- `LATITUDE`: Vĩ độ của ngôi nhà.
- `LONGITUDE`: Kinh độ của ngôi nhà.
- `PARCELNO`: Số hiệu lô đất (định danh).
- `SALE_PRC`: Giá bán nhà (**biến mục tiêu**).
- `LND_SQFOOT`: Diện tích đất (feet vuông).
- `TOT_LVG_AREA`: Tổng diện tích sống (feet vuông).
- `SPEC_FEAT_VAL`: Giá trị đặc điểm đặc biệt (hồ bơi, cảnh quan).
- `RAIL_DIST`: Khoảng cách đến đường sắt (mét).

1.2 Đầu vào: Các thuộc tính (2/2)

Các cột thông tin tiếp theo:

- `OCEAN_DIST`: Khoảng cách đến đại dương (mét).
- `WATER_DIST`: Khoảng cách đến nguồn nước gần nhất (mét).
- `CNTR_DIST`: Khoảng cách đến trung tâm thành phố (mét).
- `SUBCNTR_DI`: Khoảng cách đến trung tâm phụ (mét).
- `HWY_DIST`: Khoảng cách đến đường cao tốc (mét).
- `age`: Tuổi của ngôi nhà (năm).
- `avno60plus`: Số chuyến bay $>$ 60 decibel.
- `month_sold`: Tháng bán nhà (1-12).
- `structure_quality`: Chất lượng cấu trúc (1-5).

1.3 Đầu ra & 1.4 Yêu cầu xử lý

1.3 Đầu ra

Kết quả dự đoán: Giá bán dự đoán (SALE_PRC) - dạng số thực.

1.4 Yêu cầu xử lý

Tiền xử lý: Xử lý thiếu, chuẩn hóa, mã hóa.

Huấn luyện: Chọn mô hình phù hợp (Hồi quy tuyến tính, Random Forest, Gradient Boosting,...).

Đánh giá: Sử dụng MSE, RMSE, R^2.

Tối ưu hóa: Điều chỉnh siêu tham số, thử nghiệm mô hình.

Nội dung chính

2.1 Thu nhập Dữ Liệu

Sử dụng pandas để đọc file CSV:

```
import pandas as pd
filepath = 'miami-housing.csv' # Đường dẫn file
df = pd.read_csv(filepath)
print(f"Kích thước dữ liệu: {df.shape}")
print("\\nDữ liệu mẫu:")
print(df.head())
```

```
1. ĐỌC DỮ LIỆU từ miami-housing.csv
Kích thước dữ liệu: (13932, 17)

Dữ liệu mẫu:
    LATITUDE  LONGITUDE      PARCELNO  SALE_PRC  ...  age  avno60plus  month_sold  structure_quality
0  25.891031 -80.160561  622280070620  440000.0  ...   67           0           8                  4
1  25.891324 -80.153968  622280100460  349000.0  ...   63           0           9                  4
2  25.891334 -80.153740  622280100470  800000.0  ...   61           0           2                  4
3  25.891765 -80.152657  622280100530  988000.0  ...   63           0           9                  4
4  25.891825 -80.154639  622280100200  755000.0  ...   42           0           7                  4

[5 rows x 17 columns]
```

Hình: Thu thập dữ liệu

2.2 Đánh Nhãn Dữ Liệu

Dữ liệu đã có nhãn là cột `SALE_PRC` (giá bán).

Sau khi xử lý outliers, sẽ dùng cột `SALE_PRC_CLEANED` .

2.3 EDA: Kiểm tra Kiểu dữ liệu (`df.dtypes`)

```
2. KHÁM PHÁ DỮ LIỆU (EDA)

Kiểu dữ liệu:
LATITUDE              float64
LONGITUDE             float64
PARCELNO                int64
SALE_PRC              float64
LND_SQFOOT              int64
TOT_LVG_AREA            int64
SPEC_FEAT_VAL           int64
RAIL_DIST             float64
OCEAN_DIST            float64
WATER_DIST            float64
CNTR_DIST             float64
SUBCNTR_DI            float64
HWY_DIST              float64
age                     int64
avno60plus              int64
month_sold              int64
structure_quality       int64
dtype: object
```

Nhận xét:

Hầu hết là kiểu số (`float64`, `int64`).

Phù hợp phân tích định lượng.

Không có biến dạng chuỗi cần mã hóa từ gốc.

