ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN

Bài tập tuần 8
Hệ hỗ trợ quyết định

Chủ đề 13
Dự đoán giá nhà ở Boston

Giảng viên hướng dẫn: TS. Trần Ngọc Thăng
Mã lớp: 158242
Sinh viên thực hiện: Phan Thu Trang
MSSV: 20227156

Hà Nội, Ngày 1 tháng 6 năm 2025

Mục lục

Lời mở đầu

Danh sách hình vẽ

Phần 1. Xử lý dữ liệu

1.1. Mô tả bài toán, đầu ra, đầu vào, yêu cầu xử lý

1.1.1. Mô tả bài toán

- Mục tiêu: Xây dựng một mô hình hồi quy tuyến tính để dự đoán giá nhà dựa trên các đặc điểm của ngôi nhà.
- Tầm quan trọng/ứng dụng thực tế: Giúp người mua/bán ước lượng giá, hỗ trợ nhà đầu tư, công ty bất động sản,...

1.1.2. Đầu vào

Giới thiệu bộ dữ liệu

Bộ dữ liệu Miami Housing Dataset là một tập hợp dữ liệu chi tiết về các ngôi nhà dành cho một gia đình được bán ở khu vực Miami. Dưới đây là một mô tả chi tiết về bộ dữ liệu này:

- 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.

Các thuộc tính trong bộ dữ liệu

Bộ dữ liệu ở định dạng CSV với các cột thông tin sau:

- **LATITUDE**: Vĩ độ của ngôi nhà.
- **LONGITUDE**: Kinh độ của ngôi nhà.
- **PARCELNO**: Số hiệu lô đất (có thể dùng làm định danh duy nhất).
- **SALE_PRC**: Giá bán của ngôi nhà (biến mục tiêu).

- **LND_SQFOOT**: Diện tích đất (đơn vị: feet vuông).
- **TOT_LVG_AREA**: Tổng diện tích sống (đơn vị: feet vuông).
- **SPEC_FEAT_VAL**: Giá trị các đặc điểm đặc biệt (ví dụ: hồ bơi, cảnh quan).
- **RAIL_DIST**: Khoảng cách đến đường sắt (đơn vị: mét).
- **OCEAN_DIST**: Khoảng cách đến đại dương (đơn vị: mét).
- **WATER_DIST**: Khoảng cách đến nguồn nước gần nhất (đơn vị: mét).
- **CNTR_DIST**: Khoảng cách đến trung tâm thành phố (đơn vị: mét).
- **SUBCNTR_DI**: Khoảng cách đến trung tâm phụ (đơn vị: mét).
- **HWY_DIST**: Khoảng cách đến đường cao tốc (đơn vị: mét).
- **age**: Tuổi của ngôi nhà (tính bằng năm).
- **avno60plus**: Số chuyến bay trên 60 decibel (có thể là biến nhị phân hoặc số nguyên).
- **month_sold**: Tháng bán nhà (1-12).
- **structure_quality**: Chất lượng cấu trúc (thang điểm từ 1-5).

1.1.3. Đầu ra

Kết quả dự đoán: Giá bán dự đoán của ngôi nhà (**SALE_PRC_CLEAND**) sau khi tiền xử lý (xử lý outlier, feature engineering, v.v.).

1.1.4. Yêu cầu xử lý

- Tiền xử lý dữ liệu để đảm bảo chất lượng đầu vào cho mô hình (xử lý giá trị thiếu, chuẩn hóa dữ liệu, mã hóa biến phân loại nếu cần).
- Lựa chọn và huấn luyện mô hình học máy phù hợp (ví dụ: hồi quy tuyến tính, rừng ngẫu nhiên, hoặc gradient boosting).
- Đánh giá hiệu suất mô hình bằng các chỉ số như **MSE** (Mean Squared Error), **RMSE** (Root Mean Squared Error), hoặc $\mathbf{R^2}$.
- Tối ưu hóa mô hình nếu cần (điều chỉnh siêu tham số, thử nghiệm các mô hình khác).

1.2. Đánh nhãn & Tiền xử lý dữ liệu

1.2.1. Đánh nhãn

Trong trường hợp này, dữ liệu đã có nhãn là cột `SALE_PRC` (giá bán) - đây là biến mục tiêu của bài toán hồi quy.
Sau khi xử lý outliers, chúng ta sẽ sử dụng cột mới `SALE_PRC_CLEANED` làm biến mục tiêu cuối cùng cho mô hình hồi quy.

Thực hiện phân tích thống kê cơ bản để hiểu dữ liệu:

1. **Đọc dữ liệu**

   - Dữ liệu được cung cấp dưới dạng tệp CSV với các cột đã được định nghĩa rõ ràng.
   - Sử dụng thư viện như pandas trong Python để đọc dữ liệu từ tệp:

```
import pandas as pd
print(f"\n1. ĐỌC DỮ LIỆU từ {filepath}")
df = pd.read_csv(filepath)
```

   - Hiển thị output của `df.shape()` và `df.head()` (trình bày 5 hàng đầu tiên của bảng dữ liệu):

```
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 1: Đọc dữ liệu.

2. **Kiểu dữ liệu**

```
print("\nKiểu dữ liệu:")
print(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
```

Hình 2: Kiểm tra kiểu dữ liệu

Hầu hết các cột là kiểu số, phù hợp cho phân tích định lượng. Không có biến phân loại dạng chuỗi cần mã hóa đặc biệt trong dữ liệu gốc này.

3. **Thống kê mô tả:**

