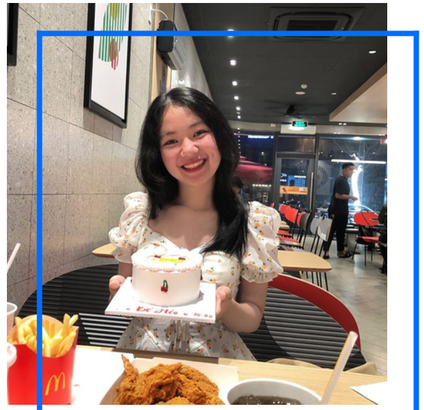
DỰ ĐOÁN GIÁ NHÀ
BẰNG HỒI QUY ĐA BIẾN

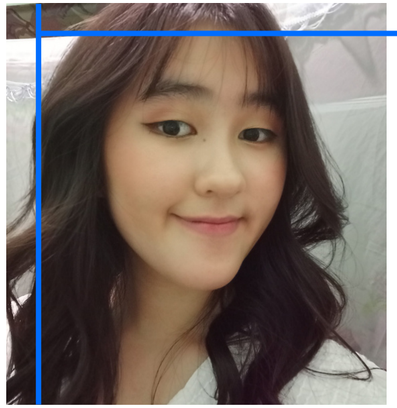
GVHD: TS.LÊ DIÊN TUẤN
NHÓM 10 - 47K21.2

THÀNH VIÊN



Nguyễn Thị Thanh Hằng

Trần Thị Hằng



Cai Thị Thùy Kiên

Bùi Lê Khánh Vy

NỘI DUNG

1.Cơ sở lý thuyết

2. Framework

3. Chuẩn bị dữ liệu

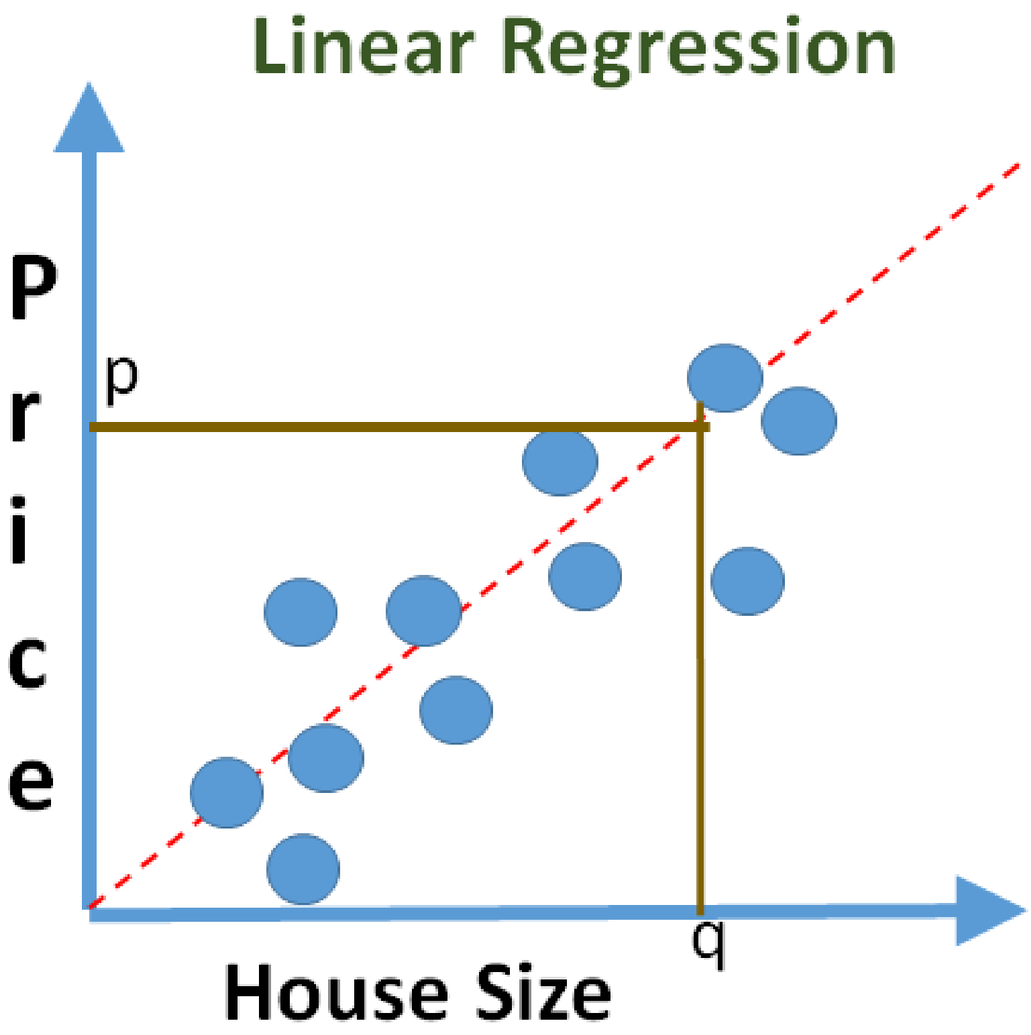
4. Tiền xử lý

5. Xây dựng mô hình hồi quy

6. Đánh giá mô hình

1

CƠ SỞ LÝ THUYẾT



1

Hồi quy tuyến tính đa biến

- Khái niệm
- Y = β0 + β1X1 + β2X2 + ... + βnXn

2

Phương pháp ước lượng

- Bình phương nhỏ nhất(OLS)