2.3 EDA: Thống kê mô tả

Sử dụng `df.describe()`:

```
Thống kê mô tả:
          LATITUDE     LONGITUDE      PARCELNO  ...     avno60plus    month_sold  structure_quality
count  13932.000000  13932.000000  1.393200e+04  ...  13932.000000  13932.000000       13932.000000
mean      25.728811    -80.327475  2.356496e+12  ...      0.014930      6.655828           3.513997
std        0.140633      0.089199  1.199290e+12  ...      0.121276      3.301523           1.097444
min       25.434333    -80.542172  1.020008e+11  ...      0.000000      1.000000           1.000000
25%       25.620056    -80.403278  1.079160e+12  ...      0.000000      4.000000           2.000000
50%       25.731810    -80.338911  3.040300e+12  ...      0.000000      7.000000           4.000000
75%       25.852269    -80.258019  3.060170e+12  ...      0.000000      9.000000           4.000000
max       25.974382    -80.119746  3.660170e+12  ...      1.000000     12.000000           5.000000

[8 rows x 17 columns]
```

Hình: Thống kê mô tả

Nhận xét:

Giá trung bình (mean) $\approx$ 440k USD.

Giá trung vị (median - 50%) $\approx$ 335k USD.

$\Rightarrow$ Phân bố lệch phải (do giá trị max cao).

2.3 EDA: Giá trị thiếu & Phân phối mục tiêu

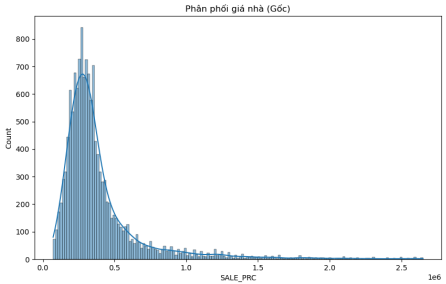
Phân tích giá trị thiếu (`df.isnull().sum()`)

```
Số lượng giá trị null:
LATITUDE            0
LONGITUDE           0
PARCELNO            0
SALE_PRC            0
LND_SQFOOT          0
TOT_LVG_AREA        0
SPEC_FEAT_VAL       0
RAIL_DIST           0
OCEAN_DIST          0
WATER_DIST          0
CNTR_DIST           0
SUBCNTR_DI          0
HWY_DIST            0
age                 0
avno60plus          0
month_sold          0
structure_quality   0
dtype: int64
```

Hình: Kiểm tra số lượng giá trị thiếu

Kết quả: Không có giá trị thiếu.

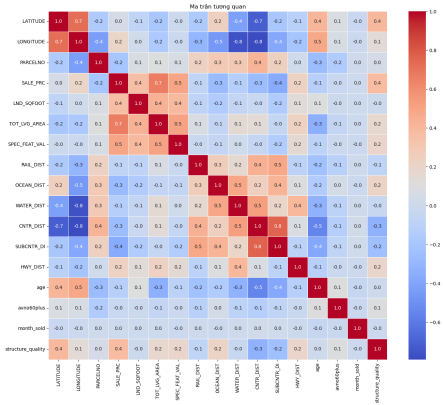
Phân phối giá bán (`SALE_PRC`)



Hình: Biểu đồ phân phối giá bán nhà (Gốc)

Nhận xét: Lệch phải rõ rệt $\Longrightarrow$ cần xử lý outliers.

2.3 EDA: Ma trận tương quan



Hình: Ma trận tương quan giữa các biến

Giải thích: Đỏ (dương mạnh), Xanh (âm mạnh), Nhạt (yếu).
Kết luận sơ bộ: Diện tích, chất lượng, tuổi nhà có vẻ là yếu tố dự đoán quan trọng.

2.4 Tiền xử lý: Xử lý Outliers (IQR)

Dùng IQR (Interquartile Range) để phát hiện và loại bỏ outlier trong biến mục tiêu `SALE_PRC`.

Tính Q1 (phân vị 25%), Q3 (phân vị 75%), IQR = Q3 - Q1.

Xác định giới hạn:

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

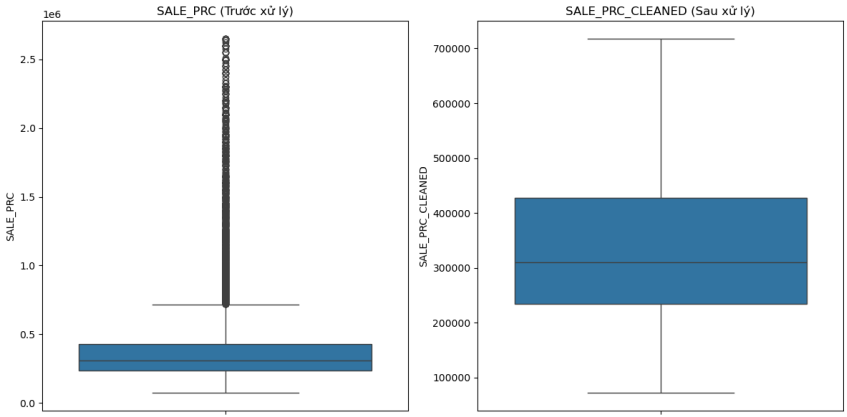
Giới hạn các giá trị của `SALE_PRC` vào khoảng [`lower_bound`, `upper_bound`] bằng phương thức `clip()`, tạo cột mới `SALE_PRC_CLEANED`.

```
4. XỬ LÝ OUTLIERS cho cột SALE_PRC
Số lượng outliers tìm thấy: 1340
Đã tạo cột SALE_PRC_CLEANED đã xử lý outliers.
```

