

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC UEH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ
KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH

UEH
UNIVERSITY

ĐỒ ÁN MÔN HỌC
BIỂU DIỄN TRỰC QUAN DỮ LIỆU

Đề tài:

Phân tích trực quan các yếu tố ảnh hưởng đến giá của kim cương.

Thành viên:	Đinh Trọng Phạm Minh Nguyễn Như Đặng Nhật	Hữu Hiền Hoàng Huy	MSSV: 31211027643 31211021754 31211027640 31211027641
--------------------	--	------------------------------------	---

Giảng Viên: TS. Nguyễn An Tê

Thành phố Hồ Chí Minh, ngày 26 tháng 12 năm 2023.

MỤC LỤC

<i>Lời mở đầu</i>	4
Tóm tắt đề tài.....	4
Tình huống kinh doanh	4
<i>Chương 1: Tổng quan về tài sản</i>	5
1 . Mục tiêu nghiên cứu.....	5
2. Phương pháp nghiên cứu.....	5
3. Tài liệu sử dụng	5
<i>Chương 2: Tổng quan về bộ dữ liệu</i>	6
1. Mô tả bộ dữ liệu được cung cấp	6
2. Các thuộc tính của bộ dữ liệu	6
2.1. Price	6
2.2. Carat	6
2.3. X	7
2.4. Y	8
2.5. Z	9
2.6. Depth	9
2.7. Table	10
2.8. Cut	11
2.9. Clarity	12
2.10. Color	13
2.11. Unnamed: 0	15
Bảng chú thích dưới đây tóm tắt lại các thuộc tính của bộ dữ liệu và ý nghĩa của chúng	15
<i>Chương 3: Tiền xử lý dữ liệu</i>	15
1. Xóa cột ‘Unnamed: 0’	15
2. Kiểm tra và xử lý trùng lặp	16
3. Kiểm tra và xử lý các giá trị x, y, z = 0	17
Khởi tạo 1 dictionary để format các feature dạng numerical hiển thị từ 1-2 số sau dấu phẩy	17
Thực hiện kiểm tra, khởi tạo DataFrame để với các quan sát chứa x, y, z = 0, sử dụng phương thức `style.apply` kèm với điều kiện để tô nền vàng cho các giá trị cần quan sát.....	17
4. Kiểm tra và xử lý dữ liệu bị thiếu (missing values)	18
Thực hiện kiểm tra, quan sát dữ liệu bị thiếu tồn tại ở các cột với heatmap	18
5. Xử lý dữ liệu nhiễu (Outliers)	22

Chương 4: Kiểm định giả thuyết.....	24
1. Kiểm định Chi bình phương:.....	24
2. Kiểm định Anova 1 chiều.....	28
3. Kiểm định Anova 2 chiều.....	31
3.1. Đánh giá ảnh hưởng của yếu tố ‘cut’ và ‘clarity’ đến giá ‘price’ của kim cương	32
3.2. Đánh giá ảnh hưởng của yếu tố ‘color’ và ‘clarity’ đến giá ‘price’ của kim cương	33
3.3. Đánh giá ảnh hưởng của yếu tố ‘color’ và ‘cut’ đến giá ‘price’ của kim cương.	34
Chương 5: Phân tích và trực quan hóa dữ liệu	35
1. Phân tích đơn biến	35
1.1. Carat	35
1.2. Price.....	36
1.3. X	37
1.4. Y	38
1.5. Z	39
1.6. Depth	40
1.7. Table	41
1.8. Cut	42
1.9. Clarity.....	43
1.10 Color	44
2. Phân tích 2 biến.....	44
2.1. Các biến categories ảnh hưởng đến giá của kim cương như thế nào?	44
2.2. Các biến numeric (biến số) ảnh hưởng đến giá của kim cương như thế nào?	48
2.3. Bên cạnh mối tương quan với biến Price, giữa các biến còn có sự tương quan nào khác với nhau không?	51
3. Phân tích 3 biến.....	58
Chương 6: Mô hình ML	62
1. Định nghĩa:	62
2. Ưu nhược điểm của mô hình:.....	62
2.1. Ưu điểm:	62
2.2. Nhược điểm:	63
3. Xây dựng mô hình:.....	63
3.1. Import thư viện:	63
3.2. Tiền xử lý dữ liệu cho mô hình:	63
3.3. Cài đặt thuật toán:.....	65
4. Đánh giá mô hình:.....	67
4.1. Đánh giá mô hình bằng các chỉ số:.....	67

4.2. Biểu đồ thể hiện giá trị dự đoán so với giá trị ban đầu	68
Chương 7: Kết luận	69
Tài liệu tham khảo	70
Bảng phân công.....	70

Lời mở đầu

Tóm tắt đề tài

Kim cương với vẻ đẹp lấp lánh luôn là biểu tượng sang trọng và đẳng cấp. Giá trị của một viên kim cương không chỉ phản ánh ở sự hiếm có của nó mà còn phụ thuộc vào các yếu tố quyết định ảnh hưởng. Với tập dữ liệu chứa hơn 54,000 hàng dữ liệu được ghi nhận về thông tin của từng viên kim cương, mở ra cơ hội khám phá về những yếu tố ảnh hưởng đến giá trị của chúng.

Tình huống kinh doanh

Doanh nghiệp kinh doanh đá quý A có một lượng dữ liệu buôn bán các mặt hàng kim cương. Bộ phận phân tích tình hình kinh doanh muốn đưa ra dự đoán cụ thể cho cấp trên cũng như một công cụ hỗ trợ cho việc trao đổi viên cương của doanh nghiệp (bao gồm thu mua nhập kim cương và ra mức giá hợp lý rao bán với thị trường). Doanh nghiệp đã nhờ bên bên nhóm tôi hỗ trợ cung cấp công cụ trên(dựa vào công nghệ thông tin, cụ thể là sử dụng mô hình máy học để thực hiện dự án trên). Bên doanh nghiệp đã cung cấp dữ liệu với các thông tin về kích thước, kiểu cắt, màu sắc, độ sâu, v.v. của kim cương. Doanh nghiệp A muốn sử dụng dữ liệu này để ước tính giá kim cương khoảng giá của các loại kim cương dựa vào các giá trị của các viên cương đã và đang sở hữu của doanh nghiệp.

Mục tiêu chính của nhóm tôi là về thuật toán, cụ thể là biểu diễn trực quan các yếu tố ảnh hưởng đến giá kim cương và dự đoán giá kim cương.

Chương 1: Tổng quan về tài

1 . Mục tiêu nghiên cứu

- Thăm dò, phân tích dữ liệu.
- Trực quan hóa thông tin từ tập dữ liệu.
- Tiến hành làm sạch dữ liệu (loại bỏ nhiễu, giữ lại thông tin quan trọng).
- Xây dựng thuật toán mô hình máy học dự đoán khoảng giá kim cương.
- So sánh hiệu suất của các mô hình máy học dựa trên biến mục tiêu là ‘Price’ để xác định mô hình phù hợp với dữ liệu.
- Nhận xét, đánh giá mối quan hệ giữ các yếu tố của kim cương, dựa trên kết quả phân tích.

2. Phương pháp nghiên cứu

- Thăm dò, phân tích dữ liệu.
- Trực quan hóa thông tin từ tập dữ liệu.
- Tiến hành làm sạch dữ liệu (loại bỏ nhiễu, outlier, giữ lại thông tin quan trọng).
- Phân tích đơn biến.
- Kiểm định giả thuyết (chi bình phương, ANOVA 1 chiều, ANOVA 2 chiều) phân tích đa biến (hai biến, ba biến).
- Xây dựng mô hình KNN.

3. Tài liệu sử dụng

- Ngôn ngữ lập trình Python.
- Bộ dữ liệu ‘Diamonds’ được lấy từ Kaggle nhưng đã được tinh chỉnh một chút nhằm cho việc nghiên cứu trong bài.

Chương 2: Tổng quan về bộ dữ liệu

1. Mô tả bộ dữ liệu được cung cấp

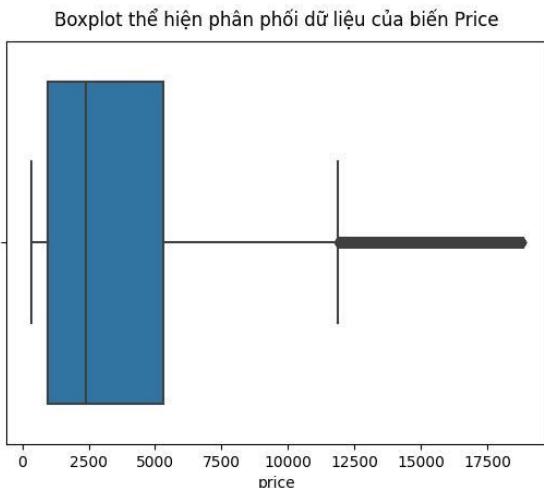
- Bộ dữ liệu ‘Diamonds’ chứa các thông tin đặc điểm của các viên kim cương, với hơn 54,000 mẫu, cung cấp cái nhìn chi tiết về giá và các thuộc tính khác của các viên kim cương. Dữ liệu gồm các thuộc tính như trọng lượng(carat), chất lượng cắt (cut), màu sắc(color), độ trong suốt (clarity), cũng như kích thước chiều dài, rộng và độ sâu của kim cương.

2. Các thuộc tính của bộ dữ liệu

Bộ dữ liệu ghi nhận các thuộc tính sau khi mô tả về 1 viên kim cương.

2.1. Price

Hình 2.1: Thông tin mô tả về biến Price



Các thông số xác suất của biến 'Price':

```
count      53201.000000
mean       3934.016898
std        3989.620914
min        326.000000
25%        951.000000
50%        2403.000000
75%        5325.000000
max       18823.000000
Name: price, dtype: float64
```

Số giá trị null của biến 'Price': 739

Tỉ lệ missing values trên tổng số dữ liệu của biến 'Price': 0.01

- Biến Price ghi nhận thông tin về giá của các viên kim cương (đơn vị: USD).
- Có 99% dữ liệu về giá kim cương được thu thập, trong đó giá các viên kim cương ghi nhận được nằm trong khoảng [326, 18.823] USD.
- Dựa vào Boxplot của biến Price, ta có thể ước lượng những giá trị ngoại lai bắt đầu xuất hiện từ các viên kim cương có giá 11.875 USD trở đi.

2.2. Carat

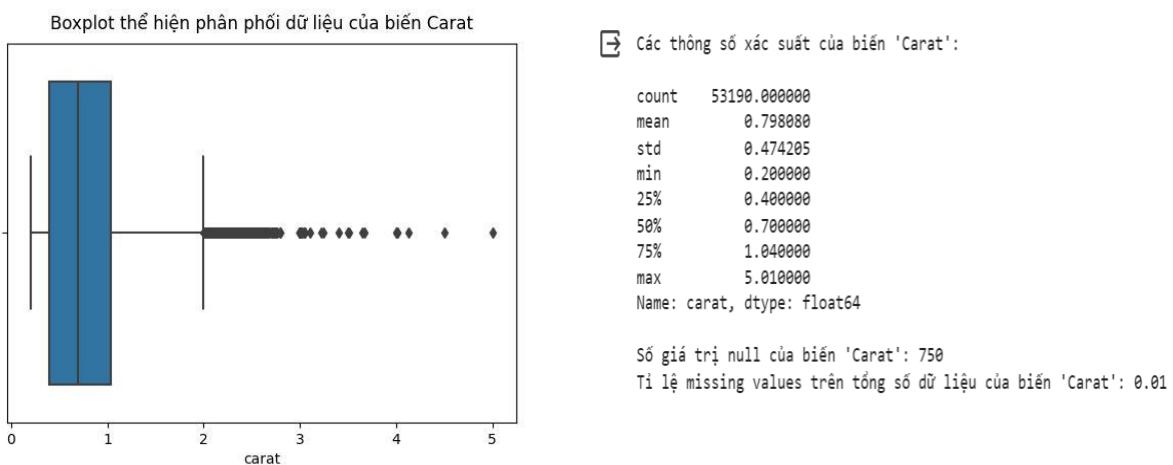
- Biến Carat ghi nhận thông tin về trọng lượng của đá quý (đơn vị: ct). Đơn vị này có thể quy đổi ra gam hoặc milimet để có cái nhìn cụ thể hơn về khối lượng và

kích thước của viên kim cương. Viên kim cương có carat càng lớn tương đương với khối lượng và kích thước của nó cũng lớn, chứng tỏ giá trị của nó càng cao.



Hình 2.2: Bảng quy đổi giữa trọng lượng và kích thước của kim cương

- Quan sát biến Carat ta ghi nhận được những thông tin như Hình dưới đây.