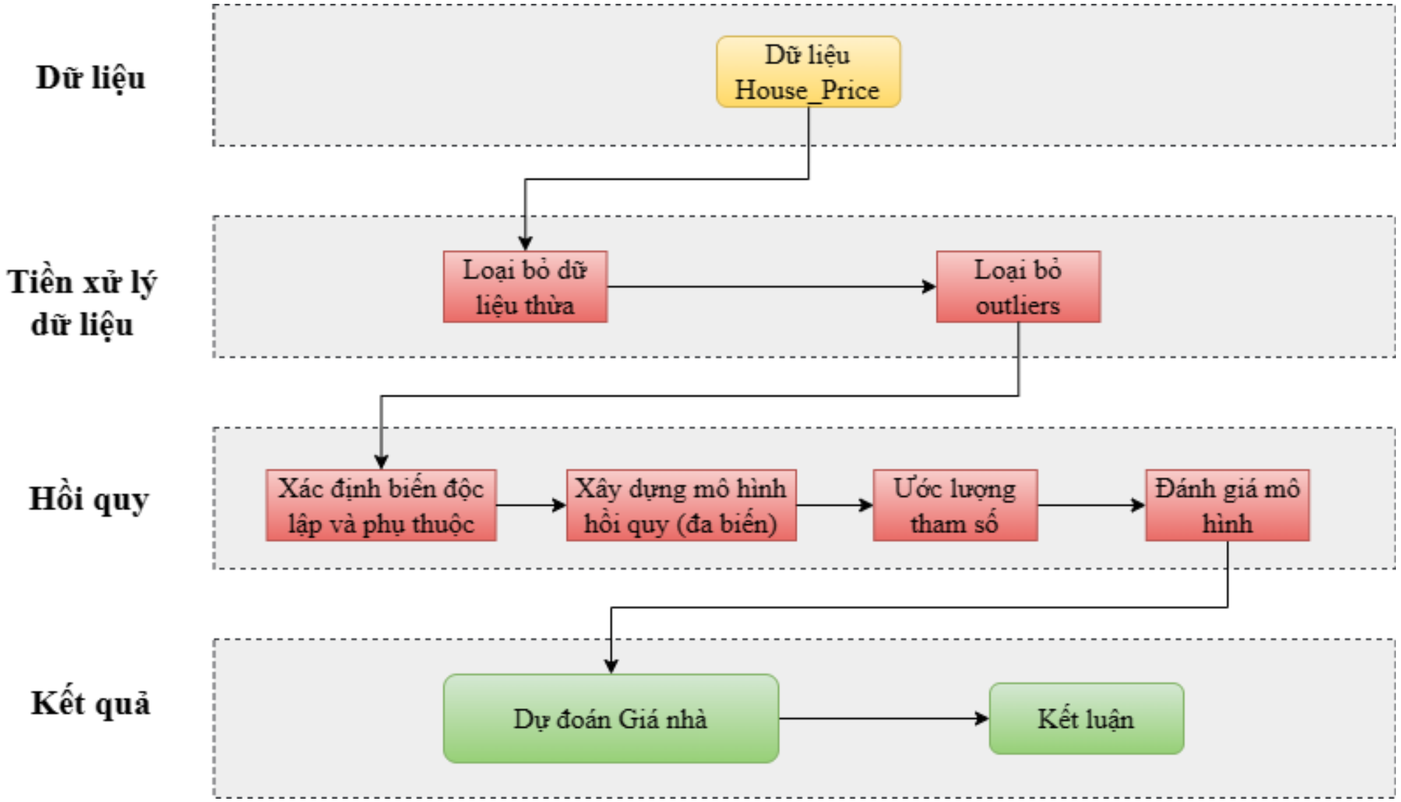
3

Đánh giá mô hình

- Mean Squared Error
- Bình phương của R (R square)
- Kiểm định t-test và F-test

2

FRAMEWORK


Dữ liệu
Dữ liệu House_Price
Tiền xử lý dữ liệu
Loại bỏ dữ liệu thừa
Loại bỏ outliers
Hồi quy
Xác định biến độc lập và phụ thuộc
Xây dựng mô hình hồi quy (đa biến)
Ước lượng tham số
Đánh giá mô hình
Kết quả
Dự đoán Giá nhà
Kết luận

