Swiss Coding Academy

FINAL PROJECT

Tên dự án: Dự báo giá kim cương

Chủ trì: Lưu Quang Huy

Ngày: 30/06/2025

Mục lục

I. Tổng quan dự án

- Xây dựng mô hình học máy để dự đoán giá kim cương dựa trên các đặc trưng vật lý và chất lượng.
- Hỗ trợ người dùng hoặc doanh nghiệp ước lượng giá trị kim cương một cách nhanh chóng và chính xác.
- So sánh hiệu quả giữa các mô hình hồi quy khác nhau để tìm ra mô hình dự báo tối ưu.
- Ứng dụng vào thị trường kim cương, thương mại điện tử, hoặc định giá sản phẩm tự động.

Dữ liệu được lấy từ Kaggle: Dataset gồm 50.000 mẫu, với các biến:

Tên biến	Mô tả
`carat`	**Trọng lượng kim cương** (tính bằng carat)
`cut`	**Chất lượng cắt** của viên kim cương (`Fair` , `Good` , `Very Good` , `Premium` , `Ideal`)
`color`	**Màu sắc kim cương** — từ `J` (kém nhất) đến `D` (tốt nhất)
`clarity`	**Độ trong suốt của kim cương**, theo thứ tự từ kém đến tốt: `I1` , `SI2` , `SI1` , `VS2` , `VS1` , `VVS2` , `VVS1` , `IF`
`x`	**Chiều dài** của kim cương (mm)
`y`	**Chiều rộng** của kim cương (mm)
`z`	**Độ sâu** của kim cương (mm)
`depth`	**Tỷ lệ chiều sâu**, tính bằng công thức: `z / mean(x, y)`
`table`	**Đường kính phần trên cùng rộng nhất** của kim cương (%)
`price`	**Giá của kim cương** (USD) — *biến mục tiêu cần dự đoán*