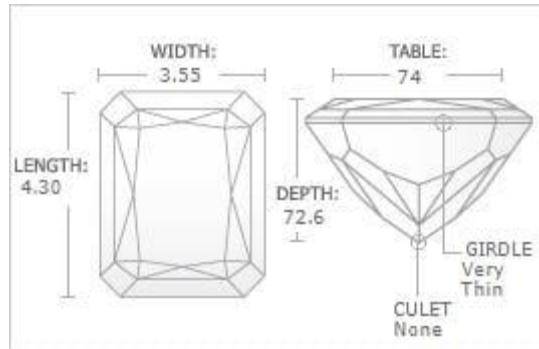
Hình 2.3: Thông tin về thống kê mô tả của biến Carat

- Có 99% dữ liệu về trọng lượng kim cương được thu thập. Các viên kim cương được ghi nhận có trọng lượng trong khoảng [0.2, 5.01] ct và các giá trị outliers bắt đầu xuất hiện từ các viên kim cương có trọng lượng 2 ct.

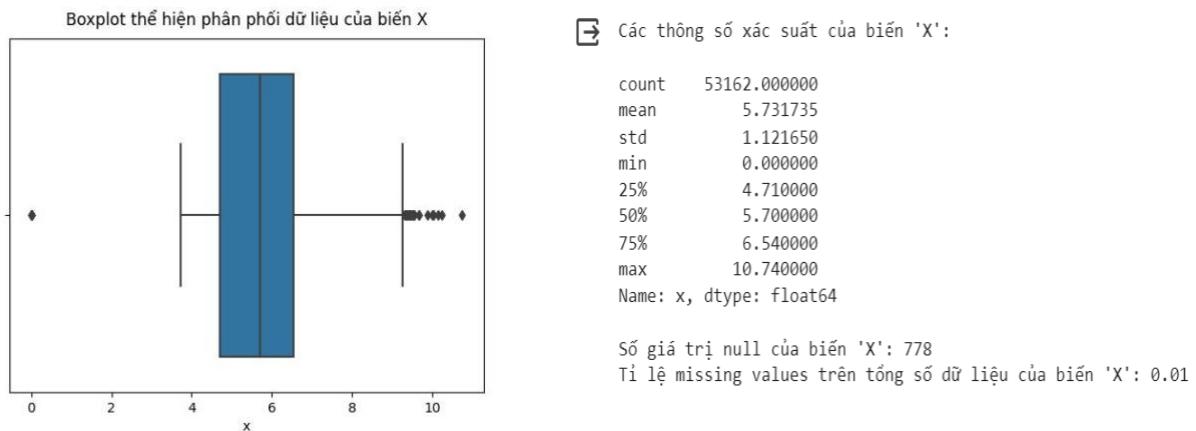
2.3. X

- Biến X ghi nhận thông tin về chiều dài của kim cương, được đo bằng milimet.

- Có 99% dữ liệu về chiều dài của kim cương được thu thập. Các viên kim cương được ghi nhận có chiều dài trong khoảng [0, 10.74]. Điều này khá bất thường vì chiều dài của viên kim cương không thể nhận giá trị 0.
- Outliers của biến X xuất hiện tại điểm 0 và khoảng (2,5.01] mm.



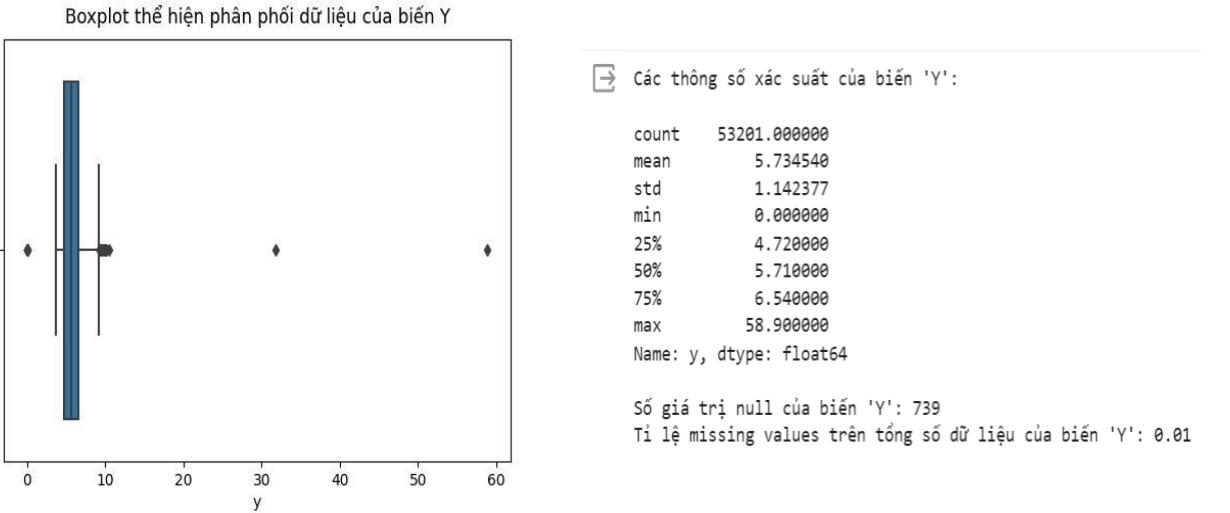
Hình 2.4: Hình minh họa các kích thước của kim cương (Tạm dịch - width: chiều rộng, length: chiều dài, depth: chiều sâu/chiều cao)



Hình 2.5: Thông tin về thống kê mô tả của biến X

2.4. Y

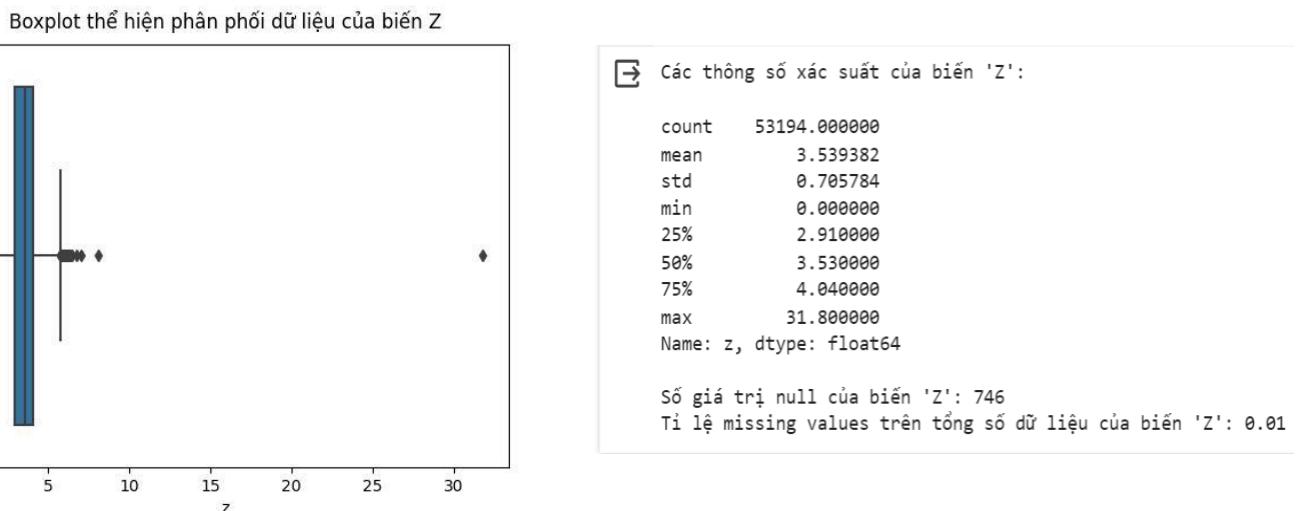
- Biến Y ghi nhận thông tin về chiều rộng của kim cương, tính bằng milimet.
- Có 99% dữ liệu về chiều rộng của kim cương được thu thập. Các viên kim cương được thu thập có chiều rộng nằm trong khoảng [0, 58.9]. Tương tự với biến X, điều này khá bất thường vì chiều rộng của viên kim cương không thể nhận giá trị 0.
- Outliers của biến Y tập trung tại điểm 0 và khoảng (9.27,58.9] mm.



Hình 2.6: Thông tin về thống kê mô tả của biến Y

2.5. Z

- Biến Z ghi nhận thông tin về chiều cao/ chiều sâu của kim cương, tính bằng milimet.
- Có 99% dữ liệu về chiều cao/ chiều sâu của kim cương được thu thập. Các viên kim cương được thu thập có chiều cao/ chiều sâu nằm trong khoảng [0, 31.8]. Cũng tương tự với biến X và biến Y, điều này khá bất thường vì chiều cao/ chiều sâu của kim cương không thể nhận giá trị 0.
- Outliers của biến Z nằm trong khoảng [0, 1.215] \cup (5.735, 31.8] mm.

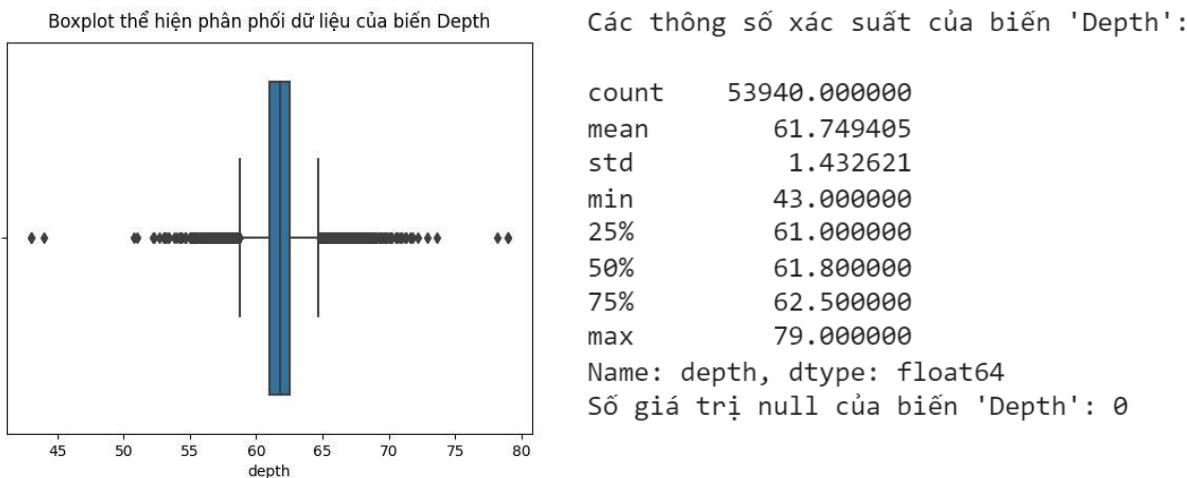


Hình 2.7: Thông tin về thống kê mô tả của biến Z

2.6. Depth

- Biến Depth ghi nhận thông tin về tỷ lệ chiều cao của kim cương so với chiều dài và chiều rộng, được tính bằng công thức:

$$depth = \frac{z}{mean(x, y)} = \frac{2z}{x + y}$$



Hình 2.8: Thông tin về thống kê mô tả của biến Depth

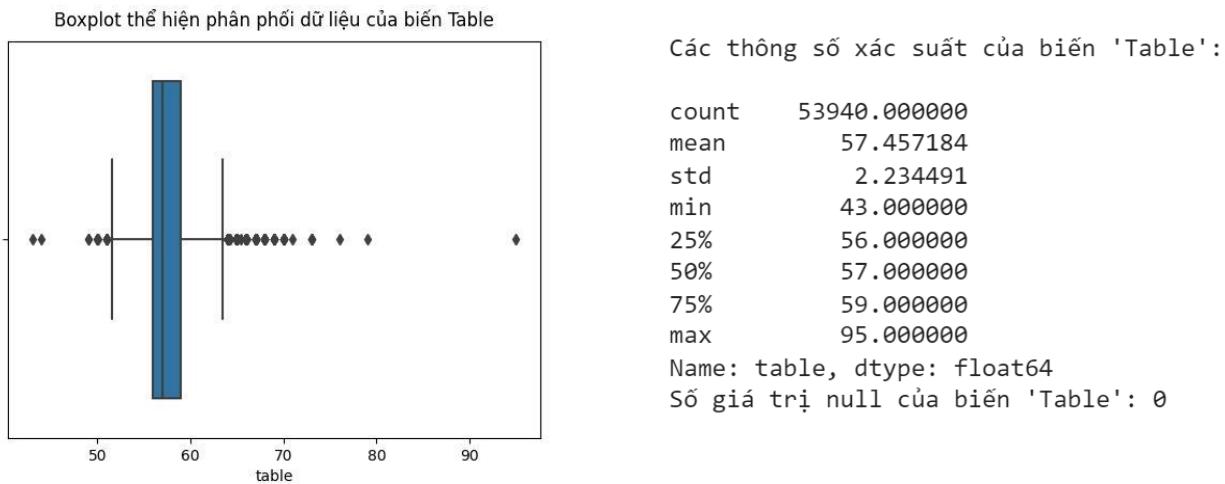
- Dữ liệu về tỷ lệ chiều cao của kim cương so với trung bình chiều dài và chiều rộng được ghi nhận đầy đủ, nằm trong khoảng [61.75, 79%] và outliers thuộc hai khoảng (40, 59.5)% U (63.5, 80)%.

2.7. Table

- Biến Table ghi nhận thông tin về độ rộng của phần đỉnh kim cương so với trung bình chiều dài và chiều rộng. Tỷ lệ Table càng cao thì phần diện tích bề mặt càng lớn, ánh sáng dễ dàng đi vào và phản chiếu ra bên ngoài, viên kim cương càng sáng và lấp lánh.