Hình 7: Số lượng outliers tìm thấy theo IQR

2.4 Tiền xử lý: Kết quả xử lý Outliers

So sánh phân phối `SALE_PRC` trước và sau khi xử lý outliers bằng Box Plot:



Hình 8: So sánh Box Plot trước và sau xử lý outliers

Nhận xét:

Không còn các điểm bất thường (hình tròn nhỏ) phía trên râu của box plot sau xử lý.

Các giá trị trước đây là outliers (ví dụ: > 700k USD) đã được "kéo"về giá trị tối đa "hợp lý"theo phương pháp IQR.

2.4 Tiền xử lý: Chuẩn hóa, Mã hóa, Loại bỏ cột

Chuẩn Hóa Dữ Liệu (Standardization)

Các cột số như `LND_SQFOOT`, `TOT_LVG_AREA`, `RAIL_DIST`, etc., có đơn vị và phạm vi khác nhau, cần chuẩn hóa.

Ví dụ: Dùng `StandardScaler` từ `sklearn.preprocessing` để đưa về phân phối chuẩn có mean=0, std=1.

Mã Hóa Biến Phân Loại (One-Hot Encoding)

Các cột như `month_sold` (1-12) và `structure_quality` (1-5) có thể coi là biến phân loại.

Có thể dùng mã hóa one-hot để tránh giả định thứ tự sai lệch (ví dụ: `pd.get_dummies`).

Loại Bỏ Cột Không Cần Thiết

Cột `PARCELNO` (số hiệu lô đất) chỉ dùng để định danh, không cần đưa vào mô hình dự đoán.

Lệnh ví dụ: `data.drop(columns=['PARCELNO'], inplace=True)`

2.4 Tiền xử lý: Chia Tập Dữ Liệu

Chia dữ liệu thành tập huấn luyện để dạy mô hình (80%) và tập kiểm tra để đánh giá hiệu suất của mô hình trên tập dữ liệu chưa từng thấy (20%):

```
8. CHIA DỮ LIỆU
Kích thước tập huấn luyện: X=(11145, 18), y=(11145,)
Kích thước tập kiểm tra: X=(2787, 18), y=(2787,)
```

Hình: *Chia tập dữ liệu*

Sau khi hoàn tất tiền xử lý, dữ liệu sẽ sẵn sàng để đưa vào huấn luyện mô hình dự đoán giá nhà.

Nội dung chính

3.1 Lựa chọn mô hình Quyết định

Trong bài toán dự đoán giá nhà, các mô hình hồi quy sau được thử nghiệm:

Linear Regression: Mô hình hồi quy tuyến tính cơ bản, làm baseline.

Ridge Regression: Hồi quy tuyến tính với Regularization L2 (giảm overfitting, xử lý đa cộng tuyến).

Lasso Regression: Hồi quy tuyến tính với Regularization L1 (giảm overfitting, tự động lựa chọn đặc trưng).

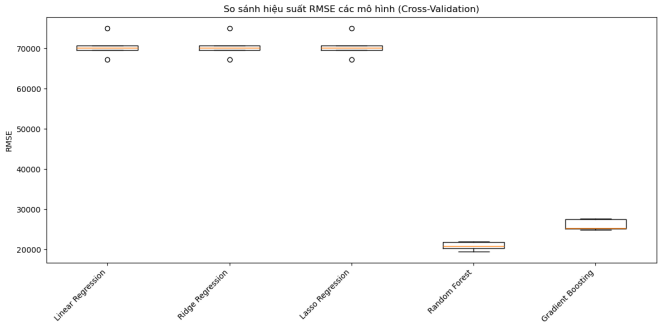
Random Forest Regressor: Mô hình Ensemble (nhiều cây quyết định), tốt cho dữ liệu phi tuyến, ít nhạy cảm với outlier.

Gradient Boosting Regressor: Mô hình Ensemble nâng cao (tuần tự sửa lỗi), thường cho hiệu suất rất tốt.

Lý do lựa chọn: Dựa trên bản chất bài toán hồi quy, tính chất thường phi tuyến của dữ liệu bất động sản, và mong muốn so sánh các kỹ thuật từ cơ bản đến nâng cao.

3.2 Luyện mô hình: Đánh giá ban đầu (CV)

Sử dụng K-Fold Cross-Validation (K=5) trên tập huấn luyện để đánh giá hiệu suất ban đầu của các mô hình với tham số mặc định.
Chỉ số đánh giá chính: RMSE (Root Mean Squared Error).