3

CHUẨN BỊ DỮ LIỆU

CHUẨN BỊ DỮ LIỆU

House_price dataset

date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_baseme	yr_built	yr_renovated	street	city	statezip	country
2014-05-02 00:00:	313000	3	2	1340	7912	2	0	0	3	1340	0	1955	2005	18810 Den	Shoreline	WA 98133	USA
2014-05-02 00:00:	2384000	5	3	3650	9050	2	0	4	5	3370	280	1921	0	709 W Bla	Seattle	WA 98119	USA
2014-05-02 00:00:	342000	3	2	1930	11947	1	0	0	4	1930	0	1966	0	26206-262	Kent	WA 98042	USA
2014-05-02 00:00:	420000	3	2	2000	8030	1	0	0	4	1000	1000	1963	0	857 170th	Bellevue	WA 98008	USA
2014-05-02 00:00:	550000	4	3	1940	10500	1	0	0	4	1140	800	1976	1992	9105 170th	Redmond	WA 98052	USA
2014-05-02 00:00:	490000	2	1	880	6380	1	0	0	3	880	0	1938	1994	522 NE 88	Seattle	WA 98115	USA
2014-05-02 00:00:	335000	2	2	1350	2560	1	0	0	3	1350	0	1976	0	2616 174th	Redmond	WA 98052	USA
2014-05-02 00:00:	482000	4	3	2710	35868	2	0	0	3	2710	0	1989	0	23762 SE	Maple Vall	WA 98038	USA
2014-05-02 00:00:	452500	3	3	2430	88426	1	0	0	4	1570	860	1985	0	46611-466	North Bend	WA 98045	USA
2014-05-02 00:00:	640000	4	2	1520	6200	2	0	0	3	1520	0	1945	2010	6811 55th	Seattle	WA 98115	USA
2014-05-02 00:00:	463000	3	2	1710	7320	1	0	0	3	1710	0	1948	1994	Burke-Gilm	Lake Fores	WA 98155	USA
2014-05-02 00:00:	1400000	4	3	2920	4000	2	0	0	5	1910	1010	1909	1988	3838-4098	Seattle	WA 98105	USA
2014-05-02 00:00:	588500	3	2	2330	14892	1	0	0	3	1970	360	1980	0	1833 220th	Sammamis	WA 98074	USA
2014-05-02 00:00:	365000	3	1	1090	6435	1	0	0	4	1090	0	1955	2009	2504 SW F	Seattle	WA 98106	USA
2014-05-02 00:00:	1200000	5	3	2910	9480	2	0	0	3	2910	0	1939	1969	3534 46th	Seattle	WA 98105	USA
2014-05-02 00:00:	242500	3	2	1200	9720	1	0	0	4	1200	0	1965	0	14034 SE	Kent	WA 98042	USA
2014-05-02 00:00:	419000	3	2	1570	6700	1	0	0	4	1570	0	1956	0	15424 SE	Bellevue	WA 98007	USA
2014-05-02 00:00:	367500	4	3	3110	7231	2	0	0	3	3110	0	1997	0	11224 SE	Auburn	WA 98092	USA
2014-05-02 00:00:	257950	3	2	1370	5858	1	0	0									

3

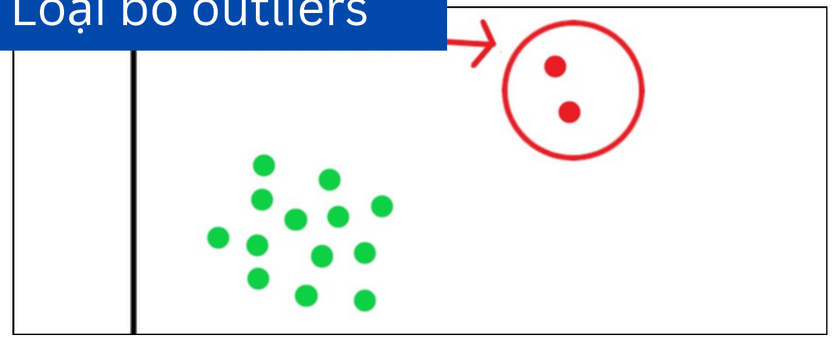
TIỀN XỬ LÝ DỮ LIỆU

TIỀN XỬ LÝ DỮ LIỆU

Loại bỏ dữ liệu thừa

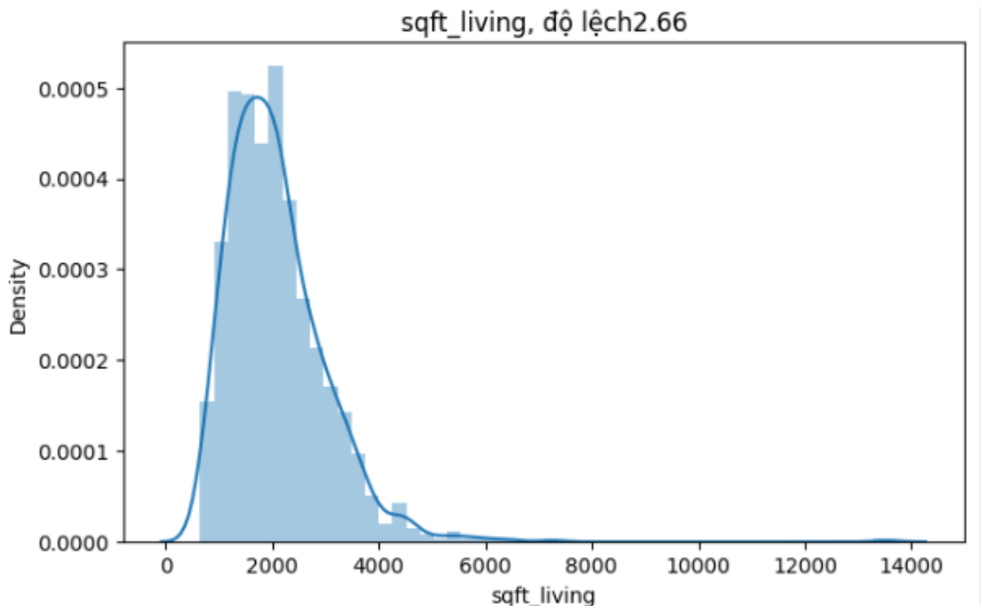
Xóa bỏ các cột dữ liệu thừa, chỉ giữ lại những dữ liệu cần thiết để xây dựng mô hình hồi quy [price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, sqft_above, sqft_basement]