Hình 2.9: Hình minh họa biến Table trên một viên kim cương thật



Hình 2.9: Thông tin về thống kê mô tả của biến Table

- Dữ liệu về độ rộng của phàn đinh kim cương so với trung bình chiều dài và chiều rộng được ghi nhận đầy đủ, nằm trong khoảng [43,95] với outliers thuộc 2 khoảng [43, 51.5) ∪ (63.5, 95)%

2.8. Cut

- Biết Cut ghi nhận thông tin về phân loại vết cắt của kim cương. Vết cắt của kim cương được đánh giá dựa trên các yếu tố:

Các góc cạnh và tỷ lệ: viên kim cương cần được cắt gọt chính xác để ánh sáng có thể phản chiếu và khúc xạ.

Các mặt đối xứng: các mặt của viên kim cương cần được cắt gọt đối xứng với nhau để tạo nên vẻ đẹp cân đối.

Độ sáng, lửa và độ lắp lánh: cần có độ sáng, lửa (là sự phân tán ánh sáng thành các màu của cầu vồng) và độ lắp lánh cao.

- Vết cắt kim cương tốt sẽ giúp viên kim cương lắp lánh và đẹp mắt hơn.
- Vết cắt kim cương được phân loại theo thang đo có chất lượng vết cắt tăng dần: Fair, Good, Very Good, Premium, Ideal
- Viên kim cương có vết cắt Ideal sẽ hội tụ tất cả các yếu tố để đánh giá một vết cắt tốt ở mức cao nhất.

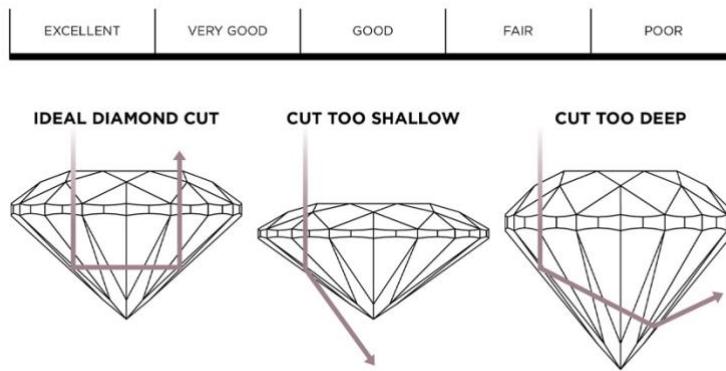
➡ Các thông số xác suất của biến 'Cut':

```

count      53940
unique      5
top        Ideal
freq       21551
Name: cut, dtype: object
Các giá trị unique của biến 'Cut' cụ thể là: ['Ideal' 'Premium' 'Good' 'Very Good' 'Fair']

```

Hình 2.10: Thông tin về thống kê mô tả của biến Cut



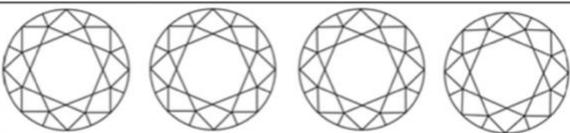
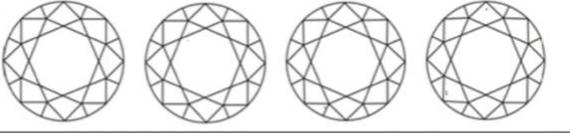
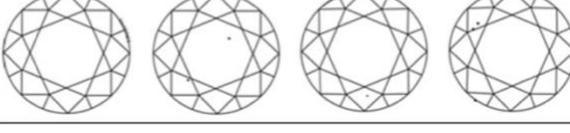
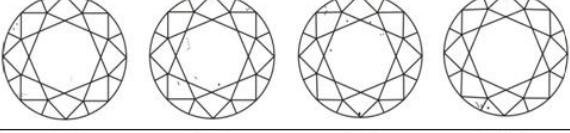
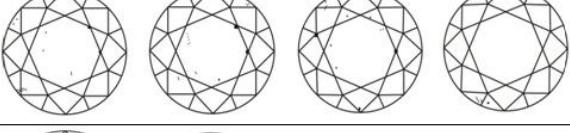
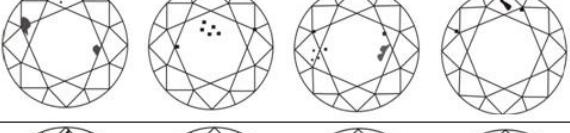
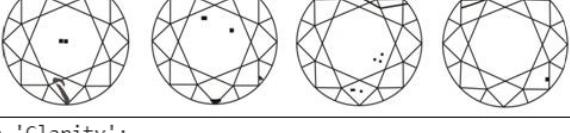
Hình 2.11: Hình minh họa về chất lượng các vết cắt

2.9. Clarity

- Biển Clarity ghi nhận thông tin về độ tinh khiết của kim cương.
- Độ tinh khiết kim cương được đánh giá dựa trên các bao thê (tạp chất) bên trong (Inclusions) và các khuyết điểm bên ngoài (Blemishes) của viên kim cương.
- Các bao thê (tạp chất) là những bao thê (tạp chất) tự nhiên có trong kim cương, gồm các tinh thể nhỏ, vết nứt, vết rõ,... Các khuyết kiềm là khiếm khuyết bên ngoài của viên kim cương gồm các vết trầy xước, các vết nứt,...
- Thang đo của Viện đá quý Hoa Kỳ (GIA) chia độ tinh khiết thành 11 mức độ từ FL (Flawless) – hoàn hảo nhất đến I3 – kém nhất. Đây là thang đo độ tinh khiết được sử dụng rộng rãi nhất hiện nay.
- Các giá trị đánh giá về độ tinh khiết của kim cương được sử dụng trong bài (sắp xếp theo thứ tự độ tinh khiết tăng dần) là:
I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF

Viết tắt	Ý nghĩa	Mô tả
IF	Internally Flawless	Không có bao thê (tạp chất) nào có thấy dưới độ phóng đại x10.
VVS1 VVS2	Very Very Slightly Included	Các bao thê (tạp chất) rất nhẹ nên người chấm điểm có kỹ năng khó có thể thấy dưới độ phóng đại x10.
VS1 VS2	Very Slightly Included	Có các bao thê (tạp chất) nhỏ, có thể nhìn thấy được dưới độ phóng đại x10, nhưng không ảnh hưởng nhiều đến tính thẩm mỹ.
SI1 SI2	Slightly Included	Có các bao thê (tạp chất) có thể nhìn thấy được dưới độ phóng đại x10, có thể ảnh hưởng đến tính thẩm mỹ.
I1 I2 I3	Included	Có các bao thê (tạp chất) rõ ràng, ảnh hưởng đến tính thẩm mỹ và độ bền.

Bảng 2.1: Chú giải về chất lượng vết cắt theo thang đo của GIA

Cấp độ	Hình ảnh minh họa			
FL - IF Hoàn toàn tinh khiết				
VVS1 Khuyết tật rất nhỏ cấp 1				
VVS2 Bao thể rất nhỏ cấp 2				
VS1 Bao thể nhỏ cấp 1				
VS2 Bao thể nhỏ cấp 2				
SI1, SI2 Bao thể nhỏ				
I1 Bao thể hiển thị cấp 1				

Hình 2.12:
từng mức độ

Hình minh họa
tinh khiết

➡ Các thông số xác suất của biến 'Clarity':

```
count      53940
unique       8
top        SI1
freq     13065
Name: clarity, dtype: object
```

Các giá trị unique của biến 'Clarity' cụ thể là: ['SI2' 'SI1' 'VS1' 'VS2' 'VVS2' 'VVS1' 'I1' 'IF']

Hình 2.13: Thông tin về thống kê mô tả của biến Clarity

2.10. Color

- Biến Color ghi nhận thông tin về màu sắc của kim cương.
- Tùy theo sắc độ của kim cương mà kim cương có thể được chia thành các nhóm màu: Colorless (Không màu), Near Colorless (Gần như không màu), Faint Yellow (Phơn phớt vàng), Very Light Yellow (Màu vàng rất nhạt), và Light Yellow (Màu vàng nhạt).
- Viên kim cương có màu rơi vào nhóm Colorless là những viên kim cương trong trẻo, với sắc màu đẹp nhất.

- Các viên kim cương trong tập dữ liệu có các giá trị màu được sắp xếp tiến dần về nhóm Colorless như sau:
J,I,H,G,F,E,D
 - Dựa trên hình 2.14, màu sắc của kim cương trong bài tập trung vào 2 nhóm là Colorless và Near Colorless. Đây cũng là hai nhóm có chất lượng màu tốt nhất trong các nhóm phân loại, thuộc phân khúc kim cương cao cấp.
- Nhóm không màu (Colorless):** có màu trải dài từ D đến F. Kim cương trong nhóm này có màu trắng nhất, không có bao thể (tạp chất) màu.
- Nhóm gần như không màu (Near Colorless):** có màu trải dài từ G đến J. Kim cương trong nhóm này có màu trắng, nhưng có thể nhìn thấy bao thể (tạp chất) màu khi đặt viên đá úp xuống.



Hình 2.14: Phân loại các nhóm màu kim cương

➡ Các thông số xác suất của biến 'Color':

```

count      53940
unique       7
top         G
freq     11292
Name: color, dtype: object
Các giá trị unique của biến 'Color' cụ thể là: ['E' 'I' 'J' 'H' 'F' 'G' 'D']

```

Hình 2.15: Thông tin về thống kê mô tả của biến Color

2.11. Unnamed: 0

- Biến Unnamed:0 ghi nhận thông tin về chỉ mục của từng viên kim cương trong tập dữ liệu.

Bảng chú thích dưới đây tóm tắt lại các thuộc tính của bộ dữ liệu và ý nghĩa của chúng

Tên thuộc tính	Mô tả	Ghi chú
Carat	Trọng lượng của viên kim cương	Đao động từ 0,2 - 5,01 carat
Cut	Chất lượng vết cắt viên kim cương	Fair, Good, Very Good, Premium, Ideal
Color	Màu sắc viên kim cương	Từ J (thấp nhất) đến D (cao nhất)
Clarity (độ tinh khiết)	Đơn vị đo (thước đo) độ tinh khiết của viên cương	I1 (thấp nhất), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (cao nhất).
X	Chiều dài	0 – 10.74
Y	Chiều rộng	0 – 58.9
Z	Chiều cao/chiều sâu	0 – 31.8
Depth	Tỷ lệ chiều cao của kim cương so với chiều dài và chiều rộng	Được tính bằng: $z/ \text{mean}(x,y)$ $= 2*z / (x+y)$ Đao động từ 43 - 79
Table	Độ rộng của phần đỉnh kim cương so với trung bình chiều dài và chiều rộng	Đao động từ 43 - 95
Price	Giá kim cương tính theo đồng đô la Mỹ (USD)	Đao động từ \$326 – \$18,823

Bảng 2.2: Tóm tắt ý nghĩa các thuộc tính của bộ dữ liệu

Chương 3: Tiền xử lý dữ liệu

1. Xóa cột ‘Unnamed: 0’

Khi quan sát thấy giá trị cột ‘Unnamed: 0’ là cột số tự nhiên chạy đều từ 1 đến hết số lượng quan sát của tập dữ liệu, nhưng giá trị 0 của cột lại bị nhảy lên tên cột, ta thực hiện xóa cột vì đã có cột index sẵn có khi đọc dữ liệu từ tập tin csv

```
[ ] data = data.drop(["Unnamed: 0"], axis=1)
data.describe()
```

	carat	depth	table	price	x	y	z
count	53190.000000	53940.000000	53940.000000	53201.000000	53162.000000	53201.000000	53194.000000
mean	0.798080	61.749405	57.457184	3934.016898	5.731735	5.734540	3.539382
std	0.474205	1.432621	2.234491	3989.620914	1.121650	1.142377	0.705784
min	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	61.000000	56.000000	951.000000	4.710000	4.720000	2.910000
50%	0.700000	61.800000	57.000000	2403.000000	5.700000	5.710000	3.530000
75%	1.040000	62.500000	59.000000	5325.000000	6.540000	6.540000	4.040000
max	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Hình 3.1: Hình ảnh thống kê mô tả sau khi xoá cột 'Unnamed:0'

2. Kiểm tra và xử lý trùng lặp

Tiến hành kiểm tra các quan sát trùng lặp của tập dữ liệu với phương thức `duplicate()` của Pandas DataFrame, kèm phương thức `sum()` để tính tổng

```
[ ] data.duplicated().sum()
```