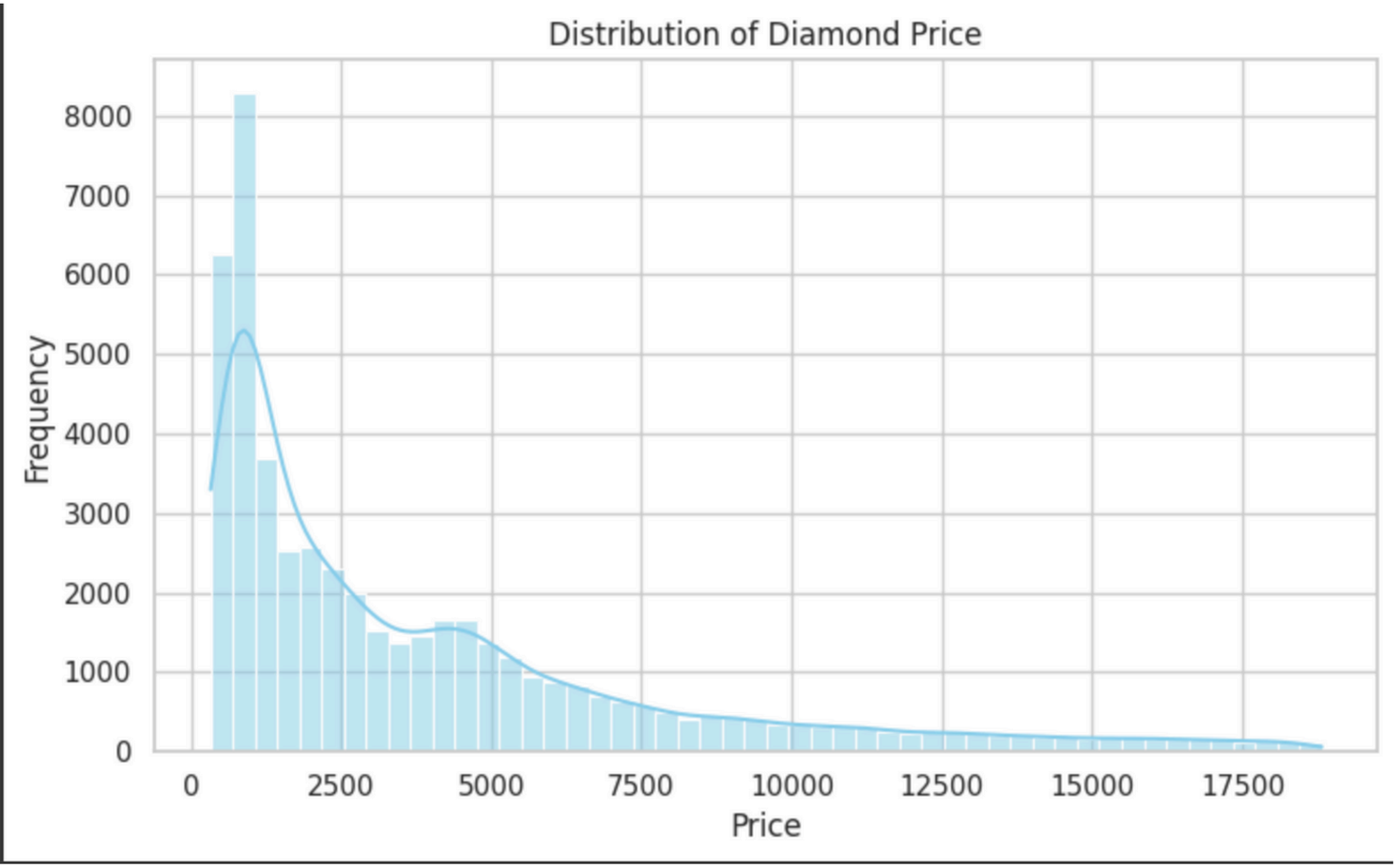
II. Cleaning Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   carat    50000 non-null  float64
 1   cut      50000 non-null  object 
 2   color    50000 non-null  object 
 3   clarity  50000 non-null  object 
 4   depth    50000 non-null  float64
 5   table    50000 non-null  float64
 6   price    50000 non-null  int64  
 7   x        50000 non-null  float64
 8   y        50000 non-null  float64
 9   z        50000 non-null  float64
dtypes: float64(6), int64(1), object(3)
memory usage: 3.8+ MB
```

- Loại bỏ Duplicate: 126 dòng
- Kiểm tra Missing Data: 0 dòng
- Loại bỏ Outlier: Thực hiện ở EDA

III. EDA

Phân bổ giá kim cương

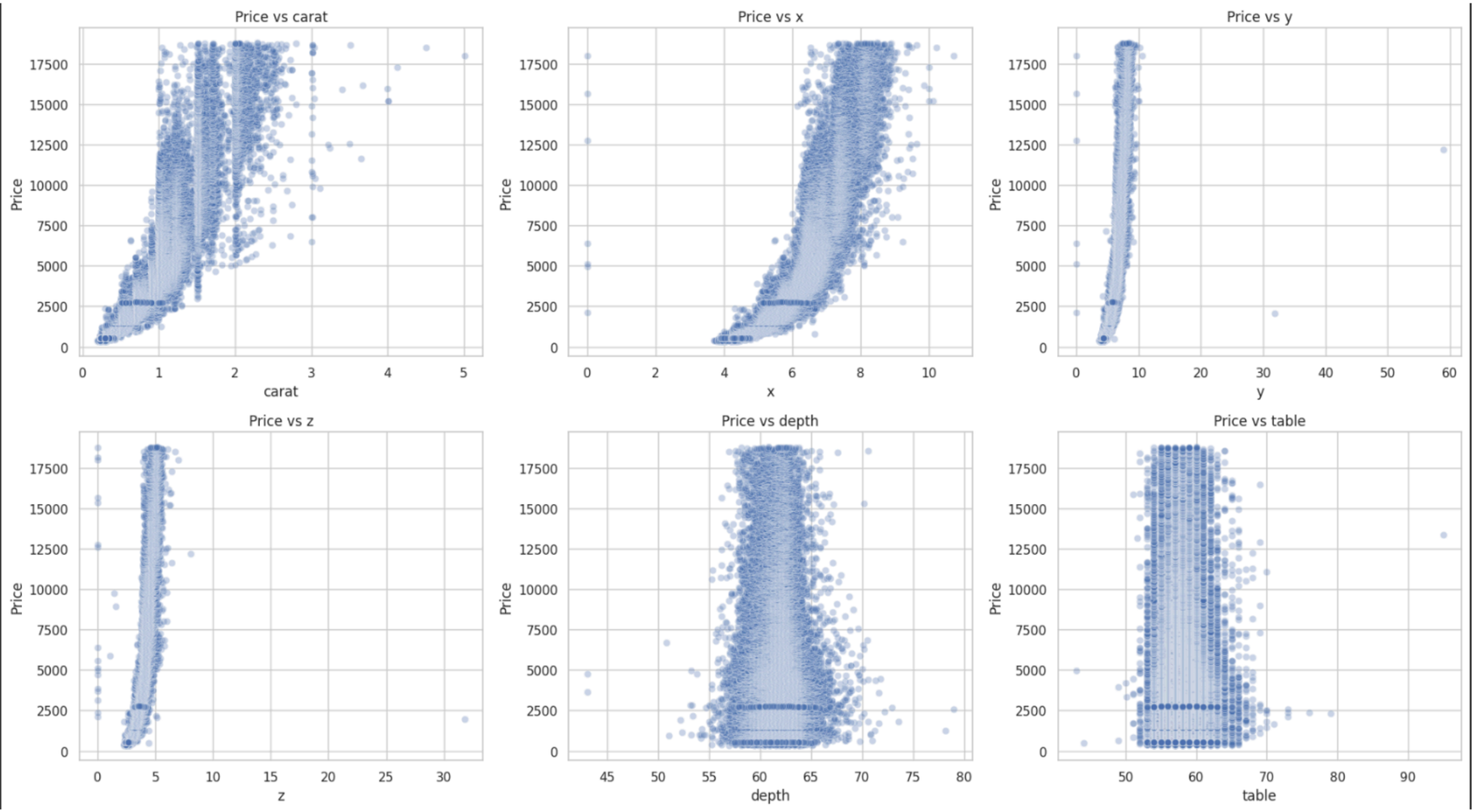


Biểu đồ cho thấy giá kim cương có **phân phối lệch phải** rõ rệt:

- Phần lớn kim cương có giá từ 500 đến 5000 USD, tập trung nhiều nhất trong khoảng **1000–2000 USD**.
- Có một số ít viên kim cương có giá rất cao (trên 10.000 USD), tạo ra đuôi dài về bên phải

III. EDA

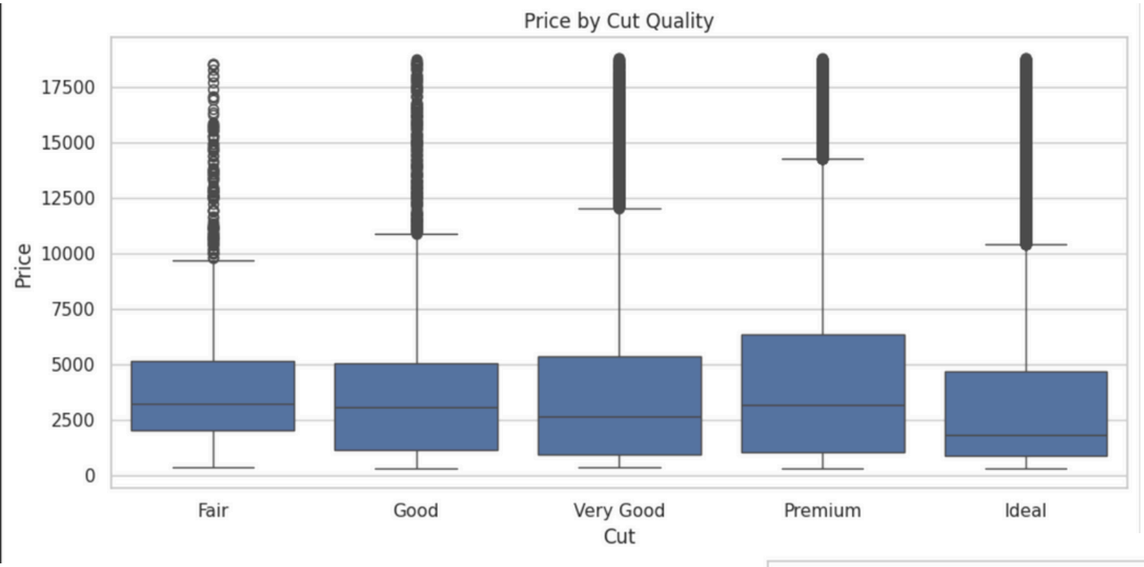
Biểu đồ phân tán của giá với các biến định lượng