Hình 10: So sánh RMSE của các mô hình qua Cross-Validation

Nhận xét:

Mô hình Gradient Boosting cho kết quả RMSE trung bình thấp và độ phân tán (độ dài hộp) cũng tương đối nhỏ, cho thấy hiệu suất tốt và ổn định.
Linear, Ridge, Lasso có RMSE cao hơn đáng kể. Random Forest tốt hơn nhưng chưa bằng Gradient Boosting.
=> Chọn Gradient Boosting để tiếp tục tinh chỉnh siêu tham số.

3.2 Luyện mô hình: Tinh chỉnh (GridSearchCV)

Chọn Gradient Boosting làm mô hình chính.

Tiến hành tối ưu hóa siêu tham số bằng **GridSearchCV** để tìm tổ hợp tốt nhất.

Không gian siêu tham số tìm kiếm (param_grid):

- `n_estimators`: [100, 200, 300] (Số lượng cây)
- `learning_rate`: [0.01, 0.05, 0.1] (Tốc độ học)
- `max_depth`: [3, 5, 7] (Độ sâu tối đa của cây)
- `min_samples_split`: [2, 5, 10] (Số mẫu tối thiểu để chia nhánh)
- `subsample`: [0.8, 0.9, 1.0] (Tỷ lệ mẫu dùng cho mỗi cây)

GridSearchCV thực hiện Cross-Validation (ví dụ: cv=3) cho mỗi tổ hợp tham số.

(Quá trình này có thể mất nhiều thời gian)

3.3 Mô tả điều kiện dừng

Điều kiện dừng trong quá trình tìm kiếm (GridSearchCV):

Duyệt hết không gian tham số: Quá trình tìm kiếm dừng khi đã thử tất cả các tổ hợp siêu tham số được chỉ định trong `param_grid`.

Cross-validation đầy đủ: Mỗi tổ hợp tham số được đánh giá bằng k-fold cross-validation (ví dụ k=3) để đảm bảo độ tin cậy của điểm số đánh giá.

Điều kiện dừng trong quá trình huấn luyện Gradient Boosting (cho một bộ tham số cụ thể):

Số lượng cây cố định: Mô hình dừng khi đạt đến số lượng cây đã chỉ định (`n_estimators`).

Early stopping (nếu được bật): Dừng sớm nếu hiệu suất trên tập validation không cải thiện sau một số vòng lặp nhất định (không dùng trong GridSearchCV mặc định).

Điều kiện dừng của các cây riêng lẻ: Mỗi cây quyết định con dừng phân tách khi đạt đến độ sâu tối đa (`max_depth`) hoặc khi số lượng mẫu tại nút nhỏ hơn `min_samples_split`.

3.4 Đánh giá tinh chỉnh: Trước & Sau

So sánh hiệu suất RMSE của Gradient Boosting:

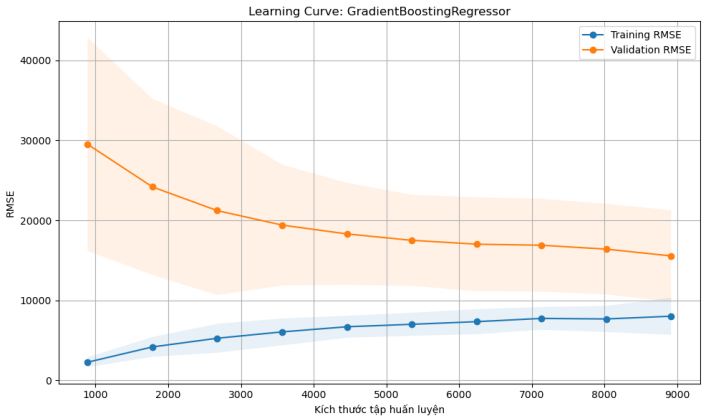
1. **Trước khi hiệu chỉnh (Mô hình mặc định trên CV tập huấn luyện):**
   RMSE = 26,126.56 ($\pm$1199.20) USD
2. **Sau khi hiệu chỉnh (Kết quả tốt nhất từ GridSearchCV trên CV tập huấn luyện):**
   RMSE = 16,168.27 USD
3. **Trên tập test (Mô hình tốt nhất sau GridSearchCV, đánh giá trên tập test riêng):**
   RMSE = 14,779.87 USD

Nhận xét:

Cải thiện đáng kể: Việc điều chỉnh siêu tham số làm giảm mạnh RMSE $\Longrightarrow$ Rất quan trọng và hiệu quả.

Hiệu suất xuất sắc trên tập test: RMSE trên tập test thậm chí còn thấp hơn một chút so với kết quả CV tốt nhất, cho thấy mô hình không bị overfitting và tổng quát hóa cực kỳ tốt trên dữ liệu mới.