```
print("\nThống kê mô tả:")
print(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 3: Thống kê mô tả

4. **Phân tích giá trị thiếu:**

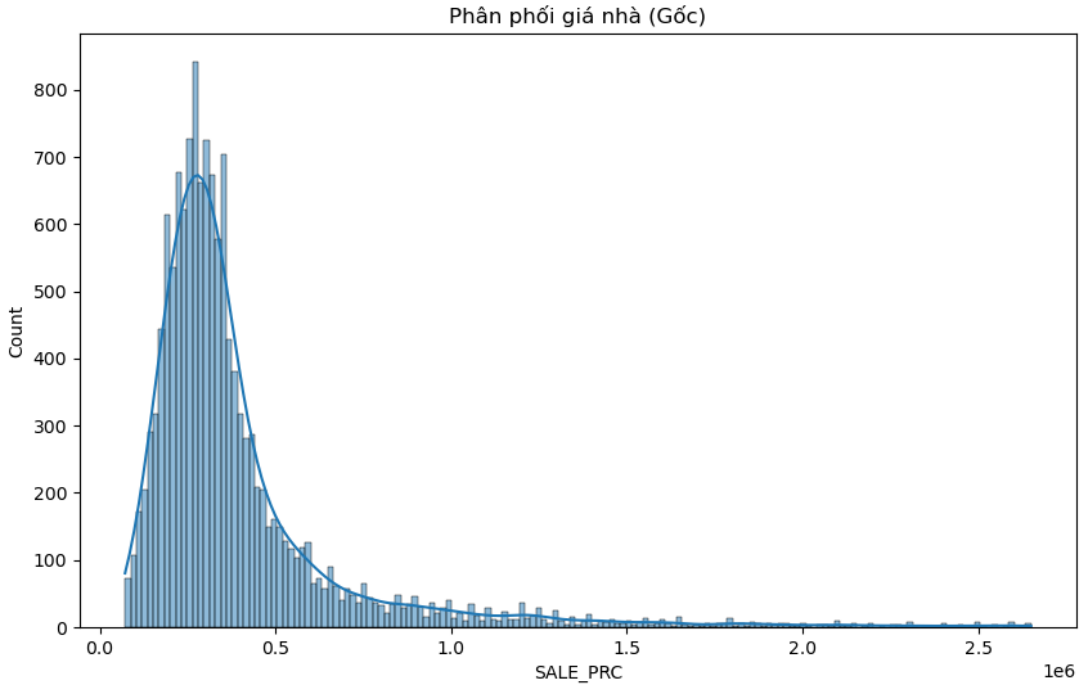
```
print("\nSố lượng giá trị null:")
print(df.isnull().sum())
```

```
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 4: Kiểm tra số lượng giá trị thiếu

Kết quả kiểm tra cho thấy không có cột nào có giá trị thiếu trong tập dữ liệu này.

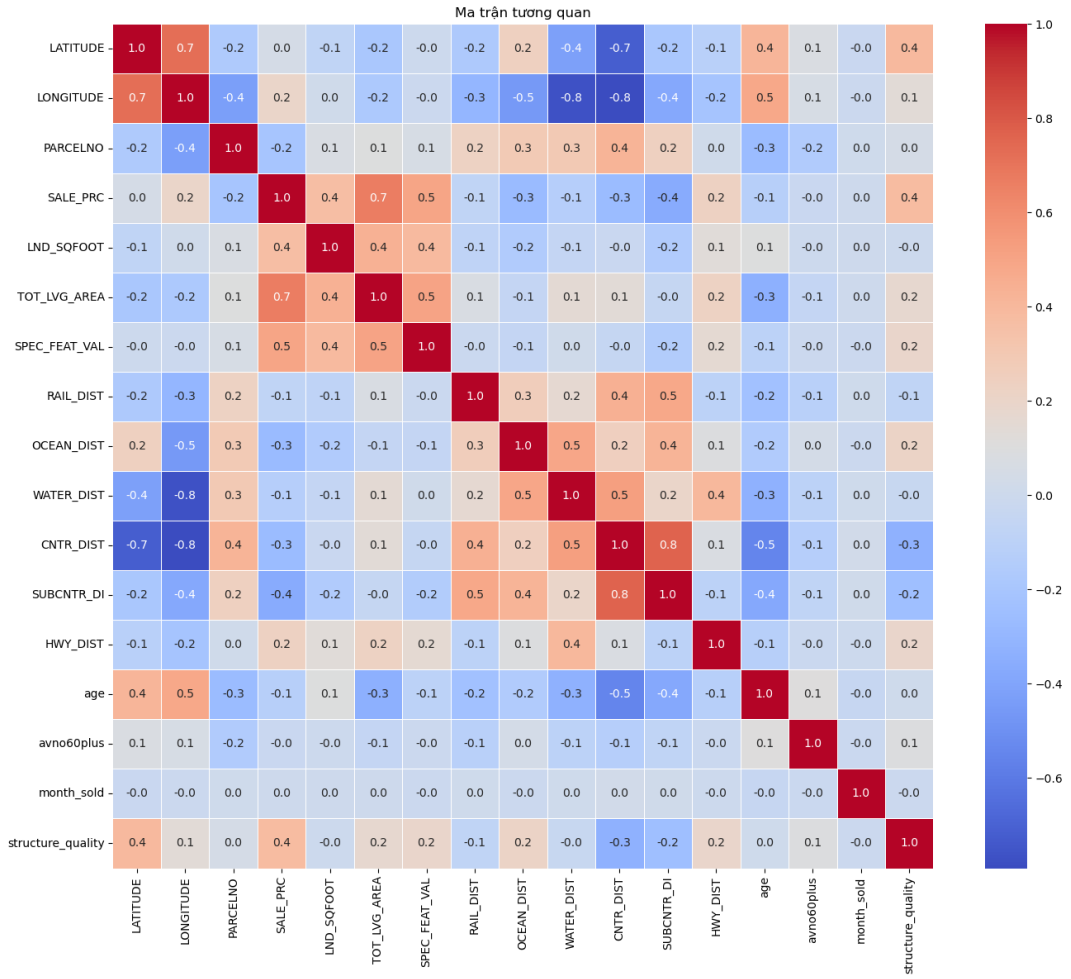
5. **Phân tích biến mục tiêu:**



Hình 5: Biểu đồ phân phối giá bán nhà

- Biểu đồ Histogram cho thấy phân phối của giá bán nhà (`SALE_PRC`) trước khi xử lý outlier: bị lệch phải rõ rệt, với một cái đuôi dài về phía giá trị cao.
- Ý nghĩa: Phần lớn các giao dịch nhà có giá trị tập trung ở mức thấp và trung bình, nhưng tồn tại một số ít giao dịch với giá trị rất cao.