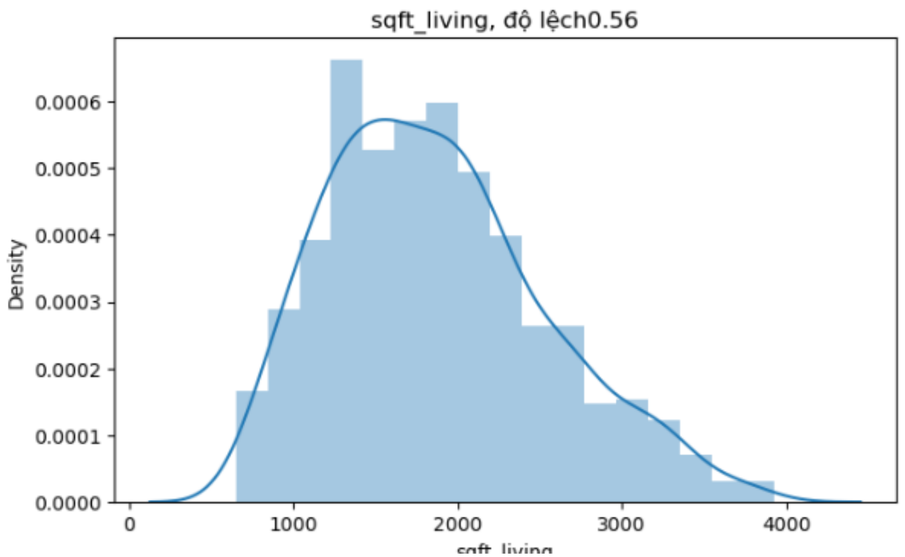
Loại bỏ outliers

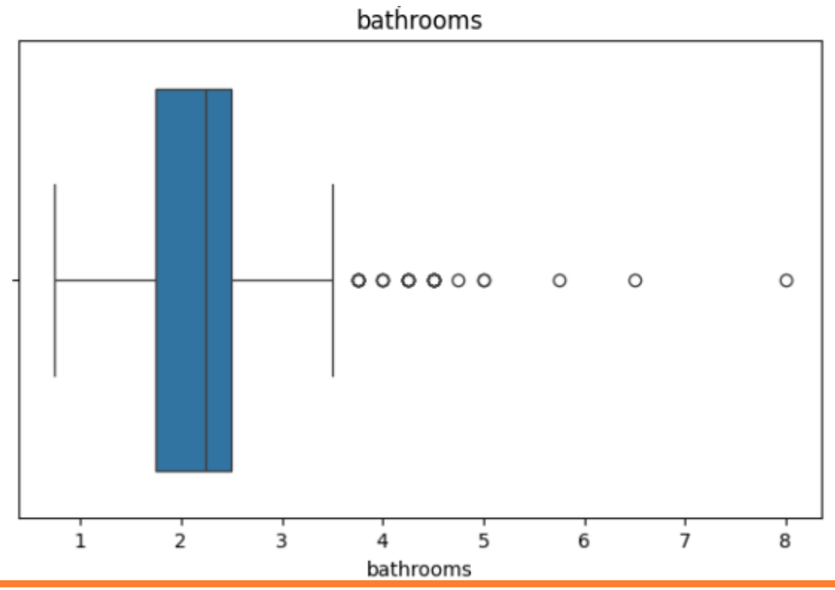


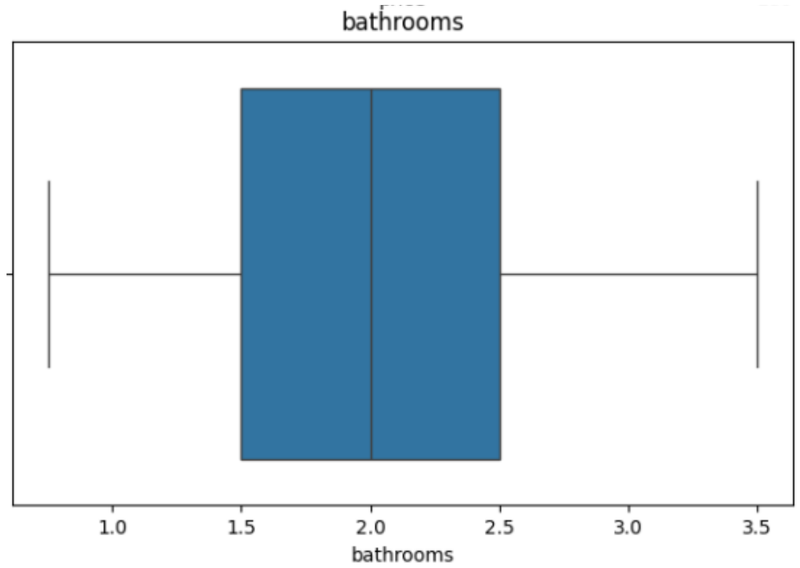
Sử dụng phương pháp IQR để loại bỏ những dữ liệu ngoại lai tồn tại trong tập dữ liệu để giảm thiểu sai sót cho mô hình