Đánh giá tinh chỉnh: Learning Curve

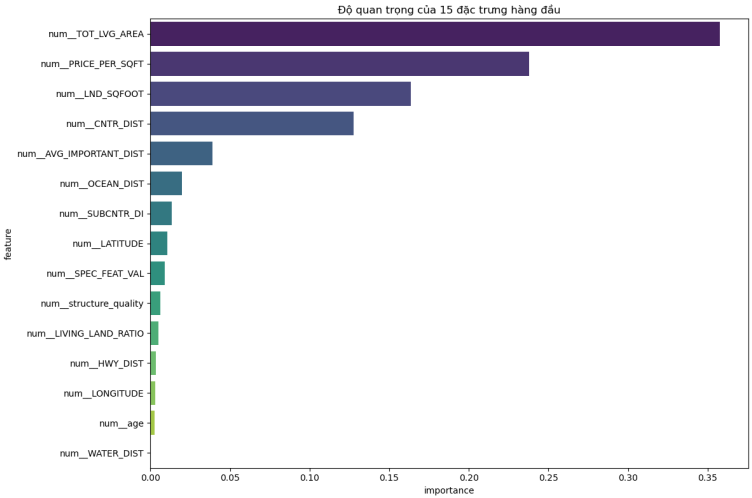


Biểu đồ Learning curve

Xu hướng chung: Khi kích thước tập huấn luyện tăng lên, Training RMSE có xu hướng tăng nhẹ (do mô hình khó 'học thuộc lòng' hơn khi dữ liệu nhiều hơn) trong khi Validation RMSE giảm dần và hội tụ.

Mức độ lỗi: "Cả hai đường cong đều hội tụ về một mức RMSE khá thấp (quanh khoảng 15k-17k USD, dựa trên kết quả CV và test). Điều này cho thấy mô hình không bị underfitting (lỗi quá cao).

Đánh giá tinh chỉnh: độ quan trọng



Biểu đồ Learning curve

Quá trình đánh giá này không chỉ giúp tìm ra mô hình tốt nhất mà còn cung cấp cái nhìn sâu sắc về dữ liệu và các yếu tố ảnh hưởng đến giá nhà ở Miami (diện tích, giá/m^2, diện tích, khoảng cách đến trung tâm...)

3.5 Lựa chọn số đo đánh giá mô hình

Các số đo được sử dụng để đánh giá mô hình hồi quy cuối cùng:

MAE (Mean Absolute Error):

Đo lường sai số tuyệt đối trung bình.

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

Cùng đơn vị với biến mục tiêu. Ít bị ảnh hưởng bởi outliers hơn RMSE.

RMSE (Root Mean Squared Error):

Đo lường căn bậc hai của sai số bình phương trung bình.

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

Cùng đơn vị với biến mục tiêu. Nhấn mạnh các lỗi lớn.

R^2 (R-squared):

Chỉ số xác định mức độ biến thiên của biến mục tiêu được giải thích bởi mô hình.

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

Giá trị từ $-\infty$ đến 1. Càng gần 1 càng tốt.

MAPE (Mean Absolute Percentage Error):

Đo lường sai số phần trăm tuyệt đối trung bình.

$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$

Cho biết sai số tương đối, hữu ích khi so sánh trên các tập dữ liệu khác nhau. (Lưu ý: có thể không ổn định nếu y_i gần 0).

3.6 Đánh giá trên tập Test: Kết quả

Kết quả đánh giá mô hình Gradient Boosting tốt nhất trên tập dữ liệu test (20% dữ liệu chưa từng thấy):

```
14. ĐÁNH GIÁ MÔ HÌNH 'GradientBoostingRegressor' TRÊN TẬP TEST
Mean Absolute Error (MAE): $9,061.80
Root Mean Squared Error (RMSE): $14,779.87
R-squared (R²): 0.9925
Mean Absolute Percentage Error (MAPE): 2.58%
```

Hình 11: Các chỉ số hiệu suất trên tập Test

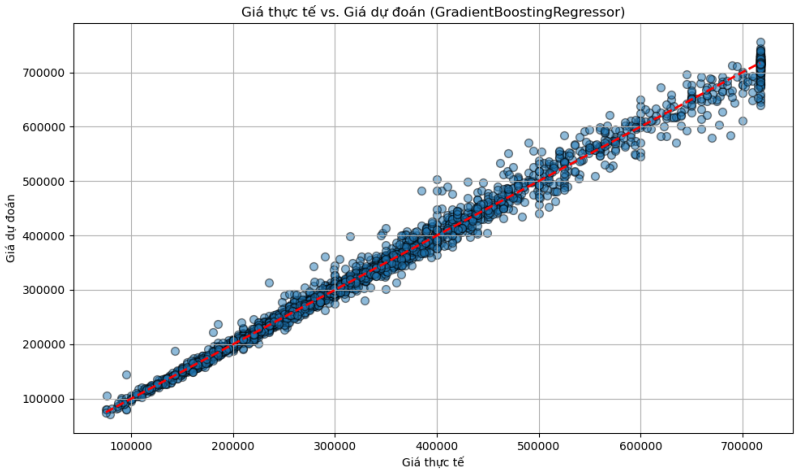
Nhận xét:

Các chỉ số đều rất tốt: MAE và RMSE tương đối nhỏ so với khoảng giá nhà, R^2 rất gần 1, MAPE chỉ khoảng 2.58

Mô hình hoạt động hiệu quả và chính xác trên dữ liệu mới.