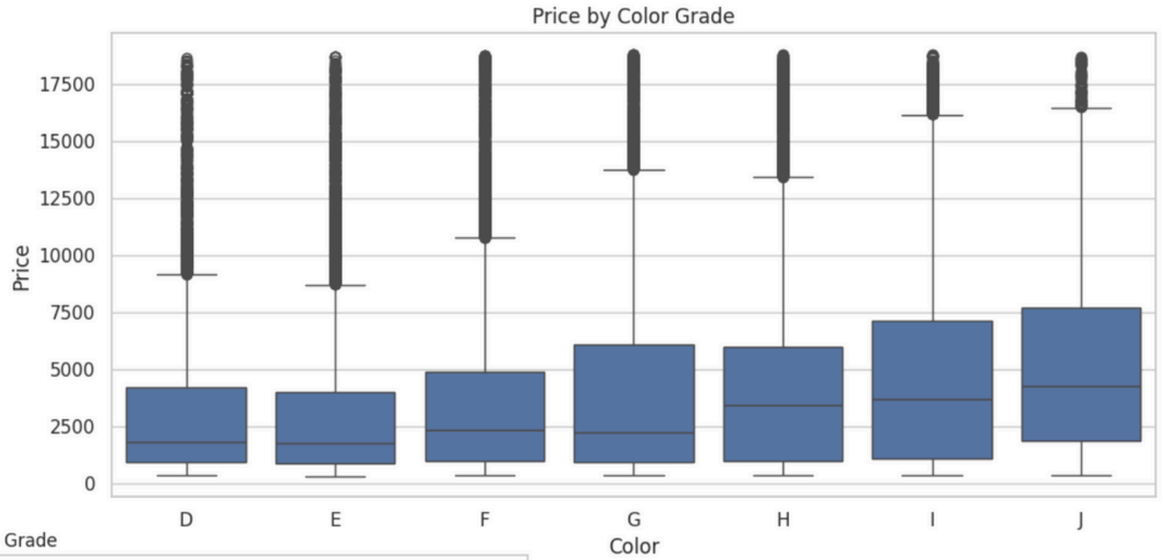


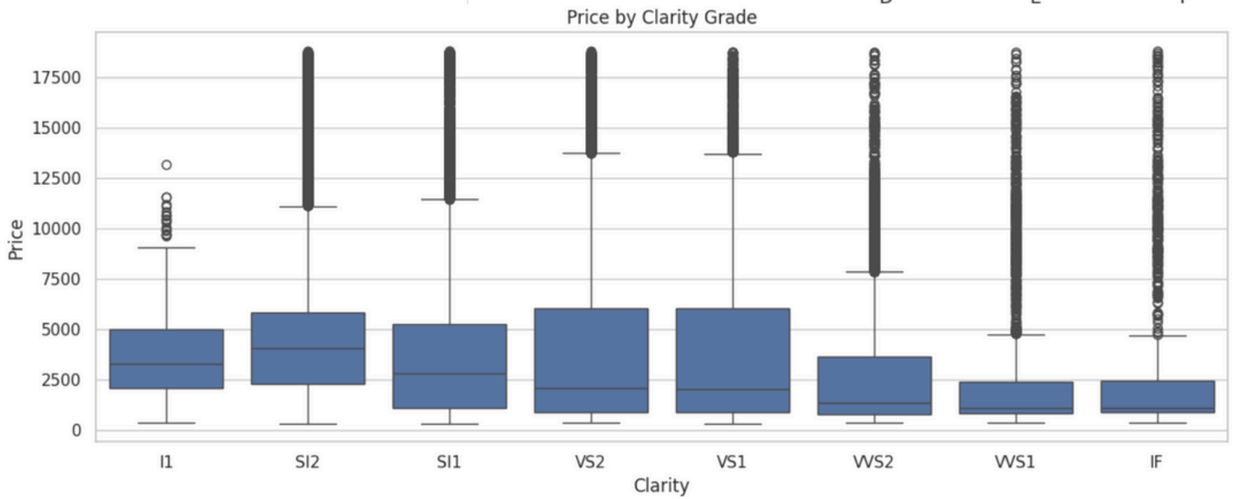
- carat, x, y, z có **tương quan dương mạnh** với price, đặc biệt là carat, cho thấy chúng là những biến quan trọng trong việc định giá.
- Đối với carat có một số outliers với carat > 3. (Loại bỏ 30 dòng)
- depth và table không có mối tương quan tuyến tính rõ ràng với price, cho thấy **tầm ảnh hưởng hạn chế** của chúng trong mô hình dự đoán.

III. EDA

Biểu đồ hộp của giá với các biến phân loại





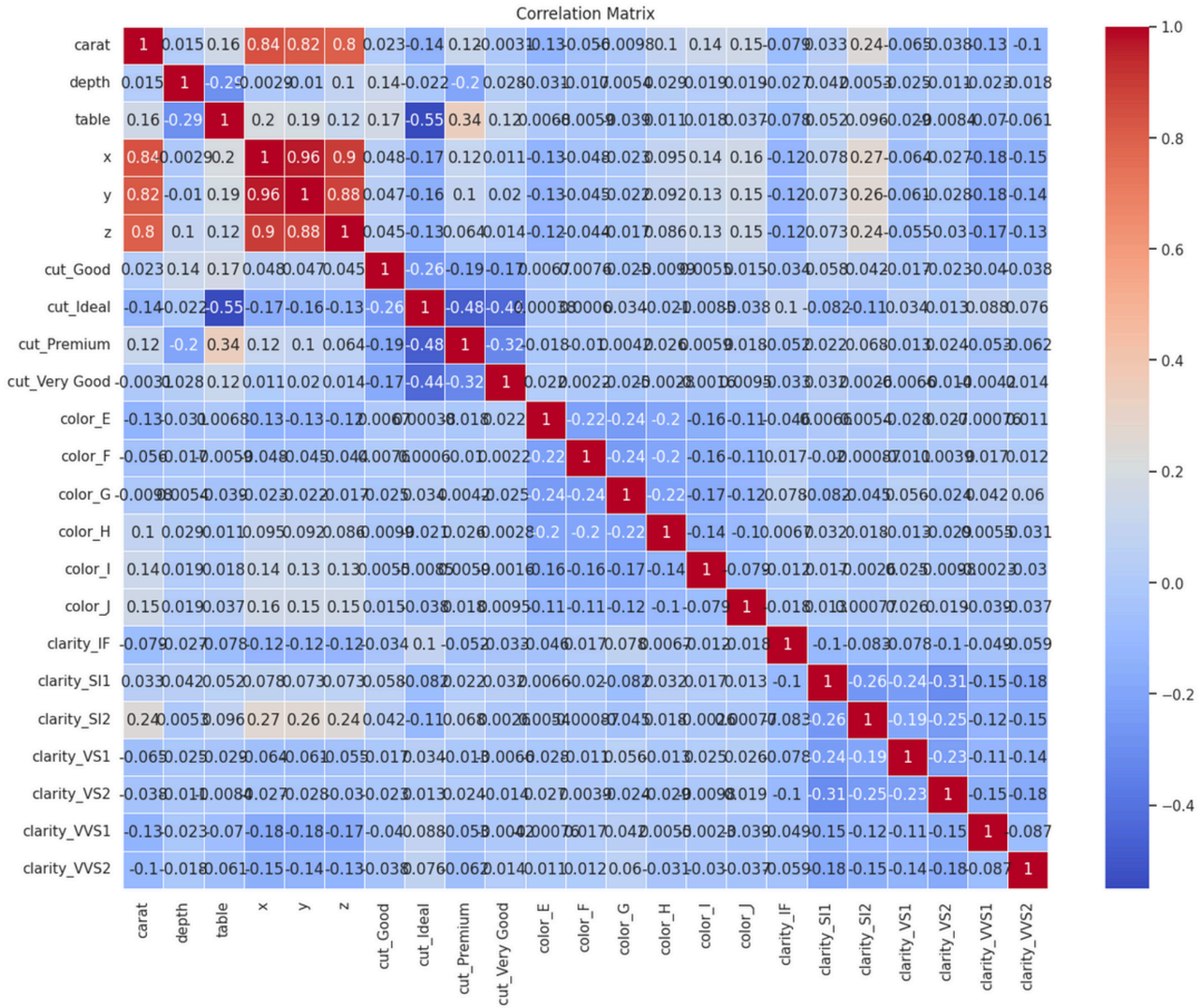


⚠️ Nhận xét về Outliers:

- Các boxplot trên cho thấy outliers xuất hiện rất nhiều ở tất cả nhóm phân loại.
- Tuy nhiên, việc loại bỏ outliers có thể làm mất đáng kể dữ liệu hợp lệ, đặc biệt là với kim cương có giá trị cao.
- Do đó, không loại bỏ outliers, mà giữ nguyên toàn bộ dữ liệu để đảm bảo mô hình học được sự đa dạng về giá và chất lượng trong thị trường thực tế.

III. EDA

Ma trận tương quan



- Ba biến x, y, z đại diện cho chiều dài, chiều rộng và chiều cao của viên kim cương.
- Tuy nhiên, theo ma trận tương quan, cả 3 biến này có:

+ **Tương quan rất cao** với nhau (ví dụ: x và y ~0.96),

+ **Tương quan cao với carat** (trên 0.8),

+Điều này cho thấy có hiện tượng **đa cộng tuyến cao**, làm tăng tính dư thừa thông tin trong mô hình.

- Ngoài ra, trong tập dữ liệu đã có biến depth, được tính dựa trên z chia cho đường kính trung bình → phần nào đã phản ánh tỷ lệ kích thước.

✅ **Kết luận**: Để giảm đa cộng tuyến, đơn giản hóa mô hình và tránh trùng lặp thông tin, quyết định loại bỏ các biến x, y, z khỏi tập dữ liệu huấn luyện.

IV. Train and Evaluate Models

1. Linear Regression # test 1

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.754
Model:                            OLS   Adj. R-squared:                  0.753
Method:                 Least Squares   F-statistic:                     7617.
Date:                Thu, 26 Jun 2025   Prob (F-statistic):               0.00
Time:                        08:25:39   Log-Likelihood:            -4.4908e+05
No. Observations:               49844   AIC:                         8.982e+05
Df Residuals:                   49823   BIC:                         8.984e+05
Df Model:                          20
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -3063.5699    652.207     -4.697      0.000   -4341.904   -1785.236
carat       6448.2588     17.365    371.347      0.000    6414.224    6482.294
depth         -6.9120      7.136     -0.969      0.333     -20.900       7.076
table         30.8705      5.271      5.857      0.000      20.540      41.201
cut_Good     269.1245     61.323      4.389      0.000     148.930     389.319
cut_Ideal    508.7513     60.926      8.350      0.000     389.335     628.167
cut_Premium  397.4155     58.849      6.753      0.000     282.070     512.761
cut_Very Good 483.2596    58.796      8.219      0.000     368.019     598.500
color_E     -161.4282     32.685     -4.939      0.000    -225.492     -97.365
color_F      -84.0065     32.998     -2.546      0.011    -148.683     -19.330
color_G     -271.7605     32.349     -8.401      0.000    -335.165    -208.356
color_H     -574.1797     34.337    -16.722      0.000    -641.482    -506.878
color_I     -664.7758     38.338    -17.340      0.000    -739.918    -589.634
color_J    -1267.8869     47.308    -26.801      0.000   -1360.611   -1175.163
clarity_IF  3532.8237     93.381     37.832      0.000    3349.795    3715.852
clarity_SI1 2862.5768     80.209     35.689      0.000    2705.366    3019.788
clarity_SI2 2192.0997     80.682     27.170      0.000    2033.963    2350.237
clarity_VS1 3490.7221     81.786     42.681      0.000    3330.421    3651.023
clarity_VS2 3244.2751     80.574     40.265      0.000    3086.350    3402.201
clarity_VVS1 3320.1471    86.245     38.497      0.000    3151.106    3489.188
clarity_VVS2 3444.6687    83.999     41.008      0.000    3280.030    3609.308
==============================================================================
Omnibus:                     9705.628   Durbin-Watson:                   0.774
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            22417.927
Skew:                           1.109   Prob(JB):                         0.00
Kurtosis:                       5.424   Cond. No.                     6.20e+03
==============================================================================
```