- Sự lệch này và sự hiện diện của các giá trị rất cao (outliers) cho thấy sự cần thiết phải xử lý các giá trị bất thường trước khi huấn luyện mô hình.

6. **Phân tích tương quan giữa các biến:**



Hình 6: Ma trận tương quan giữa các biến

- **Giải thích:** Màu đỏ đậm thể hiện tương quan dương mạnh, màu xanh đậm thể hiện tương quan âm mạnh, màu nhạt thể hiện tương quan yếu.
- **Kết luận sơ bộ:** Phân tích tương quan ban đầu cho thấy các yếu tố như diện tích, chất lượng và tuổi nhà có vẻ là những yếu tố dự đoán quan trọng cho giá nhà.

1.2.2. Tiền xử lý dữ liệu

1. **Xử lý giá trị thiếu**
   Xử lý giá trị thiếu (NaN) bằng SimpleImputer (thay bằng giá trị trung bình (cho biến số) hoặc giá trị phổ biến nhất (cho biến phân loại) – đã được thực hiện trong bộ tiền xử lý (hàm `create_preprocessor`).

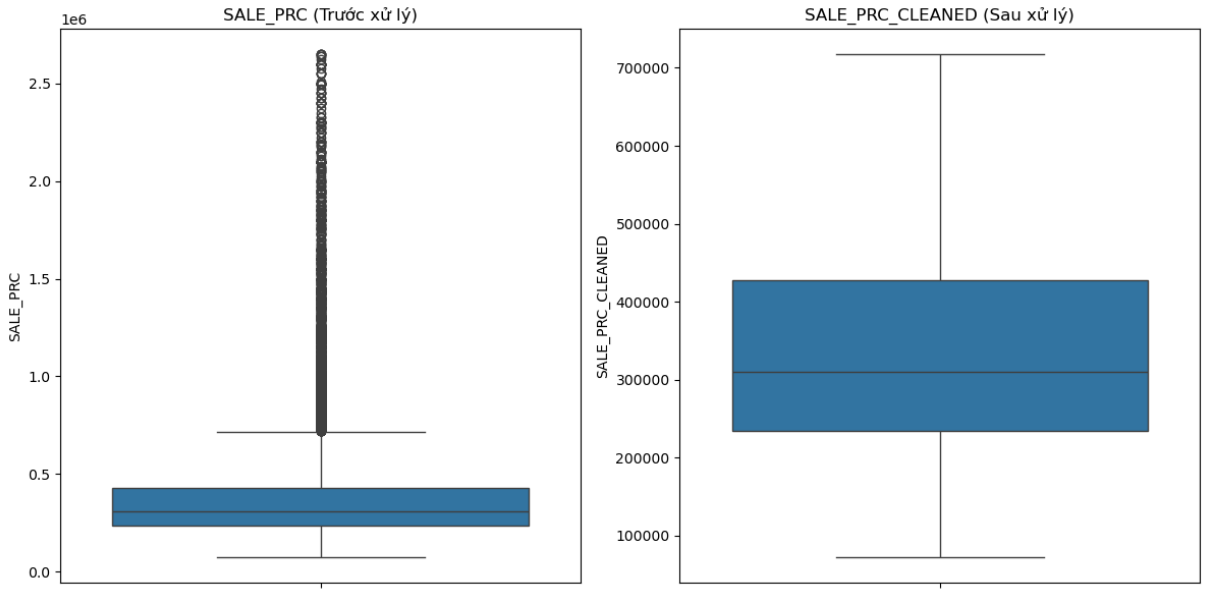
2. **Xử Lý Giá Trị Bất Thường (Outliers)**

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