127

Hình 3.2: Hình ảnh số lượng dòng trùng lặp

```
[ ] dup_rows = data[data.duplicated()]
dup_rows
```

	carat	cut	color	clarity	depth	table	price	x	y	z
1005	0.79	Ideal	G	SI1	62.3	57.0	2898.0	5.90	5.85	3.66
1006	0.79	Ideal	G	SI1	62.3	57.0	2898.0	5.90	5.85	3.66
1008	0.79	Ideal	G	SI1	62.3	57.0	2898.0	5.90	5.85	3.66
2025	1.52	Good	E	I1	57.3	58.0	3105.0	7.53	7.42	4.28
2183	1.00	Fair	E	SI2	67.0	53.0	3136.0	6.19	6.13	4.13
...
47069	0.52	Ideal	D	VS2	61.8	55.0	1822.0	5.16	5.19	3.20
47296	0.30	Good	J	VS1	63.4	57.0	394.0	4.23	4.26	2.69
49557	0.71	Good	F	SI2	64.1	60.0	2130.0	0.00	0.00	0.00
50079	0.51	Ideal	F	VVS2	61.2	56.0	2203.0	5.19	5.17	3.17
52861	0.50	Fair	E	VS2	79.0	73.0	2579.0	5.21	5.18	4.09

127 rows × 10 columns

Hình 3.3: Hình ảnh dữ liệu bị trùng lặp

Tiến hành xử lí các hàng giá trị trùng lặp với phương thức `drop_duplicates()`

```
[ ] data = data.drop_duplicates()
```

Hình 3.4: Hình ảnh code xoá dòng trùng lặp

3. Kiểm tra và xử lí các giá trị x, y, z = 0

Khởi tạo 1 dictionary để format các feature dạng numerical hiển thị từ 1-2 số sau dấu phẩy

Thực hiện kiểm tra, khởi tạo DataFrame để với các quan sát chứa x, y, z = 0, sử dụng phương thức `style.apply` kèm với điều kiện để tô nền vàng cho các giá trị cần quan sát

```
[ ] format_dict = {"carat" : "{:.2f}", "depth" : "{:.1f}", "table" : "{:.1f}", "x" : "{:.2f}", "y" : "{:.2f}", "z" : "{:.2f}"}
df_zero = data.loc[(data["x"] == 0) | (data["y"] == 0) | (data["z"] == 0)]
df_zero.style.apply(lambda x: ["background: yellow" if n == 0 else "" for n in x], axis = 1).format(format_dict)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
2207	1.00	Premium	G	SI2	59.1	59.0	3142.000000	6.55	6.48	0.00
2314	1.01	Premium	H	I1	58.1	59.0	3167.000000	6.66	6.60	0.00
4791	1.10	Premium	G	SI2	63.0	59.0	3696.000000	6.50	6.47	0.00
5471	1.01	Premium	F	SI2	59.2	58.0	3837.000000	6.50	6.47	0.00
10167	1.50	Good	G	I1	64.0	61.0	4731.000000	7.15	nan	0.00
11182	1.07	Ideal	F	SI2	61.6	56.0	4954.000000	0.00	6.62	0.00
11963	1.00	Very Good	H	VS2	63.3	53.0	5139.000000	0.00	0.00	0.00
13601	1.15	Ideal	G	VS2	59.2	56.0	5564.000000	6.88	6.83	0.00
15951	1.14	Fair	G	VS1	57.5	67.0	6381.000000	0.00	0.00	0.00
24394	2.18	Premium	H	SI2	59.4	61.0	12631.000000	8.49	nan	0.00
24520	1.56	Ideal	G	VS2	62.2	54.0	12800.000000	0.00	0.00	0.00
26123	2.25	Premium	I	SI1	61.3	58.0	15397.000000	8.52	8.42	0.00
26243	1.20	Premium	D	VVS1	62.1	59.0	15686.000000	0.00	0.00	0.00
27112	2.20	Premium	H	SI1	61.2	59.0	17265.000000	8.42	8.37	0.00
27429	2.25	Premium	H	SI2	62.8	59.0	18034.000000	0.00	0.00	0.00
27503	2.02	Premium	H	VS2	62.7	53.0	18207.000000	8.02	7.95	0.00
27739	2.80	Good	G	SI2	63.8	58.0	18788.000000	8.90	8.85	0.00
49556	0.71	Good	F	SI2	64.1	60.0	2130.000000	0.00	0.00	0.00
51506	1.12	Premium	G	I1	60.4	59.0	2383.000000	6.71	6.67	0.00

Hình 3.5: Giá trị x, y, z = 0

Tiếp đến, thực hiện đếm và loại bỏ các hàng chứa giá trị cần loại bỏ bằng cách tái định nghĩa tập dữ liệu đang xử lí với điều kiện các cột ‘x,y,z’ khác 0.

```
[ ] len(data[(data['x']==0) | (data['y']==0) | (data['z']==0)])
```

19

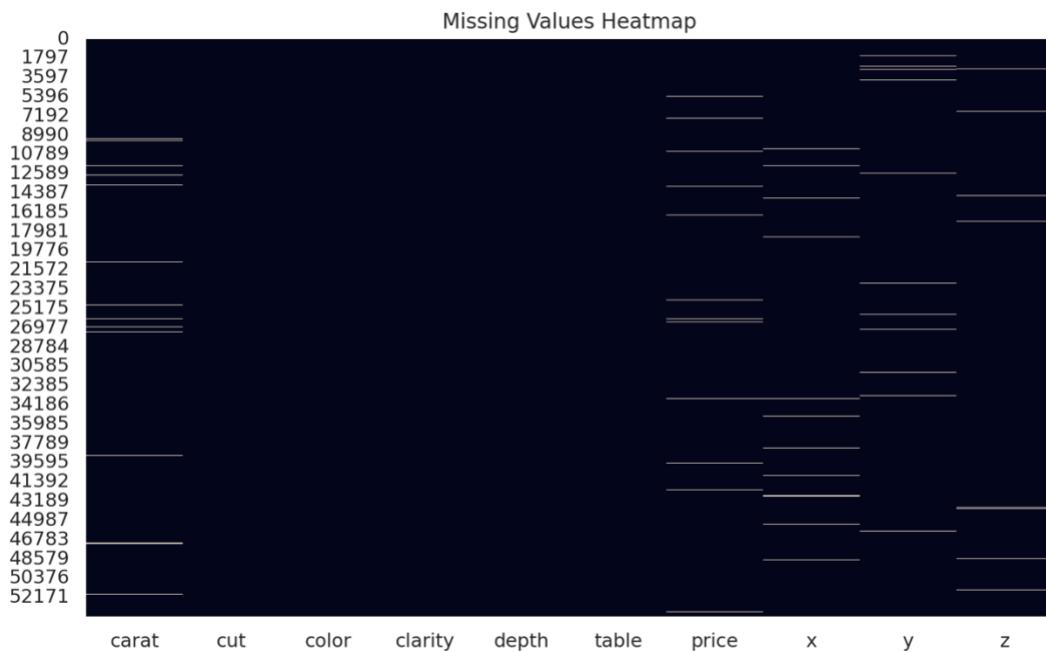
```
[ ] data = data[(data[['x','y','z']] != 0).all(axis=1)]
```

```
[ ] data.loc[(data['x']==0) | (data['y']==0) | (data['z']==0)]
```

4. Kiểm tra và xử lý dữ liệu bị thiếu (missing values)

Thực hiện kiểm tra, quan sát dữ liệu bị thiếu tồn tại ở các cột với heatmap

```
[ ] # Trực quan hóa bằng heatmap thể hiện các ô có chứa missing values của biến
    plt.figure(figsize=(10, 6))
    sns.heatmap(data.isnull(), cbar=False)
    plt.title('Missing Values Heatmap')
    plt.show()
```



Thống kê số lượng và tỉ lệ phần trăm của các giá trị missing value

```
[ ] # Tổng số missing values của từng biến  
missing_values = data.isna().sum()  
missing_values
```

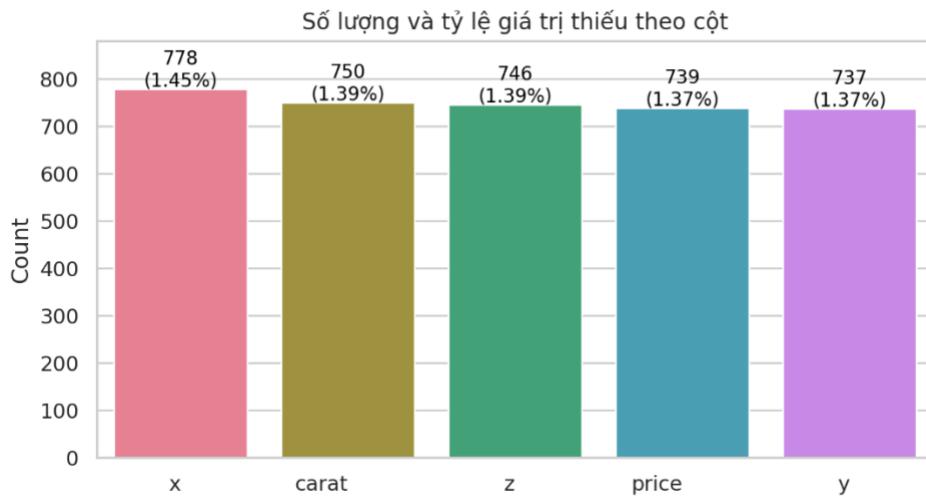
```
carat      750  
cut         0  
color       0  
clarity     0  
depth       0  
table       0  
price      739  
x           778  
y           737  
z           746  
dtype: int64
```

```
[ ] # Tỷ lệ missing values của từng biến  
missing_values = data.isnull().sum()  
missing_ratio = missing_values / len(data)  
print(missing_ratio)
```

```
carat      0.013942  
cut        0.000000  
color      0.000000  
clarity    0.000000  
depth      0.000000  
table      0.000000  
price      0.013738  
x          0.014463  
y          0.013700  
z          0.013868  
dtype: float64
```

Biểu diễn trực quan thống kê số lượng và tỉ lệ phần trăm của các giá trị missing value

```
[ ] # Tính số lượng và tỷ lệ giá trị thiếu  
missing_values_count = data.isnull().sum()  
missing_values_ratio = missing_values_count / len(data)  
  
# Lọc ra các cột có giá trị thiếu  
missing_values = pd.DataFrame({  
    'Count': missing_values_count[missing_values_count > 0],  
    'Ratio': missing_values_ratio[missing_values_count > 0]  
})  
  
# Sắp xếp giảm dần theo số lượng giá trị thiếu  
missing_values.sort_values(by='Count', ascending=False, inplace=True)  
  
# Chọn bảng màu mà bạn muốn sử dụng (ở đây mình sử dụng màu đậm hơn)  
custom_palette = sns.color_palette("husl", n_colors=len(missing_values))  
  
# Vẽ biểu đồ  
plt.figure(figsize=(8, 4))  
sns.set(style="whitegrid")  
bar_plot = sns.barplot(x=missing_values.index, y='Count', data=missing_values, palette=custom_palette)  
bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation=45, horizontalalignment='right')  
  
# Hiển thị số lượng và tỷ lệ trên đỉnh của cột  
for p, value in zip(bar_plot.patches, missing_values['Count']):  
    bar_plot.annotate(f'{value}\n({value / len(data) * 100:.2f}%)', (p.get_x() + p.get_width() / 2., p.get_height()),  
                      ha='center', va='center', xytext=(0, 10), textcoords='offset points', fontsize=10, color='black')  
plt.ylim(0, 880)  
plt.title('Số lượng và tỷ lệ giá trị thiếu theo cột')  
plt.xticks(rotation=0)  
plt.show()
```



Sử dụng phương pháp MICE để điền vào các giá trị missing value

Multiple Imputation (MI): Là một phương pháp thay thế giá trị thiếu bằng nhiều giá trị được dự đoán từ phân phối xác suất của giá trị đó. Mỗi giá trị thiếu được impute nhiều lần để tạo ra nhiều tập dữ liệu hoàn chỉnh khác nhau.

Chained Equations: Ý tưởng chính của MICE là impute từng biến thiếu một cách tuần tự, mỗi biến sẽ được dự đoán dựa trên các biến khác trong mô hình hồi quy

Các bước chạy của thuật toán MICE:

Bước 1: chọn số lần lặp k cho các bước thực hiện

Bước 2: thực hiện thay thế tạm thời bằng một giá trị gần như mean cho các giá trị còn thiếu trong mỗi cột. để bộ dữ liệu không được thiếu giá trị nào

Bước 3: đổi với cột cụ thể mà muốn áp đặt, thực hiện thay đổi giá trị cột được impute thành thiếu

Bước 4: xây dựng mô hình hồi qui, dự đoán cột đó bằng sử dụng các cột còn lại làm yếu tố dự đoán. mô hình xây dựng dựa trên các hàng mà cột định dự đoán không bị thiếu. sử dụng mô hình này để dự đoán các hàng thiếu của cột

Bước 5: lặp lại bước 2 – bước 4

Hoàn thành 1 vòng dự đoán cho tất cả các cột của tập dữ liệu tạo thành 1 lần lặp.

Với mỗi lần lặp, giá trị dữ đoán đoán của dự đoán tạm thời cho mỗi cột sẽ được cải thiện, Vì vậy, có sự liên tục giữa các lần lặp liên tiếp, ý nghĩa từ ‘chained’ trong tên của thuật toán.

Khi thực hiện sau k lần lặp, dự đoán mới nhất cho mỗi biến sẽ được giữ lại làm chỉ cuối cùng.

Các bước sử dụng phương pháp MICE:

Bước 1: sử dụng `pd.get_dummies` để chuẩn bị và mã hóa one-hot các biến phân loại

Bước 2: thực hiện **kiểm tra tên cột** đảm bảo rằng tên cột là các identifier hợp lệ (không chứa ký tự không phù hợp như ở đây là khoảng cách trống), tìm và thay thế bằng dấu gạch dưới với phương thức `replace(' ', '_')`

Bước 3: chuyển dữ liệu vào mô hình MICE `mice.MICEData` , sau đó tiến hành quá trình gán với 10 chuỗi mô hình hồi quy `update_all`.

Bước 4: lưu dữ liệu đã điền giá trị thiếu bằng phương pháp MICE vào DataFrame mới là ‘processed_df’

Kiểm tra thống kê giá trị null sau khi thực hiện xử lý

```
[ ] #Impute dữ liệu missing sử dụng phương pháp MICE
mice_df = pd.get_dummies(data,columns=['color','cut','clarity'])
for col in mice_df.columns:
    if not col.isidentifier():
        print(f"Column name '{col}' is not a valid identifier.")