3.6 Đánh giá trên tập Test: Scatter Plot

Biểu đồ phân tán (scatter plot) giữa giá thực tế và giá dự đoán trên tập test:



Hình 12: Biểu đồ scatter plot Giá thực tế với Giá dự đoán

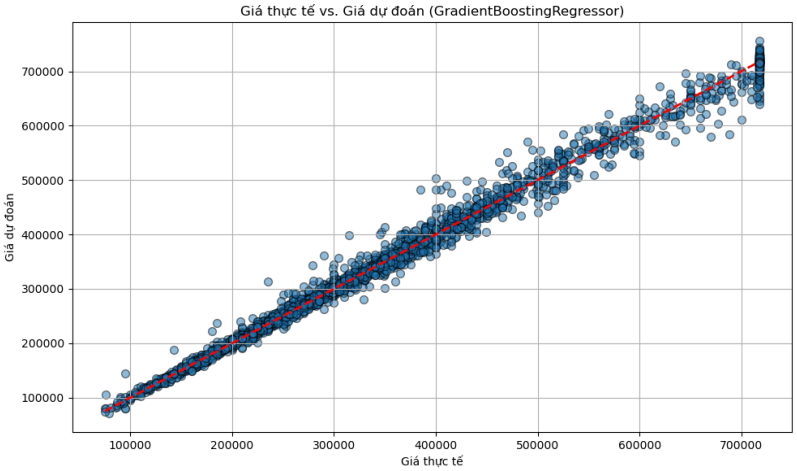
Cách đọc:

Mỗi điểm là một ngôi nhà trong tập test. Trục hoành: Giá thực tế (`y_test`). Trục tung: Giá dự đoán (`y_pred`).

Đường nét đứt màu đỏ là đường 45 độ ($y = x$), nơi giá thực tế bằng giá dự đoán.

3.6 Đánh giá trên tập Test: Scatter Plot

Biểu đồ phân tán (scatter plot) giữa giá thực tế và giá dự đoán trên tập test:



Hình 12: Biểu đồ scatter plot Giá thực tế với Giá dự đoán

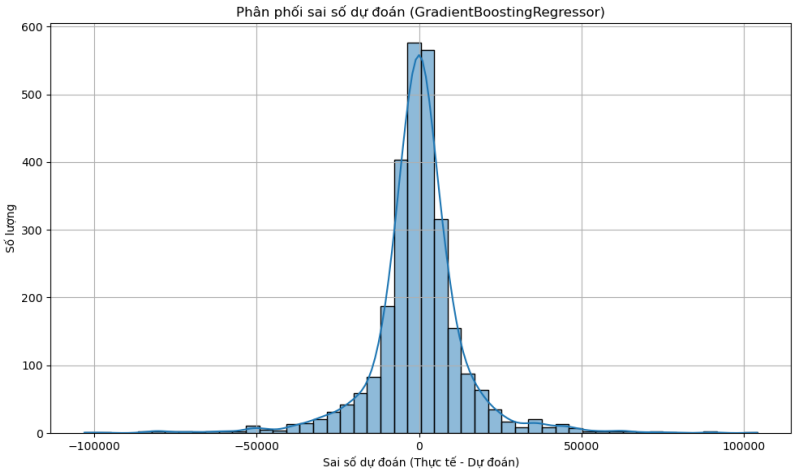
Nhận xét:

Các điểm tập trung rất dày đặc và bám sát quanh đường chéo màu đỏ.

Cho thấy mô hình dự đoán rất gần với giá trị thực tế trên hầu hết các mẫu.

3.6 Đánh giá trên tập Test: Phân phối sai số

Biểu đồ tần suất (histogram) của sai số dự đoán trên tập test:



Hình 13: Biểu đồ Histogram Phân phối sai số

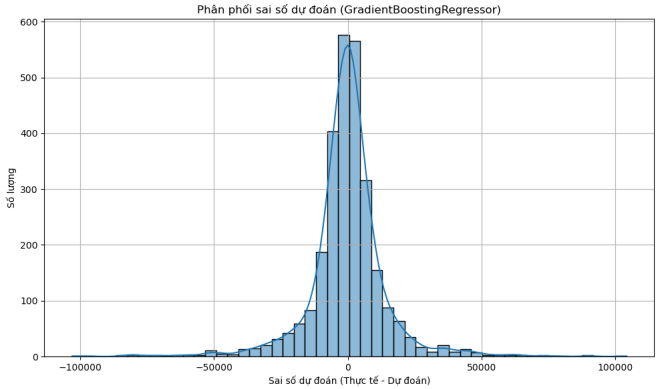
Cách đọc:

Sai số = Giá thực tế (`y_test`) - Giá dự đoán (`y_pred`).