```
    def handle_outliers(df, column):
"""Xu ly outlie bang IQR."""
print(f"\n4.XỬ LÝ OUTLIERS cho cột {column}")
Q1 = df[column].quantile(0.25)
Q3 = df[column].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers_count = df[(df[column] < lower_bound)
|(df[column] > upper_bound)].shape[0]
print(f"Số lượng outlier tìm thấy: {outliers_count}")
df[TARGET_COLUMN] = df[column].clip(lower_bound, upper_bound)
print(f"Đã tạo cột {TARGET_COLUMN} đã xử lý outliers.")
```

```
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



Hình 8: Biểu đồ Box Plot so sánh

Không còn các điểm bất thường (hình tròn nhỏ) phía trên râu nữa. Tất cả các giá trị trước đây là outliers (ví dụ: > 700k USD) đã được "kéo" về giá trị tối đa cho phép bởi "râu" trên (upper bound tính bằng Q3 + 1.5IQR).
Điều này không có nghĩa là các ngôi nhà đắt tiền bị xóa đi, mà giá trị của chúng trong cột `SALE_PRC_CLEANED` đã được giới hạn lại ở mức tối đa "hợp lý" theo phương pháp IQR.

3. **Tạo biến mới**
   Tạo các biến mới từ dữ liệu hiện có (hàm `feature_engineering`):

   - `PRICE_PER_SQFT`: Giá trên mỗi foot vuông đất (`SALE_PRC / LND_SQFOOT`).
   - `LIVING_LAND_RATIO`: Tỷ lệ diện tích sống trên diện tích đất (`TOT_LVG_AREA / LND_SQFOOT`).
   - `AVG_IMPORTANT_DIST`: Khoảng cách trung bình đến các điểm quan trọng (trung bình của `OCEAN_DIST, WATER_DIST, CNTR_DIST, HWY_DIST`).

```
6. TẠO BIẾN MỚI (Feature Engineering)
Đã tạo các biến: PRICE_PER_SQFT, LIVING_LAND_RATIO, AVG_IMPORTANT_DIST.
```

Hình 9: Tạo biến mới.

4. **Xử lý cột không cần thiết**

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

```
df.drop(columns=['PARCELNO'], inplace=True)
```

5. **Mã hoá biến phân loại**

   - Cột month_sold (1-12) và structure_quality (1-5) có thể coi là biến phân loại.
   - Sử dụng OneHotEncoder cho các biến phân loại:

```
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])
```

6. **Chuẩn hoá dữ liệu**

   - Các cột số như LND_SQFOOT, TOT_LVG_AREA, RAIL_DIST, v.v. có đơn vị và phạm vi khác nhau, cần chuẩn hóa (standardization) hoặc quy về $[0, 1]$ (min-max scaling).
   -

```
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])
```

7. **Tạo bộ tiền xử lý**

   - Tất cả được thực hiện trong một pipeline duy nhất để đảm bảo tính nhất quán
   - Lợi ích của pipeline: tự động hóa quy trình, tránh rò rỉ dữ liệu khi áp dụng trên tập huấn luyện và kiểm tra.

```
def create_preprocessor(numeric_features, categorical_features):
"""Tạo bộ tiền xử lý ColumnTransformer."""
print("\n9. TẠO BỘ TIỀN XỬ LÝ")
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])
```

```
    categorical_transformer = Pipeline(steps=[
        ('imputer', SimpleImputer(strategy='most_frequent')),
        ('onehot', OneHotEncoder(handle_unknown='ignore'))
    ])
    # cả 2 bộ tiền xử lý được kết hợp trong hàm ColumnTransformer
    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numeric_features),
            ('cat', categorical_transformer, categorical_features)
        ],
        remainder='passthrough' # Giữ lại các cột không được xử lý nếu có
    )
    print("Bộ tiền xử lý đã được tạo.")
    return preprocessor
```