# Nếu có tên cột không hợp lệ, bạn có thể điều chỉnh tên cột, ví dụ:
mice_df.columns = [col.replace(" ", "_") for col in mice_df.columns]

# Tiếp tục quá trình impute
processed_impute = mice.MICEData(mice_df)
processed_impute.update_all(10)
processed_df = processed_impute.data
# Kiểm tra giá trị null sau khi impute
for i in range(processed_df.shape[0]):
    if processed_df.iloc[i, :].isnull().any():
        print(f"Null values at index {i}:")
        print(processed_df.iloc[i, :])
    else:
        print("Không có giá trị null sau khi xử lý")
    break
```

Column name 'cut_Very Good' is not a valid identifier.
Không có giá trị null sau khi xử lý

Kiểm tra tình trạng DataFrame sau khi xử lý

```
[ ] print(processed_df)

  carat depth table price x y z color_D color_E \
0   0.23   61.5  55.0  326.0  3.05  3.98  2.43     0      1
1   0.21   59.8  61.0  326.0  3.89  3.84  2.31     0      1
2   0.23   56.9  65.0  327.0  4.05  4.07  2.31     0      1
3   0.29   62.4  58.0  334.0  4.20  4.23  2.63     0      0
4   0.31   63.3  58.0  335.0  4.34  4.35  2.75     0      0
...
53789  0.72   68.8  57.0  2757.0  5.75  5.75  3.58     1      0
53790  0.72   63.1  55.0  2757.0  5.69  5.75  3.61     1      0
53791  0.70   62.8  68.0  2757.0  5.66  5.68  3.56     1      0
53792  0.86   61.0  58.0  2757.0  6.15  6.12  3.74     0      0
53793  0.75   62.2  55.0  2757.0  5.83  5.87  3.64     1      0

  color_F ... cut_Premium cut_Very_Good clarity_I1 clarity_IF \
0       0 ...           0           0           0           0
1       0 ...           1           0           0           0
2       0 ...           0           0           0           0
3       0 ...           1           0           0           0
4       0 ...           0           0           0           0
...
53789  0 ...           0           0           0           0
53790  0 ...           0           0           0           0
53791  0 ...           0           1           0           0
53792  0 ...           1           0           0           0
53793  0 ...           0           0           0           0

  clarity_SI1 clarity_SI2 clarity_VS1 clarity_VS2 clarity_VVS1 \
0           0           0           0           0           0
1           1           0           0           1           0
2           0           0           1           0           0
3           0           0           0           1           0
4           0           1           0           0           0
...
53789   1           0           0           0           0
53790   1           0           0           0           0
53791   1           0           0           0           0
53792   0           1           0           0           0
53793   0           1           0           0           0

  clarity_VVS2
0           0
1           0
2           0
3           0
4           0
...
53789   0
53790   0
53791   0
```

Thực hiện khôi phục định dạng DataFrame ban đầu của tập dữ liệu

```
[ ] #Khôi phục lại dữ liệu trước khi áp dụng pd.dummies()
color_columns = ['color_E','color_I','color_J','color_H','color_F','color_G','color_D']
clarity_columns = ['clarity_SI2','clarity_SI1','clarity_VS1','clarity_VS2',
                   'clarity_VVS2','clarity_VVS1','clarity_I1','clarity_IF']
cut_columns = ['cut_Ideal','cut_Premium','cut_Good','cut_Very_Good','cut_Fair']

def update_df(columns_to_remove, df, new_column_to_add):
    index_dict = {col: [] for col in columns_to_remove}
    for col in columns_to_remove:
        for i in range(0, df.shape[0]):
            if df[col].iloc[i] == 1:
                index_dict[col].append(i)
    for k, v in index_dict.items():
        for i in v:
            if k == 'cut_Very_Good':
                df.loc[i, new_column_to_add] = 'Very Good'
            else:
                df.loc[i, new_column_to_add] = k.split('_')[1]
    df.drop(columns=columns_to_remove, inplace=True)

# Update the datafram
update_df(color_columns, processed_df, 'color')
update_df(clarity_columns, processed_df, 'clarity')
update_df(cut_columns, processed_df, 'cut')

print('Dataframe after recovery:\n', processed_df)
```

Dataframe after recovery:

	carat	depth	table	price	x	y	z	color	clarity	cut
0	0.23	61.5	55.0	326.0	3.95	3.98	2.43	E	SI2	Ideal
1	0.21	59.8	61.0	326.0	3.84	3.84	2.31	E	SI1	Premium
2	0.23	56.9	65.0	326.0	4.05	4.05	2.31	E	VS1	Good
3	0.29	62.4	58.0	334.0	4.20	4.23	2.63	I	VVS2	Premium
4	0.31	63.3	58.0	335.0	4.34	4.35	2.75	J	SI2	Good
...
53789	0.72	68.8	57.0	2757.0	5.75	5.76	3.50	D	SI1	Ideal
53790	0.72	63.1	55.0	2757.0	5.69	5.75	3.61	D	SI1	Good
53791	0.70	62.8	60.0	2757.0	5.66	5.68	3.56	D	SI1	Very Good
53792	0.86	61.0	58.0	2757.0	6.15	6.12	3.74	H	SI2	Premium
53793	0.75	62.2	55.0	2757.0	5.83	5.87	3.64	D	SI2	Ideal

[53794 rows x 10 columns]

5. Xử lý dữ liệu nhiễu (Outliers)

Sử dụng biểu đồ Box plot để trực quan diễn tả 5 vị trí phân bố của dữ liệu, đó là:

Với cấu trúc:

- Hộp (Box): Biểu thị phạm vi giữa tứ phân vị thứ nhất (Q1) và tứ phân vị thứ 3 (Q3) của dữ liệu. Đường chéo bên trong box thường là trung vị.
- Chân (Whiskers): Biểu thị phạm vi của dữ liệu, ngoại trừ outliers và làm mềm (1.5 lần phạm vi của hộp) hoặc cứng (min/max) tùy thuộc vào triển khai.
- Outliers: Các giá trị nằm ngoài chân của boxplot, được hiển thị là các điểm riêng lẻ.

Boxplot bên dưới giúp hình dung các giá trị ngoại lệ (outliers) mà các biến numerical. Như biểu đồ hiển thị, tất cả các biến này đều có outlier. Khi lập mô hình dữ liệu, các outlier phải được xử lý bằng phương pháp IQR để xác định outliers bằng cách thiết lập các giá trị biên Upper/Lower để tránh khớp dữ liệu quá mức nhằm xây dựng một mô hình chính xác. Sử dụng `MinMaxScaler` chuyển giá trị của các feature theo khoảng [0, 1], giúp quan sát trực quan các features trên cùng 1 scale dễ dàng hơn.

Công thức tính tập dữ liệu mới của MinMaxScaler:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

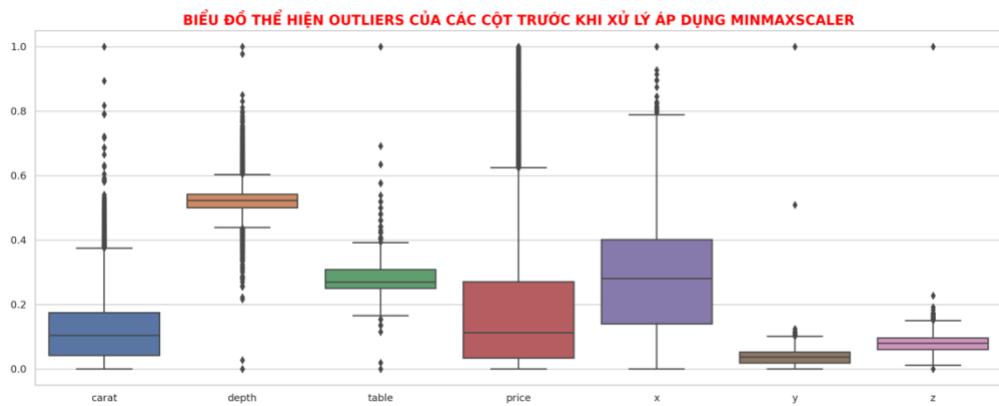
Nhược điểm khi sử dụng khi sử dụng `MinMaxScaler` là giá trị của các feature đã được chia tỉ lệ theo khoảng từ [0, 1] nên không biết được khoảng giá trị cụ thể chính xác của các feature.

Sử dụng `sns.boxplot` biểu diễn trực quan phân phối của dữ liệu, xác định outliers cho mỗi feature trên cùng 1 hình

```
[ ] # Biểu diễn trực quan outliers của tất cả các features
scaler = MinMaxScaler()
numerical_cols = processed_df.columns[processed_df.dtypes != 'object']
numerical_data = processed_df[numerical_cols]

sa = scaler.fit_transform(numerical_data)

sd = pd.DataFrame(sa,columns=numerical_data.columns)
plt.subplots(figsize = (19,7))
sns.boxplot(data = sd)
plt.title('BIỂU ĐỒ THỂ HIỆN OUTLIERS CỦA CÁC CỘT TRƯỚC KHI XỬ LÝ ÁP DỤNG MINMAXSCALER', fontsize=15, color = 'red', fontweight='bold')
plt.show()
```



Loại bỏ các giá trị outlier bằng cách sử dụng phương pháp IQR và loại bỏ các outlier dựa trên giá trị biên.

```

[ ] # Xử lý outliers
## Nhận diện outliers
def detect_outliers(df,features):
    outlier_indices = []

    for c in features:
        # 1st quartile
        Q1 = np.percentile(df[c],25)
        # 3rd quartile
        Q3 = np.percentile(df[c],75)
        # IQR
        IQR = Q3 - Q1
        # Outlier step
        outlier_step = IQR * 1.5
        # detect outlier and their indices
        outlier_list_col = df[(df[c] < Q1 - outlier_step) | (df[c] > Q3 + outlier_step)].index
        # store indices
        outlier_indices.extend(outlier_list_col)

    outlier_indices = Counter(outlier_indices)
    multiple_outliers = [list(i for i, v in outlier_indices.items() if v > 2)]

    return multiple_outliers

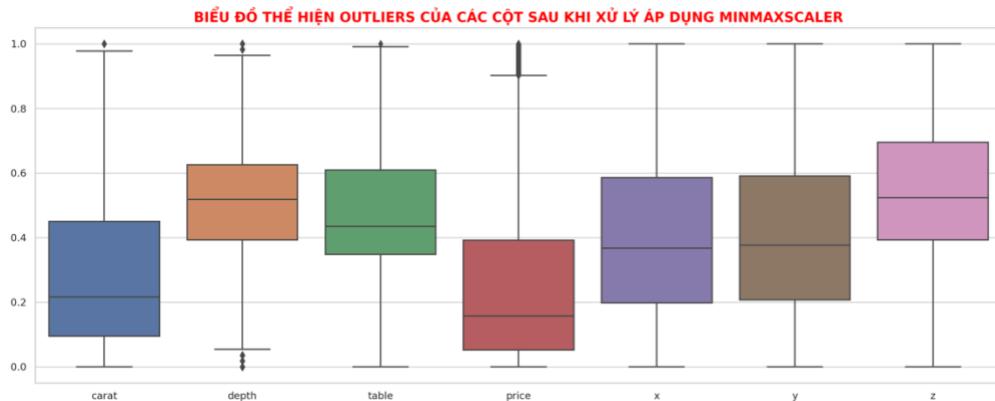
## Xử lý outliers
numerical_cols = list(processed_df.dtypes[processed_df.dtypes != 'object'].index)
for feature in numerical_cols:
    q1 = processed_df[feature].quantile(0.25)
    q3 = processed_df[feature].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
    processed_df = processed_df[(processed_df[feature] >= lower_bound) & (processed_df[feature] <= upper_bound)]