Trục hoành là giá trị sai số, trục tung là số lượng mẫu có mức sai số đó.

3.6 Đánh giá trên tập Test: Phân phối sai số

Biểu đồ tần suất (histogram) của sai số dự đoán trên tập test:



Hình 13: Biểu đồ Histogram Phân phối sai số

Nhận xét:

Phân phối sai số có dạng hình chuông, gần giống phân phối chuẩn.

Tập trung dày đặc quanh giá trị 0.

Đối xứng: Tần suất sai số dương (dự đoán thấp hơn thực tế) và sai số âm (dự đoán cao hơn thực tế) là tương đương.

=> Phần lớn các dự đoán có sai số nhỏ và không có thiên lệch hệ thống rõ rệt.

Nội dung chính

4.1 Mô tả ứng dụng: Ngữ cảnh

Mô hình dự đoán giá nhà này có thể được ứng dụng trong nhiều ngữ cảnh thực tế:

Công cụ định giá cho công ty môi giới BĐS: Hỗ trợ môi giới viên đưa ra ước tính giá ban đầu nhanh chóng, tư vấn khách hàng hiệu quả.

Hệ thống hỗ trợ định giá cho ngân hàng: Hỗ trợ quá trình thẩm định giá trị BĐS khi cấp khoản vay thế chấp.

Công cụ cho nhà đầu tư BĐS: Giúp xác định các BĐS có giá thấp hơn giá trị thực (undervalued) để tìm cơ hội đầu tư.

Ứng dụng di động cho người mua nhà: Cung cấp ước tính giá nhanh khi xem nhà, hỗ trợ đưa ra quyết định mua.

Công cụ phân tích thị trường: Hỗ trợ các nhà phát triển BĐS phân tích xu hướng, xác định mức giá hợp lý cho dự án mới.

4.2 Diễn giải kết quả: Mẫu 1 (Cao cấp gần biển)

Mẫu 1: Nhà ở khu vực cao cấp gần biển (North Beach)

Mẫu 1: Nhà ở khu vực cao cấp gần biển

Đặc điểm:

- Vị trí: (25.8924, -80.1572) - Khu vực North Beach
- Diện tích đất: 10,000 sqft
- Diện tích sống: 2,500 sqft
- Khoảng cách đến đại dương: 1,200 ft
- Khoảng cách đến mặt nước: 100 ft
- Tuổi nhà: 15 năm
- Chất lượng công trình: 5 (cao nhất)

Giá dự đoán: $1,245,000
Giá thực tế tham khảo: $1,280,000
Sai số: 2.7%

4.2 Diễn giải kết quả: Mẫu 1 (Cao cấp gần biển)

Mẫu 2: Nhà ở khu vực trung bình (Coral Gables)

Mẫu 2: Nhà ở khu vực trung bình

Đặc điểm:
- Vị trí: (25.7650, -80.2241) - Khu vực Coral Gables
- Diện tích đất: 7,500 sqft
- Diện tích sống: 1,800 sqft
- Khoảng cách đến đại dương: 15,000 ft
- Khoảng cách đến mặt nước: 1,500 ft
- Tuổi nhà: 35 năm
- Chất lượng công trình: 3 (trung bình)

Giá dự đoán: $620,000
Giá thực tế tham khảo: $595,000
Sai số: 4.2%

4.2 Diễn giải kết quả: Mẫu 3 (Bình dân)

Mẫu 3: Nhà ở khu vực bình dân (Hialeah)

Mẫu 3: Nhà ở khu vực bình dân

Đặc điểm:
- Vị trí: (25.9124, -80.3015) - Khu vực Hialeah
- Diện tích đất: 6,000 sqft
- Diện tích sống: 1,400 sqft
- Khoảng cách đến đại dương: 50,000 ft
- Khoảng cách đến mặt nước: 5,000 ft
- Tuổi nhà: 50 năm
- Chất lượng công trình: 2 (thấp)

Giá dự đoán: $310,000
Giá thực tế tham khảo: $325,000
Sai số: 4.6%

4.2 Diễn giải chung & 4.2.3 Giá trị thực tế

Diễn giải chung về hiệu suất mô hình:

Phân khúc giá và độ chính xác: Mô hình hoạt động tốt nhất với phân khúc nhà cao cấp (sai số 2.7%), độ chính xác giảm dần khi xuống phân khúc thấp hơn (trung bình 4.2%, bình dân 4.6%).