LOẠI BỎ OUTLIERS

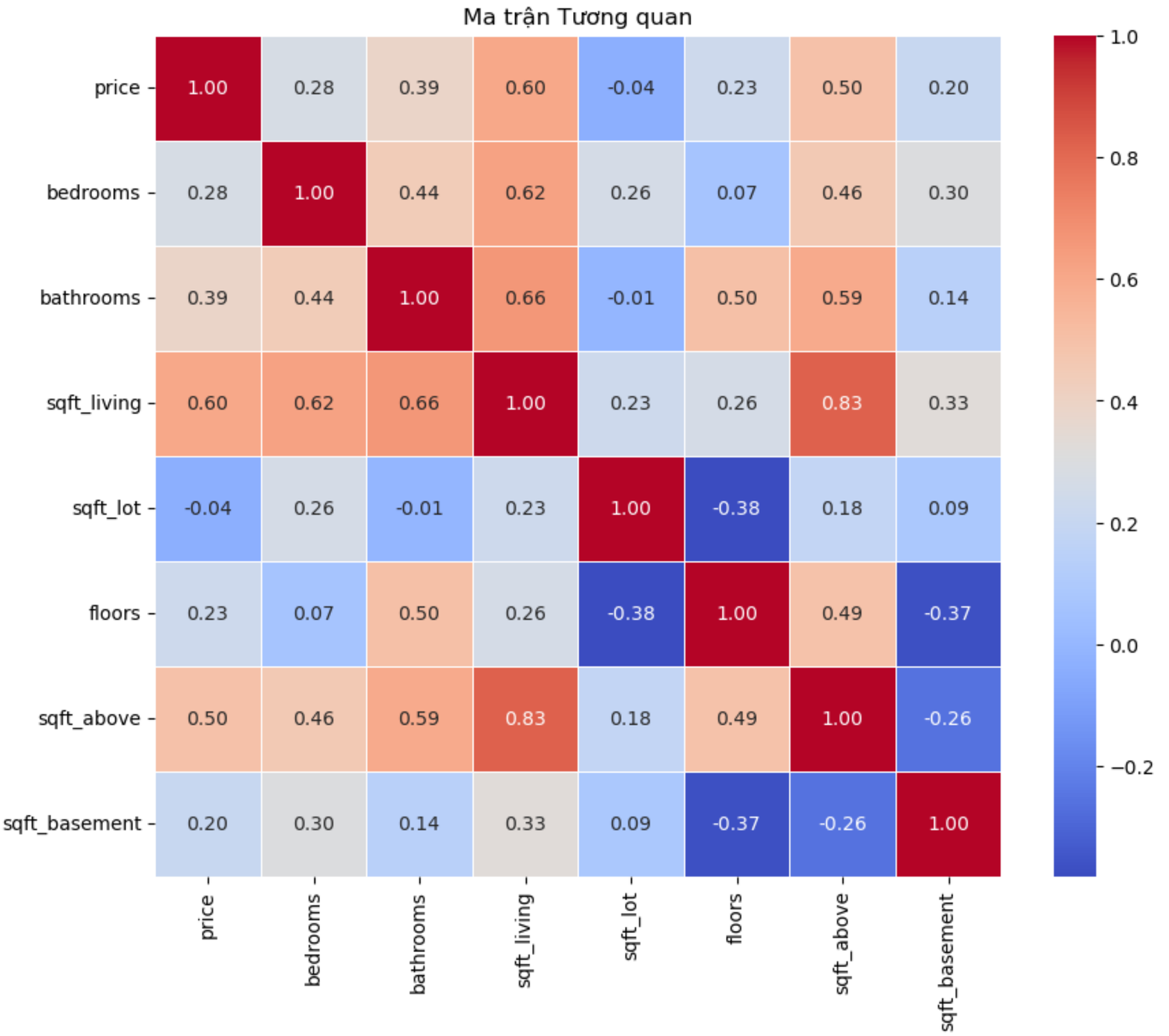




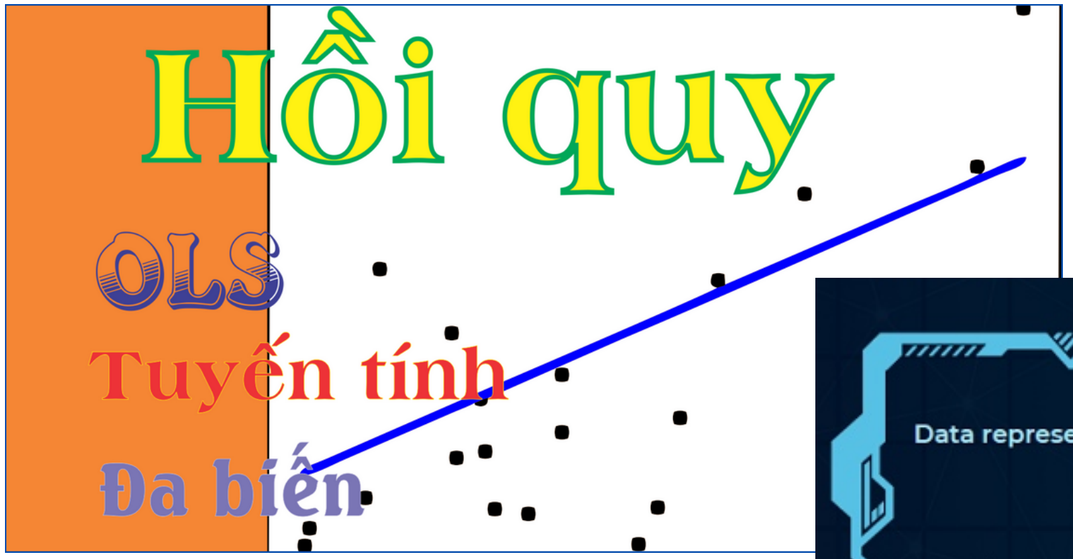


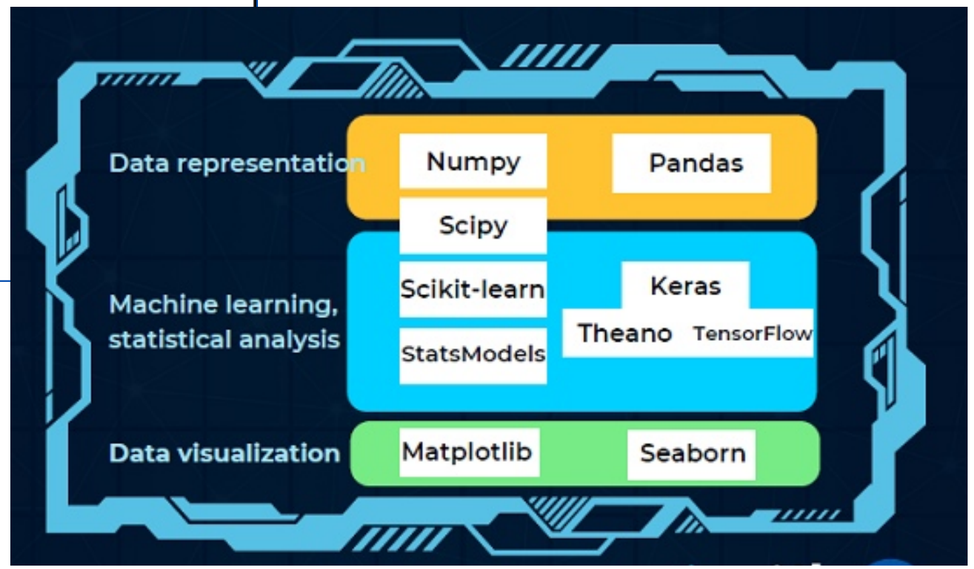


MA TRẬN TƯƠNG QUAN



5 MÔ HÌNH HỒI QUY ĐA BIẾN





5.1 XÂY DỰNG MÔ HÌNH HỒI QUY ĐA BIẾN

Bước 1: Chọn biến độc lập và biến phụ thuộc

```
x = data_drop[['bedrooms', 'bathrooms',
'sqft_living','sqft_lot','floors','sqft_above','sqft_basement']]

y = data_drop['price']
```

Bước 2: Chia dữ liệu thành 2 phần:

- Data_train: dữ liệu dùng để huấn luyện mô hình hồi quy
- Data_test: dữ liệu dùng để kiểm tra lại mô hình đã được huấn luyện

```
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=42)
```

Bước 3: Xây dựng mô hình hồi quy tuyến tính trên tập dữ liệu huấn luyện

```
lm = LinearRegression()

lm.fit(x_train, y_train)
```

5.1 ƯỚC LƯỢNG THAM SỐ

Hệ số chặn

```
print('Hệ số chặn (intercept):', lm.intercept_)
```

```
Hệ số chặn (intercept): 92559.794299978
```

Hệ số hồi quy của từng biến độc lập

```
pd.DataFrame(lm.coef_, x_train.columns)
```

bedrooms	-24397.662463
bathrooms	-39868.203975
sqft_living	186.644751
sqft_lot	-1.061603
floors	34173.210806
sqft_above	90.708300
sqft_basement	95.936451

NHẬN XÉT

- **Mối quan hệ với biến phụ thuộc ?**
- **Mức độ ảnh hưởng đến biến phụ thuộc ?**
- **Ý nghĩa ?**

6

ĐÁNH GIÁ MÔ HÌNH

DỰ BÁO GIÁ DỰA VÀO TẬP TEST

```python
# Dự báo giá dựa vào tập test
y_pred = lm.predict(x_test)

# Tạo DataFrame từ dự đoán
y_pred_df = pd.DataFrame(y_pred, columns=['Price_predict'], index=x_test.index)

# Kết hợp các DataFrame và so sánh kết quả
result_compare = pd.concat([x_test, y_test, y_pred_df], axis=1)
result_compare['Deviation'] = result_compare['Price_predict'] - result_compare['price']

# Hiển thị kết quả
result_compare
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	sqft_above	sqft_basement	price	Price_predict	Deviation
521	2	2.50	1480	1369	3.0	1480	0	420000	455642.772675	35642.772675
737	5	1.75	2000	3750	2.0	2000	0	585000	519873.637599	-65126.362401
740	4	3.00	3150	9130	2.0	3150	0	695000	807680.630427	112680.630427