8. **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%):

```
  print("\n8. CHIA DỮ LIỆU")
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=TEST_SIZE, random_state=RANDOM_STATE
)
```

```
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 10: Chia tập dữ liệu

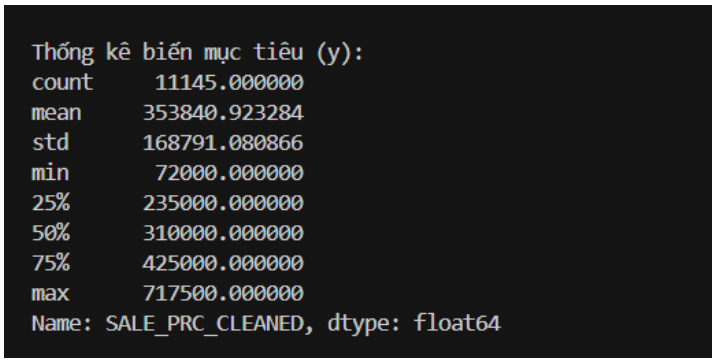
Sau khi hoàn tất tiền xử lý, dữ liệu sẽ sẵn sàng để đưa vào huấn luyện mô hình dự đoán giá nhà.

1.3. Thống kê dữ liệu mẫu

Mục tiêu: khám phá đặc điểm phân phối của biến mục tiêu và các biến độc lập, cũng như mối quan hệ giữa chúng, để có cái nhìn toàn diện về dữ liệu mà mô hình sẽ học. Các phân tích dưới đây được thực hiện trên tập dữ liệu huấn luyện (80% dữ liệu gốc).
Hàm `analyze_training_data(X_train, y_train)` được sử dụng để thực hiện các phân tích này).

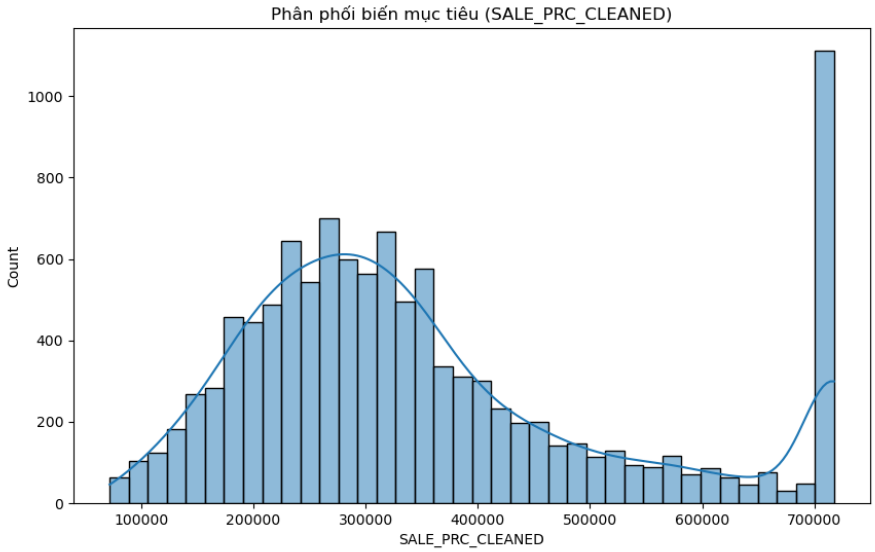
1. **Phân tích biến mục tiêu** `y_train`

    - In kết quả của `y.describe()` bao gồm các giá trị: trung bình (mean), độ lệch chuẩn (std), giá trị nhỏ nhất (min), các tứ phân vị (25%, 50% - median, 75%), và giá trị lớn nhất (max).



Hình 11: Thống kê mô tả biến mục tiêu (y).

    - Phân phối biến mục tiêu:



Hình 12: Biểu đồ histogram phân phối biến mục tiêu.

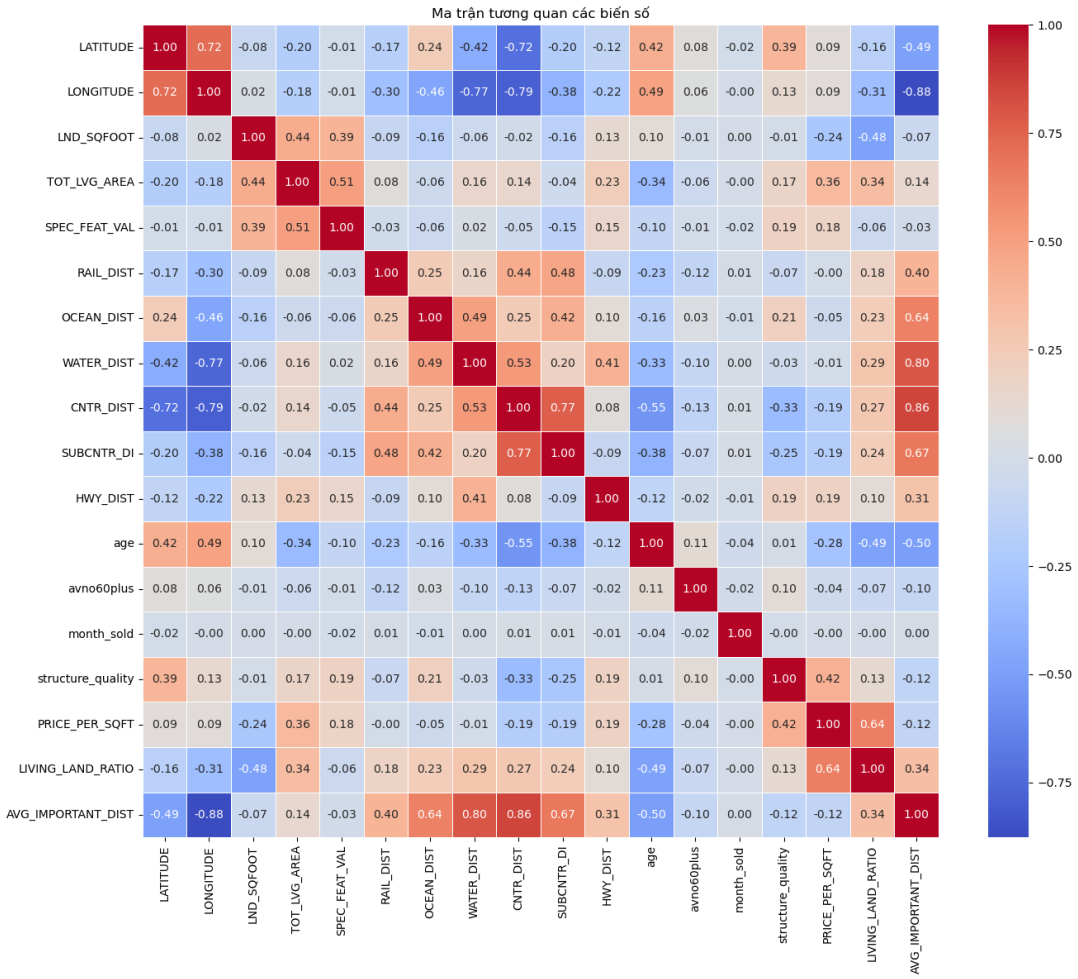
2. **Phân tích biến số** `X_train`

- In thống kê mô tả cho tất cả các biến số:

```
Thống kê các biến số:
          LATITUDE     LONGITUDE  ...  LIVING_LAND_RATIO  AVG_IMPORTANT_DIST
count  11145.000000  11145.000000  ...       11145.000000        11145.000000
mean      25.728856    -80.327866  ...           0.285971        30040.033874
std        0.140410      0.089321  ...           0.140787        12532.741618
min       25.434333    -80.542172  ...           0.021579         3265.650000
25%       25.621602    -80.404320  ...           0.183215        19738.200000
50%       25.731857    -80.339501  ...           0.251662        28674.125000
75%       25.852156    -80.258330  ...           0.363801        38959.700000
max       25.974382    -80.119746  ...           1.191853        69179.125000

[8 rows x 18 columns]
```

Hình 13: Thống kê mô tả các biến số

- Ma trận tương quan:



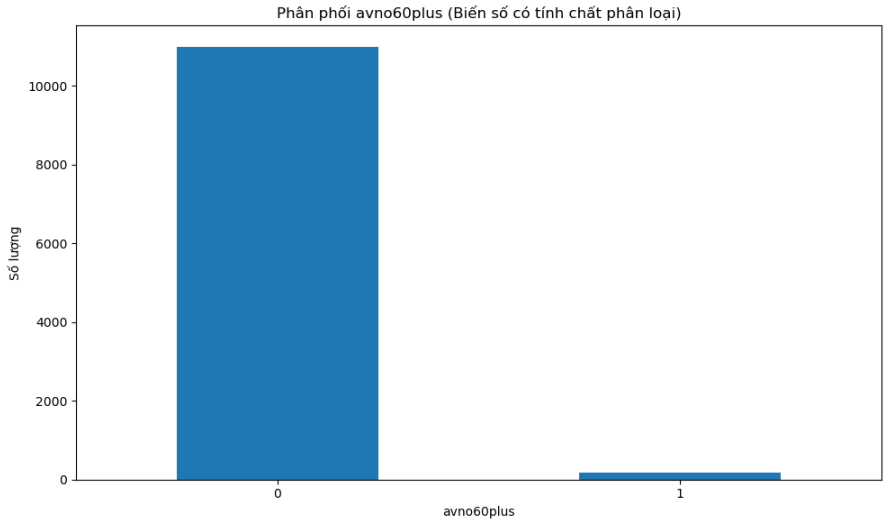
Hình 14: Ma trận tương quan các biến số.

3. **Phân tích các biến phân loại**

- Output thông báo không có biến phân loại:

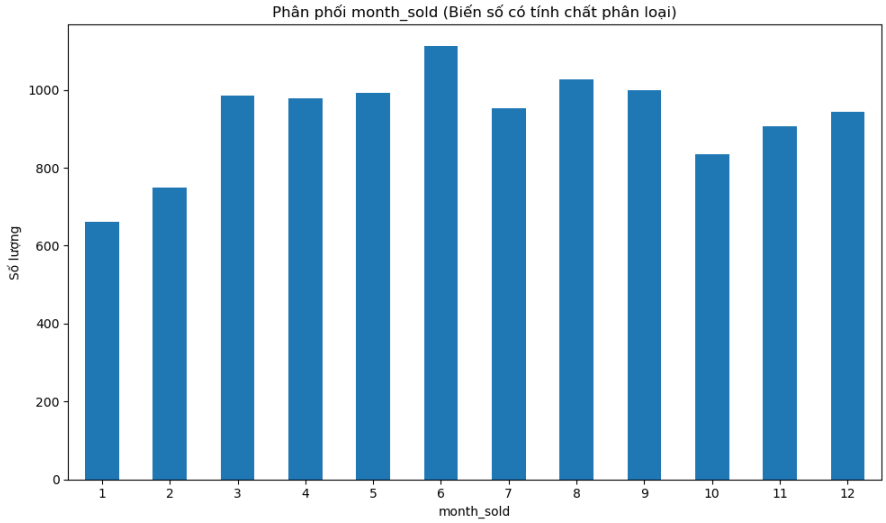
```
Danh sách các biến phân loại:
[]