[ ] processed_df.loc[detect_outliers(processed_df,['carat', 'depth', 'table', 'x', 'y', 'z', 'price'])]

```

carat depth table price x y z color clarity cut

- Box plot của các cột numerical sau khi loại bỏ outliers bằng phương pháp IQR.



Chương 4: Kiểm định giả thuyết

1. Kiểm định Chi bình phương:

- Kiểm định Chi bình phương là phương pháp thống kê được dùng để xác định mối quan hệ giữa các biến độc lập rời rạc. Phương pháp này thường được dùng trong các trường hợp:

So sánh mức độ khác biệt giữa các tần số quan sát và tần số dự kiến trong cùng một bảng tần số và thông qua đó, đánh giá mức độ khác biệt có ý nghĩa gì so với thống kê hay không

Kiểm tra các giả thuyết về các biến rời rạc để xem mức độ liên quan, sự đồng đều và mối qua hệ

- Các bước thực hiện kiểm định:

Bước 1: Xây dựng giả thuyết

H_0 : Các biến không có mối quan hệ tương quan, không có sự khác biệt

H_1/H_a : Có mối quan hệ tương quan hoặc sự khác biệt giữa các biến

$$X^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Bước 2: Tính toán giá trị kiểm định

Trong đó:

O_i là giá trị quan sát được

E_i là giá trị mong đợi

c là bậc tự do

$$X_{\alpha}^2 (\text{df})$$

Bước 3: Xác định miền bác bỏ và không thể bác bỏ. Sử dụng hoặc p_{value}

Trong đó: alpha là mức ý nghĩa thống kê

df là bậc tự do

p_{value} là xác suất tìm thấy một kết quả kiểm định khi giả thuyết

H_0 đúng

$X^2_\alpha (\text{df})$

Nếu $p_value < \alpha$ hoặc $X^2 >$ ta bác bỏ giả thuyết H_0 . Ngược lại nếu

$X^2_\alpha (\text{df})$

$p_value > \alpha$ hoặc $X^2 >$ thì ta chấp nhận giả thuyết H_0 .

Bước 4: Ra quyết định (bác bỏ hay chấp nhận giả thuyết H_0)

Bước 5: Kết luận

- Nắm vững các thông tin cũng như mục đích sử dụng của kiểm định Chi bình phương, ta có thể áp dụng kiểm định này để kiểm định mối tương quan giữa các cặp biến phân loại trong bài: {cut, color}, {cut, clarity}, {clarity, color}.
- Thư viện scipy.stats có chứa hàm chi2_contingency sẽ giúp chúng ta thực hiện điều này.
- Hàm chi2_contingency nhận đầu vào là một bảng dữ liệu tần số hoặc bảng chéo và thực hiện kiểm định Chi bình phương để kiểm tra mức độ liên quan hoặc độc lập giữa các biến trong bảng đó. Kết quả của hàm này bao gồm giá trị kiểm định Chi bình phương, giá trị p, bảng tần số mong đợi, và các thông tin khác liên quan đến quá trình kiểm định.
- Phía dưới là đoạn code thực hiện kiểm định Chi bình phương để kiểm định mức độ tương quan giữa các cặp biến phân loại trong tập dữ liệu kim cương. Mức ý nghĩa được xác định là 0.05 (mức ý nghĩa phổ biến, thường được sử dụng trong thống kê). Trong số các tham số của kết quả trả về, nhóm chủ yếu sử dụng giá trị p_value để xác định kết quả kiểm định.

Kiểm định mối tương quan giữa biến Cut và Color

```
# Kiểm định chi bình phương mối tương quan giữa cut và color
_, p_, _ = chi2_contingency(cut_color)
print('Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan
giữa biến \'Cut\' và \'Color\'..')
print('Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến
\'Cut\' và \'Color\'..')
if p < 0.05:
    print('Bác bỏ H0')
    print('Hai biến \'Cut\' và \'Color\' có mối quan hệ với nhau')
else:
    print('Chấp nhận H0')
```

```
print('Không đủ bằng chứng để kết luận hai biến \'Cut\' và \'Color\' có  
mối quan hệ với nhau')
```

Kết quả

Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến 'Cut' và 'Color'.

Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến 'Cut' và 'Color'.

Bắc bỏ H0

Hai biến 'Cut' và 'Color' có mối quan hệ với nhau

Kiểm định mối tương quan giữa biến Cut và Clarity

```
# Kiểm định chi bình phương mối tương quan giữa cut và clarity  
_, p ,_/_ = chi2_contingency(cut_clarity)  
print('Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan  
giữa biến \'Cut\' và \'Clarity\'.')  
print('Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến  
\'Cut\' và \'Clarity\'.')  
if p < 0.05:  
    print('Bắc bỏ H0')  
    print('Hai biến \'Cut\' và \'Clarity\' có mối quan hệ với nhau')  
else:  
    print('Chấp nhận H0')  
    print('Không đủ bằng chứng để kết luận hai biến \'Cut\' và \'Clarity\'  
có mối quan hệ với nhau')
```

Kết quả

➡ Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến 'Cut' và 'Clarity'.

Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến 'Cut' và 'Clarity'.

Bắc bỏ H0

Hai biến 'Cut' và 'Clarity' có mối quan hệ với nhau

Kiểm định mối tương quan giữa biến Color và Clarity

```
# Kiểm định chi bình phương mối tương quan giữa cut và clarity  
_, p ,_/_ = chi2_contingency(cut_clarity)  
print('Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan  
giữa biến \'Cut\' và \'Clarity\'.')  
print('Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến  
\'Cut\' và \'Clarity\'.')  
if p < 0.05:  
    print('Bắc bỏ H0')  
    print('Hai biến \'Cut\' và \'Clarity\' có mối quan hệ với nhau')  
else:  
    print('Chấp nhận H0')  
    print('Không đủ bằng chứng để kết luận hai biến \'Cut\' và \'Clarity\'  
có mối quan hệ với nhau')
```

Kết quả

⇒ Giả thuyết H_0 : Không có mối quan hệ hoặc không có sự tương quan giữa biến 'Color' và 'Clarity'.
Giả thuyết H_1 : Có mối quan hệ hoặc có sự tương quan giữa biến 'Color' và 'Clarity'.
Bắc bỏ H_0
Hai biến 'Color' và 'Clarity' có mối quan hệ với nhau

2. Kiểm định Anova 1 chiều

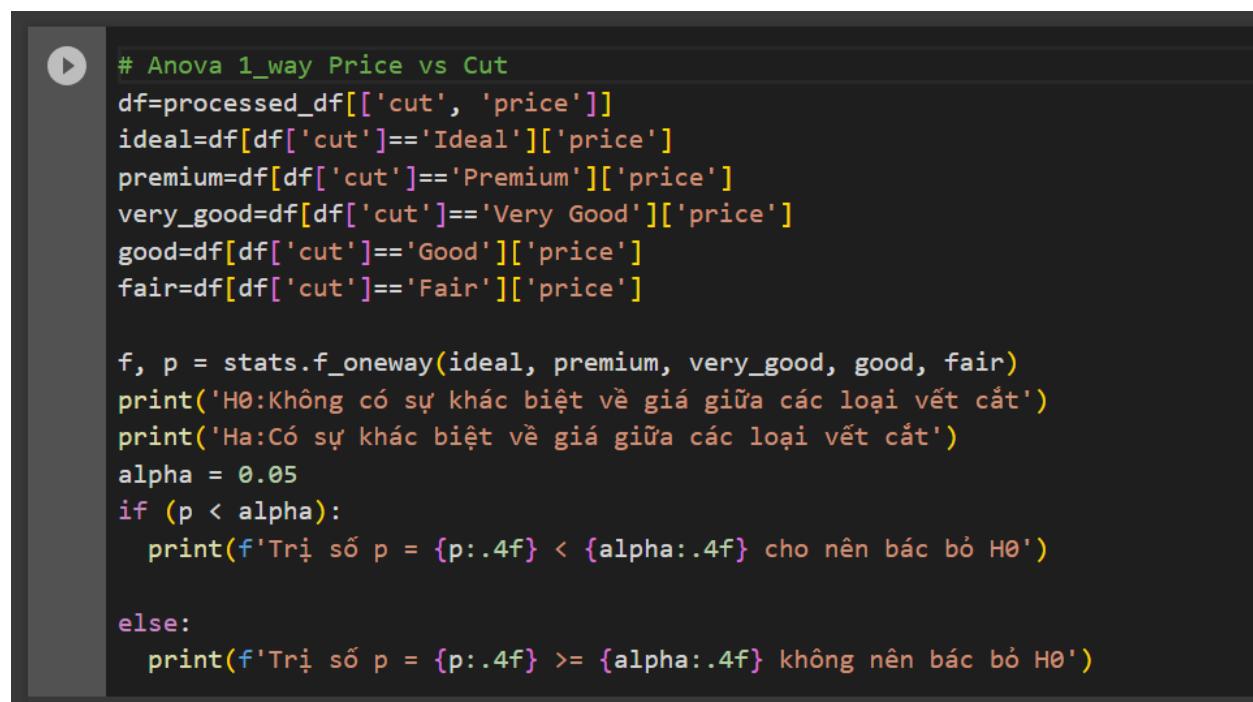
2.1. Kiểm định tác động của các thuộc tính vết cắt 'cut' đến giá 'Price'

- Kiểm định ANOVA 1 chiều về mối tương quan của một biến numerical và một biến categorical:

ANOVA là viết tắt của analysis of Variance hay còn gọi là phân tích phương sai. Phân tích phương sai 1 chiều (One-way ANOVA) là một phép phân tích phương sai được sử dụng để xem xét sự khác nhau về giá trị trung bình của một biến phụ thuộc theo ba hay nhiều nhóm của một biến độc lập. Trong ANOVA một chiều có hai giả thuyết có thể xảy ra: H_0 - không có sự khác biệt giữa các nhóm; H_a : có sự khác biệt giữa các nhóm. Nếu P-value nhỏ hơn một mức ý nghĩa alpha nào đó thì ta có đủ bằng chứng để nói rằng ít nhất một trong các giá trị trung bình của các nhóm khác với các giá trị khác. (trong báo cáo này, ANOVA 1 chiều được thực hiện để kiểm định liệu rằng với các loại vết cắt (cut) khác nhau thì giá của các viên kim cương (Price) có thay đổi hay không).

Để thực hiện kiểm định ANOVA 1 chiều trước tiên ta cần thêm vào thư viện cần thiết:

```
import scipy.stats as stats
```



```
# Anova 1_way Price vs Cut
df=processed_df[['cut', 'price']]
ideal=df[df['cut']=='Ideal']['price']
premium=df[df['cut']=='Premium']['price']
very_good=df[df['cut']=='Very Good']['price']
good=df[df['cut']=='Good']['price']
fair=df[df['cut']=='Fair']['price']

f, p = stats.f_oneway(ideal, premium, very_good, good, fair)
print('H0: Không có sự khác biệt về giá giữa các loại vết cắt')
print('Ha: Có sự khác biệt về giá giữa các loại vết cắt')
alpha = 0.05
if (p < alpha):
    print(f'Tri số p = {p:.4f} < {alpha:.4f} cho nên bác bỏ H0')
else:
    print(f'Tri số p = {p:.4f} >= {alpha:.4f} không nên bác bỏ H0')
```

- Kết quả:

```
H0: Không có sự khác biệt về giá giữa các loại vết cắt  
Ha: Có sự khác biệt về giá giữa các loại vết cắt  
Trị số p = 0.0000 < 0.0500 cho nên bác bỏ H0
```

2.2. Hậu kiểm Tukey HSD:

- Tại sao phải hậu kiểm Tukey HSD:

Bởi vì kiểm định ANOVA 1 chiều chỉ cho biết kết quả cuối cùng là có sự khác biệt hay không khác biệt giữa giá trị trung bình của các nhóm chứ nó không cho thấy so sánh giữa các nhóm cụ thể mà ta kiểm định.

Hậu kiểm Tukey HSD là gì?

Là một trong những phương pháp kiểm tra hậu kiểm. Hậu kiểm Tukey cho phép chúng ta so sánh theo cặp giá trị trung bình của các nhóm. Trong đó:

+ Tham số endog là biến phụ thuộc (trong trường hợp này, giá của kim cương).

+ Tham số groups là biến độc lập (loại vết cắt của kim cương).

+ Tham số alpha=0.05 xác định mức ý nghĩa cho phân tích Tukey HSD, thường được đặt là 0.05.

- Hậu kiểm Tukey HSD cho kiểm định tác động của các thuộc tính vết cắt ‘cut’ đến giá ‘Price’:

```
[ ] #Tukey test( Hậu kiểm Tukey)  
from statsmodels.stats.multicomp import pairwise_tukeyhsd  
m_comp = pairwise_tukeyhsd(endog=df['price'],  
                           groups=df['cut'],  
                           alpha=0.05)  
  
print(m_comp)
```

- Kết quả:

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05  
=====  
group1    group2    meandiff  p-adj      lower      upper   reject  
-----  
Fair       Good     -505.3111  0.0174   -952.0109   -58.6114   True  
Fair       Ideal    -945.2356   0.0   -1379.3891   -511.082   True  
Fair       Premium  -309.6833  0.2978   -745.8603   126.4937  False  
Fair Very Good -519.9967  0.0102   -956.6184   -83.375   True  
Good      Ideal    -439.9245   0.0   -566.4421   -313.4068  True  
Good      Premium  195.6278  0.0006    62.3321   328.9236  True  
Good Very Good -14.6856  0.9983   -149.4296   120.0584  False  
Ideal      Premium  635.5523   0.0   553.5986   717.506   True  
Ideal Very Good 425.2389   0.0   340.9502   509.5275  True  
Premium  Very Good -210.3134   0.0   -304.4709   -116.156   True  
-----
```

- Nhận xét:

Có thể thấy, trong cột p-adjust chỉ có 2 cặp giá trị là (Fair và Premium) với (Good và Very Good) là có giá trị lớn hơn alpha = 0.05 và trả về giá trị False ở cột reject. Còn lại các cặp khác đều trả về giá trị là True. Cho nên ta có thể tin rằng việc bác bỏ giả thuyết Ho là đúng đắn.