660	4	1.75	2010	8625	1.0	1340	670	364000	511199.167947	147199.167947
411	3	1.50	1500	6337	1.0	1500	0	270000	403039.910969	133039.910969
...	...	...	...	...	...	...	...	...	...	...
408	2	1.00	1840	3300	1.5	1130	710	490000	565694.393526	75694.393526
332	4	2.50	1810	6509	2.0	1810	0	270000	458744.105374	188744.105374
208	3	1.00	1230	9568	1.0	1230	0	387000	344658.650682	-42341.349318
613	3	1.00	1390	12823	1.0	1390	0	319000	385579.621739	66579.621739
78	4	2.50	2120	10202	1.0	1620	500	755000	509243.917140	-245756.082860

Đây là kết quả dự đoán giá dựa vào tập test và sau đó so sánh kết quả dự đoán với giá thực tế, từ đó đánh giá hiệu suất của mô hình trên dữ liệu

ĐÁNH GIÁ MÔ HÌNH SAU KHI TEST

```python
# Đánh giá mô hình sau khi test
# Tính toán MSE
mse_test = mean_squared_error(y_test, y_pred)

# Tính toán R^2
r2_test = r2_score(y_test, y_pred)

# Hiển thị kết quả
print("Mean Squared Error trên tập kiểm tra:", mse_test)
print("R-squared trên tập kiểm tra:", r2_test)
```

```
Mean Squared Error trên tập kiểm tra: 34281366193.473965
R-squared trên tập kiểm tra: 0.5823224427313729
```

- Sử dụng MSE để đo lường trung bình của bình phương sai số giữa giá trị dự đoán và giá trị thực tế trên tập kiểm tra.
- Sau đó tiến hành tính toán R^2 để đánh giá khả năng giải thích của mô hình trên cùng tập dữ liệu này

KIỂM ĐỊNH T-TEST

```python
# Kiểm định t-test cho từng biến độc lập
for column in x_test.columns:
    t_stat, p_val = stats.ttest_ind(result_compare[column], result_compare['Deviation'], equal_var=False)
    print(f'Biến độc lập "{column}":')
    print(f'  - Giá trị t-score: {t_stat}')
    print(f'  - Giá trị p-value: {p_val}')
    if p_val < 0.05:
        print('    => Có đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.')
    else:
        print('    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.')
    print()
```

✓ 0.0s

```
Biến độc lập "bedrooms":
  - Giá trị t-score: -0.22172318687367107
  - Giá trị p-value: 0.8247567422547195
    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.