Không có biến phân loại nào trong dataset
Tất cả các biến đều là biến số (numeric)
```

- Các biến có tính chất phân loại về mặt ý nghĩa: `avno60plus`, `month_sold`, `structure_quality`.
- Phân phối `avno60plus`:



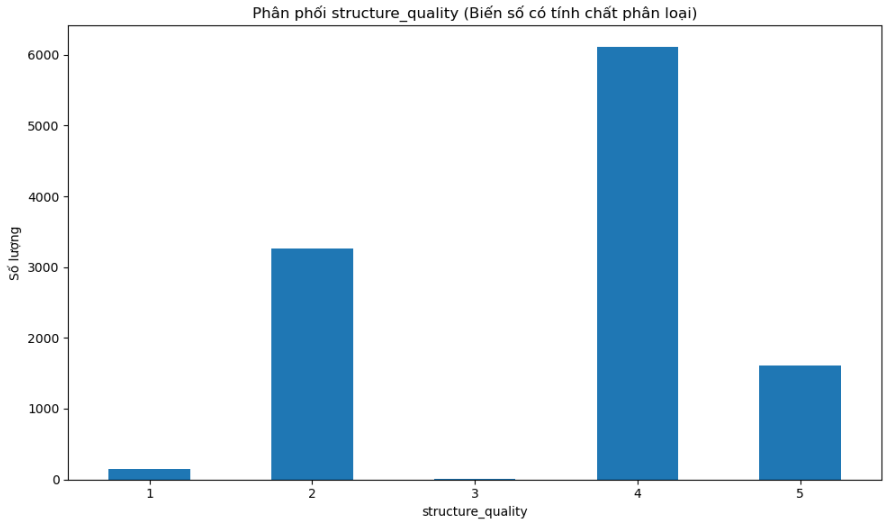
Hình 15: Phân phối `avno60plus`.

- Phân phối `month_sold`:



Hình 16: Phân phối `month_sold`.

- Phân phối `structure_quality`:

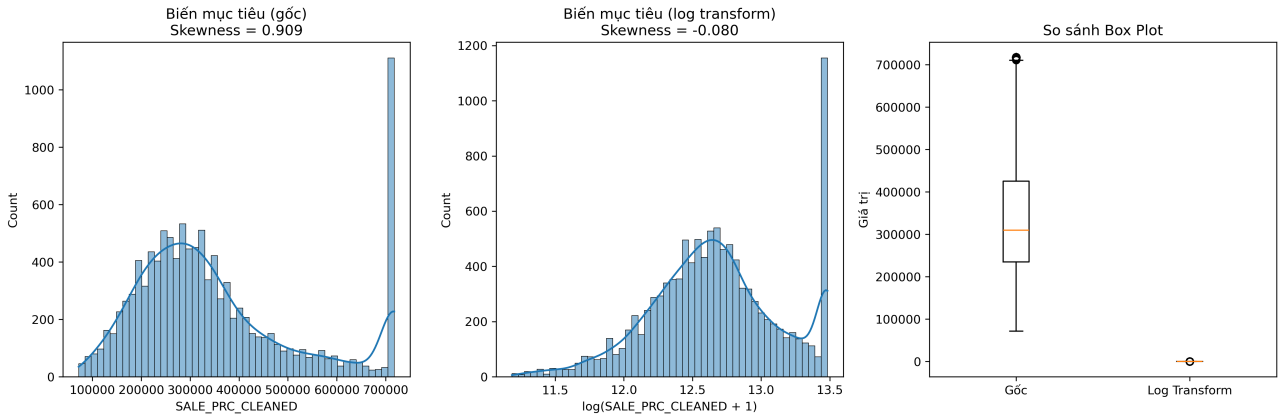


Hình 17: Phân phối `structure_quality`.

1.4. Chuyển đổi dữ liệu

Để đảm bảo dữ liệu phù hợp với giả định của mô hình hồi quy, chúng ta thực hiện chuyển đổi các biến có phân phối lệch bằng log transform.

1. **Chuyển đổi biến mục tiêu**
    - Áp dụng log transform cho. biến mục tiêu `SALE_PRC_CLEANED`
    - Độ lệch ban đầu của biến mục tiêu: 0.909,
    - Độ lệch sau log transform: -0.080,
    - Cải thiện độ lệch: 0.829.



Hình 18: So sánh phân phối biến mục tiêu trước và sau log transform

2. **Chuyển đổi các biến độc lập**

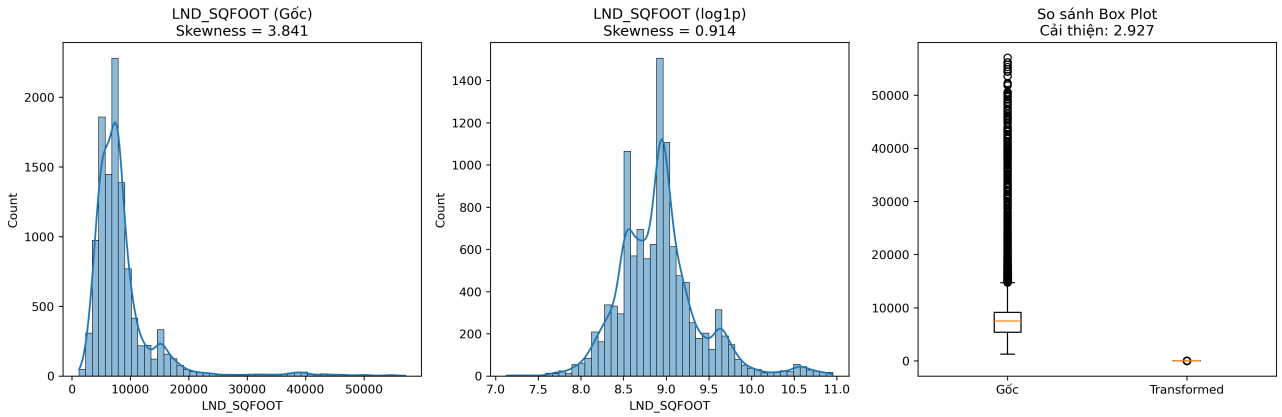
   Áp dụng log transform cho các biến có $|\text{skewness}| > 1.0$