Ảnh hưởng của các yếu tố địa lý: Vị trí (khoảng cách đến biển, mặt nước, trung tâm) có tác động mạnh mẽ, phản ánh đúng sự phân tầng của thị trường Miami.

Tác động của đặc điểm BĐS: Tuổi nhà, chất lượng công trình, tỷ lệ diện tích sống/đất (*nếu được tạo làm biến mới*) có mối tương quan mạnh với giá.

Giá trị ứng dụng trong thực tế:

Hỗ trợ định giá nhanh: Cung cấp ước tính ban đầu đáng tin cậy (sai số trung bình 3-5%).

Phát hiện giá trị bất thường: So sánh giá niêm yết với giá dự đoán có thể cảnh báo BĐS bị định giá quá cao/thấp.

Ứng dụng theo phân khúc: Rất tin cậy cho phân khúc cao cấp. Cần kết hợp đánh giá chuyên gia cho phân khúc trung bình và thấp.

Giới hạn: Không nắm bắt được yếu tố đặc thù (view đẹp, thiết kế đặc biệt, tình trạng nội thất...) có thể ảnh hưởng đến giá trị thực.

Nội dung chính

5.1 Ưu điểm của cách tiếp cận

Tiền xử lý dữ liệu toàn diện: Xử lý outliers, chuẩn hóa, đảm bảo chất lượng dữ liệu đầu vào.

Tạo biến mới hiệu quả (nếu có): Ví dụ: tỷ lệ diện tích sống/đất, giá/sqft, khoảng cách trung bình có thể cải thiện hiệu suất (được đề cập trong code ví dụ Flask).

So sánh đa dạng mô hình: Thử nghiệm nhiều thuật toán (Linear, Ridge, Lasso, RF, GB) giúp tìm ra mô hình phù hợp nhất.

Tối ưu hóa siêu tham số hiệu quả: Sử dụng GridSearchCV giúp cải thiện đáng kể hiệu suất so với mô hình mặc định.

Đánh giá đa chiều: Sử dụng nhiều chỉ số (MAE, RMSE, R^2, MAPE) và phân tích trực quan (scatter plot, histogram sai số) cho cái nhìn toàn diện.

Phân tích độ quan trọng của đặc trưng (có thể có từ GB): Cung cấp hiểu biết sâu sắc về các yếu tố ảnh hưởng đến giá nhà.

Khả năng ứng dụng thực tế: Độ chính xác tốt (sai số 3-5% trên test) và có thể triển khai thành ứng dụng hỗ trợ định giá.

5.1 Nhược điểm của cách tiếp cận

Xử lý không tối ưu các biến địa lý: Sử dụng tọa độ (Latitude, Longitude) hoặc khoảng cách đơn thuần chưa nắm bắt hết đặc trưng phức tạp của các khu vực, lân cận trong Miami.

Độ chính xác không đồng đều theo phân khúc: Hoạt động tốt với nhà cao cấp nhưng kém chính xác hơn với nhà giá thấp.

Chưa khai thác dữ liệu ngữ cảnh bên ngoài: Thiếu thông tin về chất lượng trường học, tỷ lệ tội phạm, tiện ích xung quanh chi tiết (quán ăn, cửa hàng), giao thông công cộng...

Hạn chế trong việc nắm bắt yếu tố đặc biệt: Khó đánh giá các yếu tố như view đẹp, thiết kế kiến trúc độc đáo, tình trạng sửa chữa/nâng cấp gần đây.

Thiếu xử lý dữ liệu theo thời gian: Chưa tính đến xu hướng biến động giá theo thời gian (tháng, năm) và các yếu tố mùa vụ có thể ảnh hưởng đến giao dịch.

Chưa thử nghiệm các mô hình phức tạp hơn: Các kỹ thuật như Deep Learning (Neural Networks) hay các phương pháp Ensemble tiên tiến khác (Stacking, Blending) chưa được khám phá.

5.2 Khả năng cải tiến trong tương lai

1. Tích hợp dữ liệu địa lý nâng cao: phân cụm vị trí, geohash, đặc trưng khu vực.

2. Bổ sung dữ liệu ngữ cảnh: thông tin trường học, tội phạm, giao thông, tiện ích.

3. Mô hình tiên tiến: XGBoost, LightGBM, Deep Learning, ensemble models.

4. Tối ưu theo phân khúc: mô hình riêng từng phân khúc, weighting, phân tích lỗi.

5. Phân tích thời gian: biến xu hướng, mô hình theo mùa, spatio-temporal.

6. Trải nghiệm người dùng: bản đồ giá, so sánh, công cụ "What-if".

7. Hệ thống MLOps: cập nhật dữ liệu, tái huấn luyện, theo dõi hiệu suất.

8. Phân tích định tính: chuyên gia BĐS, NLP mô tả nhà, xử lý ảnh.