Biến độc lập "bathrooms":
  - Giá trị t-score: -0.22181948379586366
  - Giá trị p-value: 0.8246818779693934
    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.

Biến độc lập "sqft_living":
  - Giá trị t-score: -0.07031657324902088
  - Giá trị p-value: 0.9440123002423422
    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.

Biến độc lập "sqft_lot":
  - Giá trị t-score: 0.989312055649631
  - Giá trị p-value: 0.3236057130420199
    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.

Biến độc lập "floors":
  - Giá trị t-score: -0.22186758463347006
  - Giá trị p-value: 0.8246444834524482
    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.
...
  - Giá trị t-score: -0.19907355005786914
  - Giá trị p-value: 0.8424084593870829
    => Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng.
```

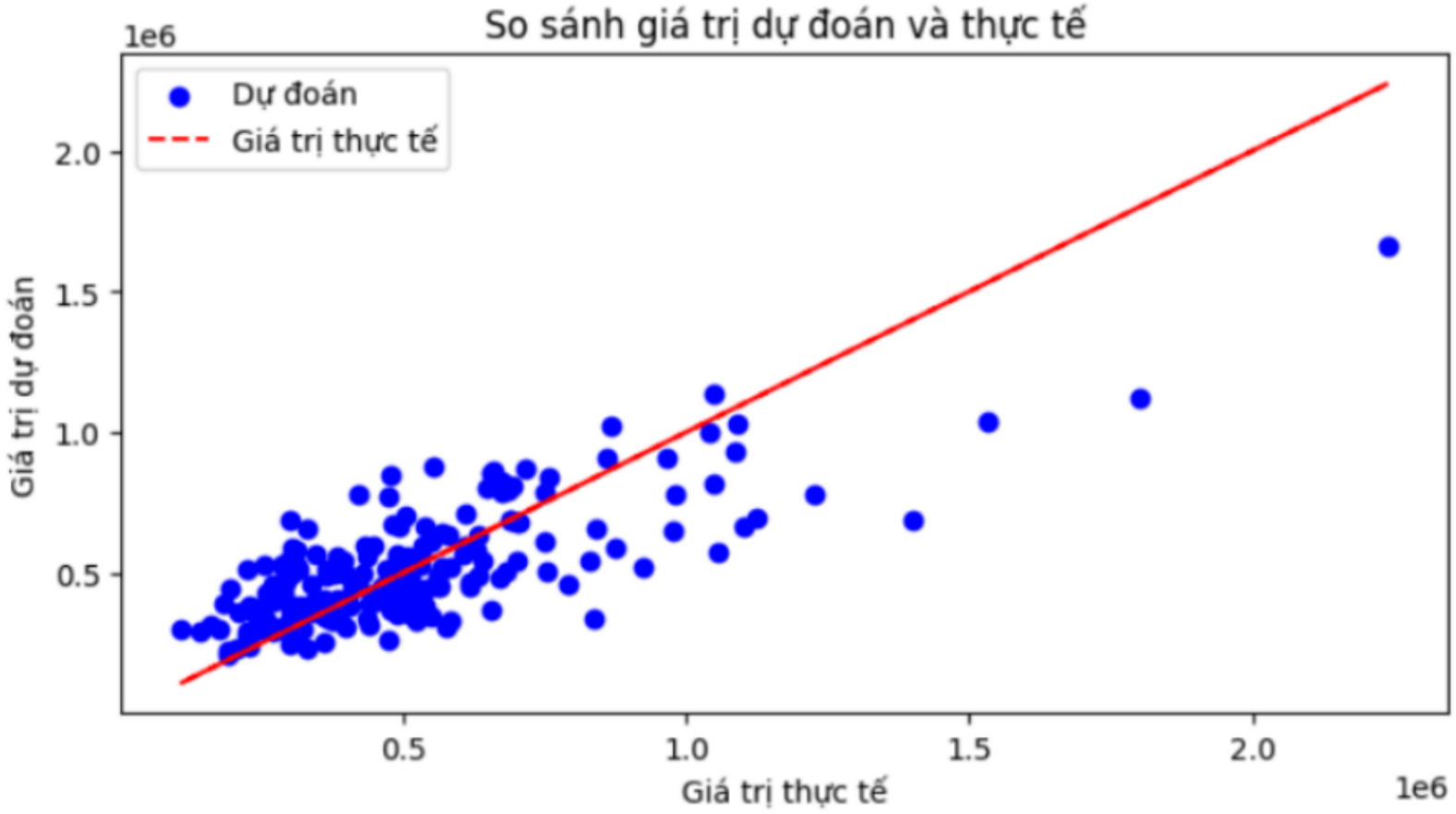
Từ kết quả đánh giá t-test trên, ta thấy tất cả các biến độc lập đều có mức ảnh hưởng nhất định đến biến phụ thuộc.

KIỂM ĐỊNH F-TEST