- Mô hình giải thích được 75,3%
- depth, color_F không phù hợp với mô hình khi p_value < 0.05, nên loại bỏ

IV. Train and Evaluate Models

1. Linear Regression # test 2

OLS Regression Results

Dep. Variable:	price	R-squared:	0.754
Model:	OLS	Adj. R-squared:	0.753
Method:	Least Squares	F-statistic:	8462.
Date:	Thu, 26 Jun 2025	Prob (F-statistic):	0.00
Time:	08:25:45	Log-Likelihood:	-4.4908e+05
No. Observations:	49844	AIC:	8.982e+05
Df Residuals:	49825	BIC:	8.984e+05
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-3682.6387	298.113	-12.353	0.000	-4266.944	-3098.333
carat	6445.0857	17.326	371.983	0.000	6411.126	6479.045
table	32.9846	4.835	6.821	0.000	23.507	42.462
cut_Good	282.5231	60.060	4.704	0.000	164.806	400.240
cut_Ideal	532.9384	56.640	9.409	0.000	421.924	643.952
cut_Premium	418.2271	55.304	7.562	0.000	309.831	526.623
cut_Very Good	501.9236	56.098	8.947	0.000	391.972	611.876
color_E	-112.1548	26.358	-4.255	0.000	-163.816	-60.494
color_G	-221.9450	25.561	-8.683	0.000	-272.045	-171.846
color_H	-525.0658	28.107	-18.681	0.000	-580.156	-469.976
color_I	-615.4075	32.808	-18.758	0.000	-679.711	-551.104
color_J	-1218.4981	42.916	-28.392	0.000	-1302.614	-1134.382
clarity_IF	3533.4586	93.243	37.895	0.000	3350.701	3716.216
clarity_SI1	2869.5845	80.156	35.800	0.000	2712.477	3026.692
clarity_SI2	2199.5165	80.611	27.285	0.000	2041.517	2357.516
clarity_VS1	3494.7463	81.706	42.772	0.000	3334.601	3654.891
clarity_VS2	3249.9704	80.514	40.365	0.000	3092.162	3407.779
clarity_VVS1	3321.6099	86.156	38.553	0.000	3152.743	3490.477
clarity_VVS2	3448.6961	83.924	41.093	0.000	3284.205	3613.188

Omnibus:	9703.522	Durbin-Watson:	0.774
Prob(Omnibus):	0.000	Jarque-Bera (JB):	22434.070
Skew:	1.108	Prob(JB):	0.00
Kurtosis:	5.427	Cond. No.	1.98e+03

- Mô hình giải thích được 75,3%
- Sau khi điều chỉnh, tất cả các biến đã phù hợp với mô hình Linear Regression

IV. Train and Evaluate Models

2. Train Models

Linear Regression

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
r2 = r2_score(y_test, predictions)
mse = np.mean((y_test - predictions) ** 2)
rmse = np.sqrt(mse)
```

Random Forest

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)

r2_rf = r2_score(y_test, y_pred_rf)
rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))
```

XGBoost

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
xgb_model = xgb.XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=6)
xgb_model.fit(X_train, y_train)
y_pred_xgb = xgb_model.predict(X_test)

r2_xgb = r2_score(y_test, y_pred_xgb)
rmse_xgb = np.sqrt(mean_squared_error(y_test, y_pred_xgb))
```

LightGBM

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
lgb_model = lgb.LGBMRegressor(n_estimators=100, learning_rate=0.1, max_depth=-1)
lgb_model.fit(X_train, y_train)
y_pred_lgb = lgb_model.predict(X_test)

r2_lgb = r2_score(y_test, y_pred_lgb)
rmse_lgb = np.sqrt(mean_squared_error(y_test, y_pred_lgb))
```

IV. Train and Evaluate Models

2. Evaluate Models

Model	R2 Score	RMSE
Linear Regression	0.7561	1960.30
Random Forest	0.8025	1794.67
XGBoost	0.8180	1722.69
LightGBM	0.8056	1742.46

📌 Nhận xét tổng quan về hiệu suất mô hình

Linear Regression cho kết quả thấp nhất:

🔹 R^2 thấp nhất: 0.7561

🔹 RMSE cao nhất: 1960.3

👉 Mô hình tuyến tính đơn giản không phù hợp với mối quan hệ phi tuyến và phân phối phức tạp của dữ liệu.

Random Forest, XGBoost và LightGBM hoạt động tốt hơn nhờ khả năng:

✅ Học được các mối quan hệ phi tuyến

✅ Xử lý hiệu quả tương tác giữa các biến đầu vào

XGBoost là mô hình tốt nhất, với:

⭐ R^2 cao nhất: 0.8180 → Giải thích được khoảng 82% phương sai của biến mục tiêu price

⭐ RMSE thấp nhất: 1722.69 → Sai số dự đoán thấp nhất trong tất cả các mô hình

✅ Kết luận

XGBoost là lựa chọn tối ưu cho bài toán dự đoán giá kim cương trên tập dữ liệu hiện tại.

THANK YOU FOR LISTENING!