2.3. Kiểm định chi bình phương về mối tương quan giữa hai biến categorical:

```
[ ] cut = processed_df['cut']
color = processed_df['color']
clarity = processed_df['clarity']

# cut_color crosstab
cut_color = pd.crosstab(cut,color)
cut_clarity = pd.crosstab(cut,clarity)
color_clarity = pd.crosstab(color,clarity)

# Kiểm định chi bình phương mối tương quan giữa cut và color
_, p ,_,_ = chi2_contingency(cut_color)
print('Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến \'Cut\' và \'Color\' .')
print('Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến \'Cut\' và \'Color\' .')
if p < 0.05:
    print('Bắc bỏ H0')
    print('Hai biến \'Cut\' và \'Color\' có mối quan hệ với nhau')
else:
    print('Chấp nhận H0')
    print('Không đủ bằng chứng để kết luận hai biến \'Cut\' và \'Color\' có mối quan hệ với nhau')
```

- Kết quả:

```
Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến 'Cut' và 'Color'.
Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến 'Cut' và 'Color'.
Bắc bỏ H0
Hai biến 'Cut' và 'Color' có mối quan hệ với nhau
```

```
[ ] # Kiểm định chi bình phương mối tương quan giữa cut và clarity
_, p ,_,_ = chi2_contingency(cut_clarity)
print('Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến \'Cut\' và \'Clarity\' .')
print('Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến \'Cut\' và \'Clarity\' .')
if p < 0.05:
    print('Bắc bỏ H0')
    print('Hai biến \'Cut\' và \'Clarity\' có mối quan hệ với nhau')
else:
    print('Chấp nhận H0')
    print('Không đủ bằng chứng để kết luận hai biến \'Cut\' và \'Clarity\' có mối quan hệ với nhau')
```

- Kết quả:

```
Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến 'Cut' và 'Clarity'.
Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến 'Cut' và 'Clarity'.
Bắc bỏ H0
Hai biến 'Cut' và 'Clarity' có mối quan hệ với nhau
```

```
[ ] # Kiểm định chỉ bình phương mối tương quan giữa color và clarity
_, p ,,_ = chi2_contingency(color_clarity)
print('Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến \'Color\' và \'Clarity\'')
print('Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến \'Color\' và \'Clarity\'')
if p < 0.05:
    print('Bắc bỏ H0')
    print('Hai biến \'Color\' và \'Clarity\' có mối quan hệ với nhau')
else:
    print('Chấp nhận H0')
    print('Không đủ bằng chứng để kết luận hai biến \'Color\' và \'Clarity\' có mối quan hệ với nhau')
```

Giả thuyết H0: Không có mối quan hệ hoặc không có sự tương quan giữa biến 'Color' và 'Clarity'.
 Giả thuyết H1: Có mối quan hệ hoặc có sự tương quan giữa biến 'Color' và 'Clarity'.
 Bắc bỏ H0
 Hai biến 'Color' và 'Clarity' có mối quan hệ với nhau

```
[ ] viz = processed_df.copy()
# Anova 1_way Table vs Cut
model = ols(formula='table ~ cut', data=viz).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
if anova_table['PR(>F)']['cut'] < 0.05:
    print('Có sự khác biệt có ý nghĩa thống kê giữa các nhóm cut về giá trị trung bình của biến table')
else:
    print('Không có sự khác biệt có ý nghĩa thống kê giữa các nhóm cut về giá trị trung bình của biến table')
```

- Kết quả:

	sum_sq	df	F	PR(>F)
cut	65049.443180	4.0	6127.422798	0.0
Residual	123223.935116	46429.0	Nan	Nan

Có sự khác biệt có ý nghĩa thống kê giữa các nhóm cut về giá trị trung bình của biến table

```
[ ] # Anova 1_way Depth vs Cut
model = ols(formula='depth ~ cut', data=viz).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
if anova_table['PR(>F)']['cut'] < 0.05:
    print('Có sự khác biệt có ý nghĩa thống kê giữa các nhóm cut về giá trị trung bình của biến depth')
else:
    print('Không có sự khác biệt có ý nghĩa thống kê giữa các nhóm cut về giá trị trung bình của biến depth')
```

- Kết quả:

	sum_sq	df	F	PR(>F)
cut	9721.577392	4.0	2552.237946	0.0
Residual	44212.483925	46429.0	Nan	Nan

Có sự khác biệt có ý nghĩa thống kê giữa các nhóm cut về giá trị trung bình của biến depth

3. Kiểm định Anova 2 chiều

Anova 2 chiều là một phương pháp thống kê được sử dụng để kiểm tra sự ảnh hưởng của hai biến độc lập (factors) đến một biến phụ thuộc (outcome). Cụ thể, nó thường được sử dụng khi bạn muốn kiểm tra sự ảnh hưởng của hai yếu tố đồng thời và xem xét xem có sự tương tác giữa chúng không.

Mục đích chính của Anova 2 chiều là kiểm tra xem có sự khác biệt ý nghĩa thống kê giữa các nhóm được tạo ra bởi hai yếu tố hay không. Nếu có sự tương tác giữa hai yếu tố, điều này chỉ ra rằng ảnh hưởng của một yếu tố có thể thay đổi tùy thuộc vào mức của yếu tố khác.

Giả thuyết H0 cho Anova 2 chiều thường có dạng như sau:

H0: Không có sự khác biệt có ý nghĩa thống kê giữa các nhóm được tạo ra bởi yếu tố 1.

H0: Không có sự khác biệt có ý nghĩa thống kê giữa các nhóm được tạo ra bởi yếu tố 2.

H0: Không có sự tương tác giữa yếu tố 1 và yếu tố 2.

Anova 2 chiều thực hiện tính toán các giá trị sum of squares (tổng bình phương), bậc tự do, giá trị F, và giá trị p để đánh giá xem sự khác biệt giữa các nhóm có ý nghĩa thống kê hay không.

Cụ thể, giá trị F được tính dựa trên tỷ lệ giữa biến thông kê giữa các nhóm và biến thông kê trong nhóm.

Giá trị p sau đó được sử dụng để quyết định xem có đủ bằng chứng để bác bỏ giả thuyết không có sự khác biệt hay không. Nếu giá trị p nhỏ hơn một ngưỡng ý nghĩa (thường là 0.05), ta bác bỏ giả thuyết H0 và kết luận rằng có sự khác biệt ý nghĩa thống kê.

Sử dụng hàm `ols` để tạo mô hình tuyến tính, trong đó biến 'price' là biến phụ thuộc là 'cut', 'clarity' và tương tác giữa 'cut' và 'clarity'.

Sử dụng hàm `anova_lm` với (typ=2) để thực hiện kiểm định Anova 2 chiều

Kết quả của Anova được lưu trong anova_table.

- Bảng Anova bao gồm các cột chính như sau:

- Sum_sq (Sum of Squares): tổng biến động các thành phần: biến động giữa các nhóm do 'cut' (C(cut)), clarity (C(clarity)), và biến động tương tác giữa cut và clarity (C(cut):C(clarity)).
- Df (Degrees of Freedom): bậc tự do tương ứng với từng thành phần biến động.
- F (F-statistic): Giá trị F được tính bằng cách chia tỷ lệ giữa biến động giữa các nhóm và biến động trong nhóm.
- PR(>F): Giá trị p, xác suất của giả thuyết H0 (không có sự khác biệt ý nghĩa thống kê).

3.1. Đánh giá ảnh hưởng của yếu tố 'cut' và 'clarity' đến giá 'price' của kim cương

Sử dụng hàm `ols` để tạo mô hình tuyến tính, trong đó biến 'price' là biến phụ thuộc là 'cut', 'clarity' và tương tác giữa 'cut' và 'clarity'.

Sử dụng hàm `anova_lm` với (typ=2) để thực hiện kiểm định Anova 2 chiều

Mô hình tuyến tính được xây dựng như sau: price ~ C(cut) + C(clarity) + C(cut):C(clarity)

```
[ ] # Price Vs Clarity and Cut
# Tạo mô hình ANOVA 2 chiều
pr_cl_ct = ols('price ~ C(cut) + C(clarity) + C(cut):C(clarity)', data=viz).fit()

# Kiểm định ANOVA
anova_table = sm.stats.anova_lm(pr_cl_ct, typ=2)

# In ra giả thuyết H0, H1
print('Giả thuyết H0: Không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và độ tinh khiết.')
print('Giả thuyết H1: Có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và độ tinh khiết.')

# In ra bảng ANOVA 2 chiều
print('Bảng ANOVA 2 chiều:')
print(anova_table)

# Danh sách các điều kiện
conditions = [
    ("Cut", 'C(cut)'),
    ("Clarity", 'C(clarity)'),
    ("Tương tác giữa Cut và Clarity", 'C(cut):C(clarity)')
]

# Kiểm tra từng điều kiện
print('\nPhân tích kết quả:')
for condition, col in conditions:
    p_value = anova_table.loc[col, 'PR(>F)']
    if p_value < 0.05:
        print(f"\t{condition} ảnh hưởng đến giá trị 'price'.")
        print(f"\t\tCó sự khác biệt ý nghĩa thống kê với giá trị 'price'.")
    else:
        print(f"\tKhông có sự khác biệt ý nghĩa thống kê với giá trị 'price' qua {condition}.")


```

Giả thuyết H0: Không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và độ tinh khiết.
Giả thuyết H1: Có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và độ tinh khiết.

Bảng ANOVA 2 chiều:

	sum_sq	df	F	PR(>F)
C(cut)	1.809467e+09	4.0	69.826839	4.748300e-59
C(clarity)	7.994496e+09	7.0	176.288846	9.205123e-29
C(cut):C(clarity)	1.366788e+09	28.0	7.534857	6.774818e-30
Residual	3.006305e+11	46405.0	NaN	NaN

Phân tích kết quả:
Cut ảnh hưởng đến giá trị 'price'. Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.
Clarity ảnh hưởng đến giá trị 'price'. Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.
Tương tác giữa Cut và Clarity ảnh hưởng đến giá trị 'price'. Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.

Bảng ANOVA 2 chiều cho thấy các giá trị sum of squares (sum_sq), bậc tự do (df), giá trị F, và giá trị p (PR(>F)) cho từng yếu tố và tương tác giữa chúng.

C(cut): Giả thuyết H0 là không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt. Giả thuyết này bị bác bỏ với giá trị $p < 0.05$.

C(clarity): Giả thuyết H0 là không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các độ tinh khiết. Giả thuyết này cũng bị bác bỏ với giá trị $p < 0.05$.

C(cut):C(clarity): Giả thuyết H0 là không có tương tác có ý nghĩa thống kê giữa "cut" và "clarity" đối với giá trung bình của kim cương. Giả thuyết này cũng bị bác bỏ với giá trị $p < 0.05$.

"cut," "clarity," và tương tác giữa chúng đều ảnh hưởng đến giá trị "price"

3.2. Đánh giá ảnh hưởng của yếu tố 'color' và 'clarity' đến giá 'price' của kim cương

Mô hình tuyến tính được xây dựng như sau: $\text{price} \sim \text{C(color)} + \text{C(clarity)} + \text{C(color)}:\text{C(clarity)}$

```
[ ] # Price vs Clarity và Color
# Tạo mô hình ANOVA 2 chiều
pr_cl_cr = ols('price ~ C(color) + C(clarity) + C(color):C(clarity)', data=viz).fit()

# Kiểm định ANOVA
anova_table = sm.stats.anova_lm(pr_cl_cr, typ=2)

# In ra giả thuyết H0, H1
print('Giả thuyết H0: Không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa độ tinh khiết và màu sắc.')
print('Giả thuyết H1: Có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa độ tinh khiết và màu sắc.')

# In ra bảng ANOVA 2 chiều:
print('Bảng ANOVA 2 chiều:')
print(anova_table)

# Danh sách các điều kiện
conditions = [
    ("Color", 'C(color)'),
    ("Clarity", 'C(clarity)'),
    ("Tương tác giữa Color và Clarity", 'C(color):C(clarity)')
]

# Kiểm tra từng điều kiện
print('\nPhân tích kết quả:')
for condition, col in conditions:
    p_value = anova_table.loc[col, 'PR(>F)']
    if p_value < 0.05:
        print(f"{condition} ảnh hưởng đến giá trị 'price'."
              f" Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.")
    else:
        print(f"Không có sự khác biệt ý nghĩa thống kê với giá trị 'price' qua {condition}.")
```

Giả thuyết H0: Không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa độ tinh khiết và màu sắc.
 Giả thuyết H1: Có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa độ tinh khiết và màu sắc.