```python
from scipy.stats import f

# Số lượng biến độc lập
k = x_train.shape[1]

# Số lượng quan sát
n = len(y_train)

# Độ tự do của mô hình
df_model = k - 1

# Độ tự do của sai số
df_residual = n - k

# Giá trị của F-statistic
f_statistic = (lm.score(x_train, y_train) / df_model) / ((1 - lm.score(x_train, y_train)) / df_residual)

# Giá trị p-value
p_value = f.cdf(f_statistic, df_model, df_residual)

print(f'Giá trị F-statistic: {f_statistic}')
print(f'Giá trị p-value: {p_value}')
if p_value < 0.05:
    print('=> Có đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng của bất kỳ biến độc lập nào.')
else:
    print('=> Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng của bất kỳ biến độc lập nào.')
```

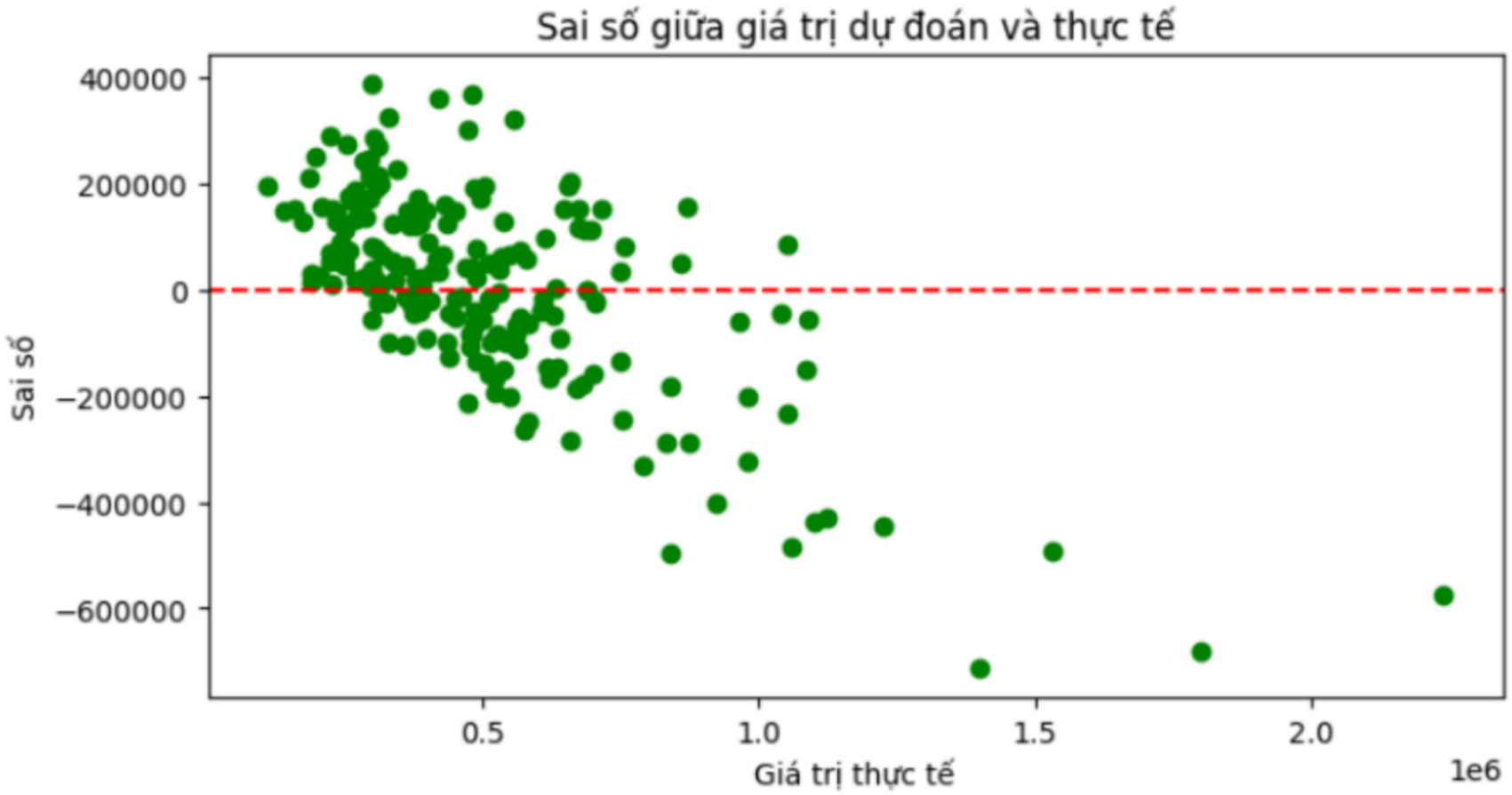
```
Giá trị F-statistic: 131.9905885771572
Giá trị p-value: 0.9999999999999999
=> Không đủ bằng chứng để bác bỏ giả thiết không có ảnh hưởng của bất kỳ biến độc lập nào.
```

BIỂU ĐỒ SCATTER PLOT



Từ kết quả biểu đồ có thể thấy các điểm dữ liệu được biểu diễn bằng các chấm màu xanh. Đường đỏ nét đứt biểu diễn xu hướng của giá trị thực tế

BIỂU ĐỒ SCATTER PLOT



Từ kết quả biểu đồ có thể thấy các điểm dữ liệu được biểu diễn bằng các chấm xanh lá cây tập trung xung quanh đường đỏ điều này cho thấy mô hình có khả năng dự đoán khá chính xác với một số mẫu trong tập test

THANK YOU