```
=== TỔM TẮT KẾT QUẢ CHUYỂN ĐỔI ===
Số biến đã chuyển đổi: 8/18

Chi tiết các biến đã chuyển đổi:
------------------------------------------------------------------------
Biến                Skew gốc   Skew mới   Cải thiện  Phép biến đổi
------------------------------------------------------------------------
LND_SQFOOT          3.841      0.914      2.927      log1p
TOT_LVG_AREA        1.349      0.295      1.054      log1p
SPEC_FEAT_VAL       1.958      -1.159     0.799      log1p(x + 1)
WATER_DIST          1.127      -1.671     -0.543     log1p(x + 1.0)
HWY_DIST            1.102      -0.883     0.219      log1p
avno60plus          7.911      7.911      0.000      log1p(x + 1)
PRICE_PER_SQFT      2.957      0.208      2.749      log1p
LIVING_LAND_RATIO   1.315      0.934      0.381      log1p
```

Hình 19: Kết quả chuyển đổi các biến độc lập

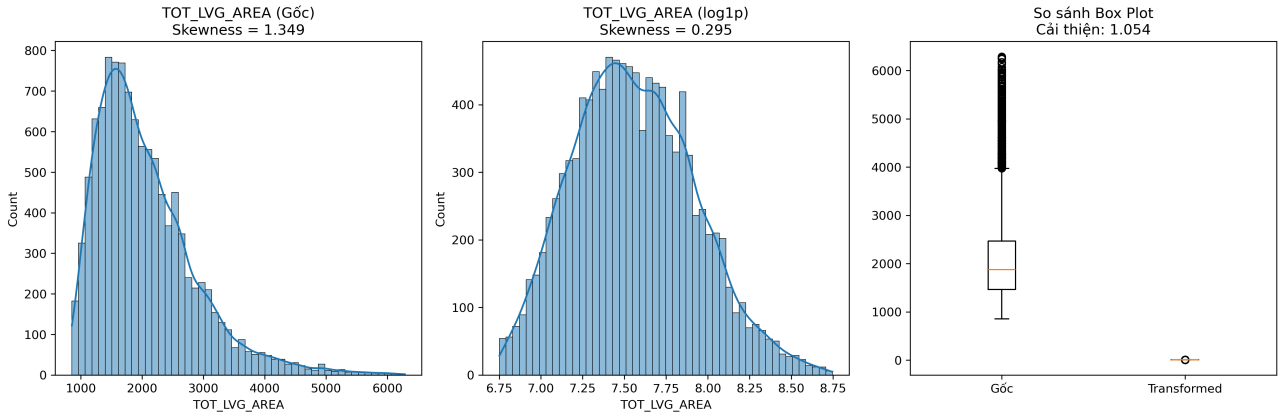
Vẽ biểu đồ cho 3 biến quan trọng:

- `LND_SQFOOT`



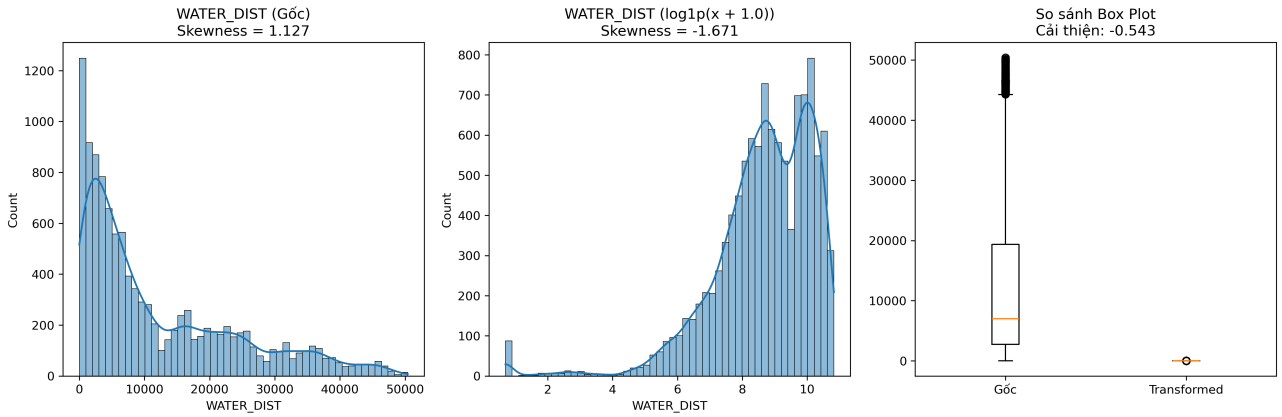
Hình 20: Chuyển đổi biến `LND_SQFOOT`.

- `TOT_LVG_AREA`



Hình 21: Chuyển đổi biến `TOT_LVG_AREA`.

- `WATER_DIST`



Hình 22: Chuyển đổi biến `WATER_DIST`.

3. **Tóm tắt kết quả**

- Đã chuyển đổi biến mục tiêu và 8/18 biến độc lập.
- Tất cả các biến đều có độ lệch được cải thiện.
- Dữ liệu sau chuyển đổi không chứa giá trị NaN.
- Đảm bảo tính nhất quán giữa tập train và test.
- Dữ liệu đã sẵn sàng cho việc huấn luyện mô hình

Phần 2. Đánh giá mô hình

2.1. Đề xuất và lựa chọn các tiêu chí đánh giá