Bảng ANOVA 2 chiều:

	sum_sq	df	F	PR(>F)
C(color)	7.969367e+09	6.0	211.041029	1.0009050e-266
C(clarity)	1.058158e+10	7.0	240.185575	0.000000e+00
C(color):C(clarity)	3.879132e+09	42.0	14.675049	2.240565e-102
Residual	2.919582e+11	46389.0	Nan	Nan

Phân tích kết quả:

Color ảnh hưởng đến giá trị 'price'. Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.
 Clarity ảnh hưởng đến giá trị 'price'. Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.
 Tương tác giữa Color và Clarity ảnh hưởng đến giá trị 'price'. Có sự khác biệt ý nghĩa thống kê với giá trị 'price'.

C(color): Giả thuyết H0 là không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức màu sắc. Giả thuyết này bị bác bỏ với giá trị $p < 0.05$.

C(clarity): Giả thuyết H0 là không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các độ tinh khiết. Giả thuyết này cũng bị bác bỏ với giá trị $p < 0.05$.

C(color):C(clarity): Giả thuyết H0 là không có tương tác có ý nghĩa thống kê giữa "color" và "clarity" đối với giá trung bình của kim cương. Giả thuyết này cũng bị bác bỏ với giá trị $p < 0.05$.

"color," "clarity," và tương tác giữa chúng đều ảnh hưởng đến giá trị "price" của kim cương

3.3. Đánh giá ảnh hưởng của yếu tố 'color' và 'cut' đến giá 'price' của kim cương

Mô hình tuyến tính được xây dựng như sau: $\text{price} \sim \text{C(color)} + \text{C(cut)} + \text{C(color)}:\text{C(cut)}$

```
[1] # Price vs Cut và Color
# Tạo mô hình ANOVA 2 chiều
pr_cr_ct = ols('price ~ C(color) + C(cut) + C(color):C(cut)', data=viz).fit()

# Kiểm định ANOVA
anova_table = sm.stats.anova_lm(pr_cr_ct, typ=2)

# In ra giả thuyết H0, H1
print("Giả thuyết H0: Không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và màu sắc.")
print("Giả thuyết H1: Có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và màu sắc.")

# In ra bảng ANOVA 2 chiều
print("Bảng ANOVA 2 chiều:")
print(anova_table)

# Danh sách các điều kiện
conditions = [
    ("Color", 'C(color)'),
    ("Cut", 'C(cut)'),
    ("Tương tác giữa Color và Cut", 'C(color):C(cut)')
]

# Kiểm tra từng điều kiện
print("\nPhân tích kết quả:")
for condition, col in conditions:
    p_value = anova_table.loc[col, 'PR(>F)']
    if p_value < 0.05:
        print(f'{condition} ảnh hưởng đến giá trị "price".')
        print(f" Có sự khác biệt ý nghĩa thống kê với giá trị "price".")
    else:
        print(f"Không có sự khác biệt ý nghĩa thống kê với giá trị "price" qua {condition}.")
```

Giả thuyết H0: Không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và màu sắc.
Giả thuyết H1: Có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt và màu sắc.
Bảng ANOVA 2 chiều:

	sum_sq	df	F	PR(>F)
C(color)	6.772002e+09	6.0	173.023112	1.469945e-218
C(cut)	3.199189e+09	4.0	122.607826	2.816574e-104
C(color):C(cut)	4.773119e+08	24.0	3.048802	7.265882e-07
Residual	3.027424e+11	46410.0	Nan	Nan

Phân tích kết quả:
Color ảnh hưởng đến giá trị "price". Có sự khác biệt ý nghĩa thống kê với giá trị "price".
Cut ảnh hưởng đến giá trị "price". Có sự khác biệt ý nghĩa thống kê với giá trị "price".
Tương tác giữa Color và Cut ảnh hưởng đến giá trị "price". Có sự khác biệt ý nghĩa thống kê với giá trị "price".

C(color): Giả thuyết H0 là không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức màu sắc. Giả thuyết này bị bác bỏ với giá trị $p < 0.05$.

C(cut): Giả thuyết H0 là không có sự khác biệt có ý nghĩa thống kê về giá trung bình của kim cương giữa các mức cắt. Giả thuyết này cũng bị bác bỏ với giá trị $p < 0.05$.

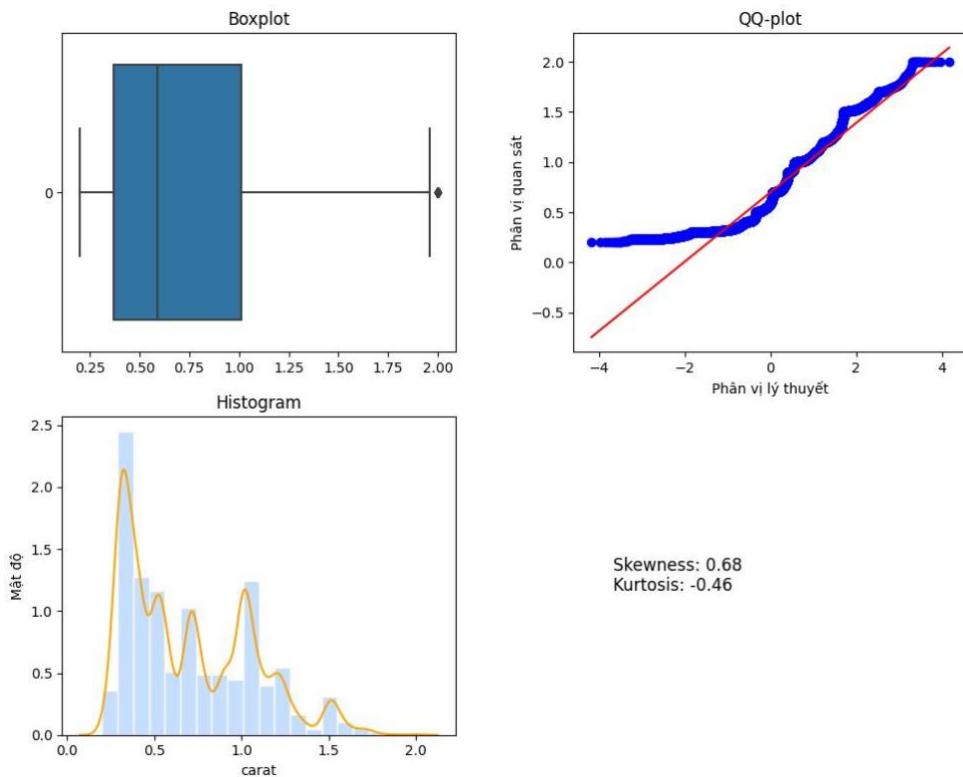
C(color):C(cut): Giả thuyết H0 là không có tương tác có ý nghĩa thống kê giữa "color" và "cut" đối với giá trung bình của kim cương. Giả thuyết này cũng bị bác bỏ với giá trị $p < 0.05$.

"color," "cut," và tương tác giữa chúng đều ảnh hưởng đến giá trị "price" của kim cương

Chương 5: Phân tích và trực quan hóa dữ liệu

1. Phân tích đơn biến

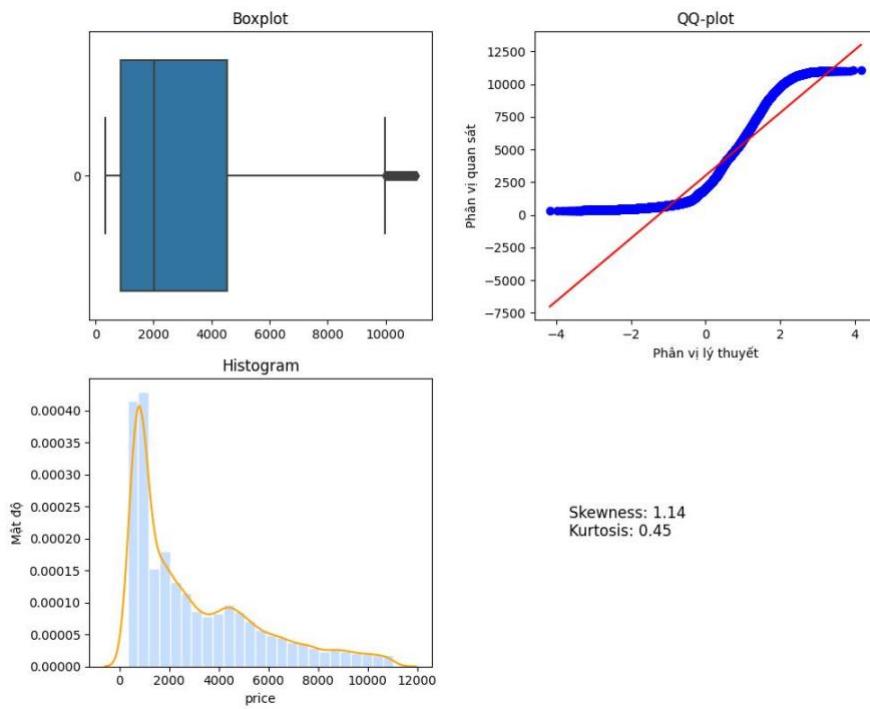
1.1. Carat



Hình 4.1: Các biểu đồ về hình dạng phân phối của biến Carat

- Quan sát histogram của biến Carat ta có thể thấy, trọng lượng của kim cương tập trung chủ yếu trong khoảng $(0.25, 0.5) \cup (1, 1.125)$ ct. Có thể thấy, trọng lượng của những viên kim cương thu thập được trong bộ dữ liệu tập trung vào hai mức là nhỏ và lớn nếu ta chia các khoảng carat $[0, 0.25, 0.5, 1, 2.5]$ tương đương với phân loại [Rất nhỏ, Nhỏ, Bình thường, Lớn].
- Boxplot của biến Carat cho biết outliers của thuộc tính này chính là những viên kim cương có kích thước lớn hơn 2 ct.
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $0.68 > 0$ cho biết dữ liệu hơi lệch phải và giá trị kurtosis = $-0.46 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

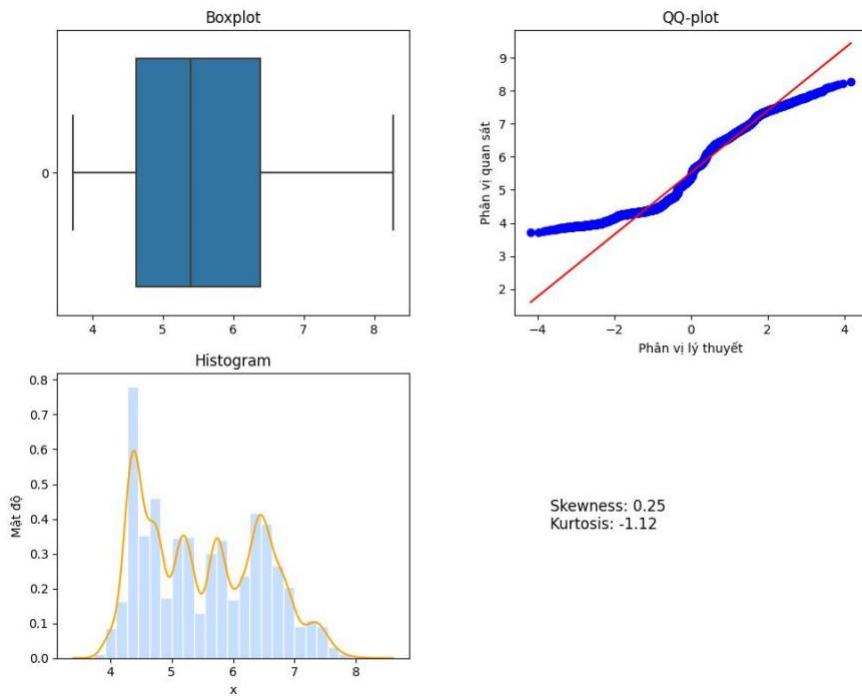
1.2. Price



Hình 5.2: Các biểu đồ về hình dạng phân phối của biến Price

- Quan sát histogram của biến Price ta có thể thấy, giá của kim cương tập trung chủ yếu trong khoảng (0, 2000) USD. Có thể thấy, giá của những viên kim cương thu thập được tập trung nhiều vào mức giá phổ thông và mức giá rẻ, tiếp cận được đến đa dạng tập khách hàng hơn.
Mức giá phổ thông là mức giá được phân loại nếu ta giả định chia các khoảng giá [0,1500,4000,10000,90000] tương ứng với các mức [Mức giá phổ thông, Mức giá rẻ, Mức giá đắt, Mức giá cao cấp].
- Boxplot của biến Price cho biết outliers của thuộc tính này chính là những viên kim cương có giá lớn hơn 10000 USD.
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $1.14 > 0$ cho biết dữ liệu lệch phải và giá trị kurtosis = $-0.45 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

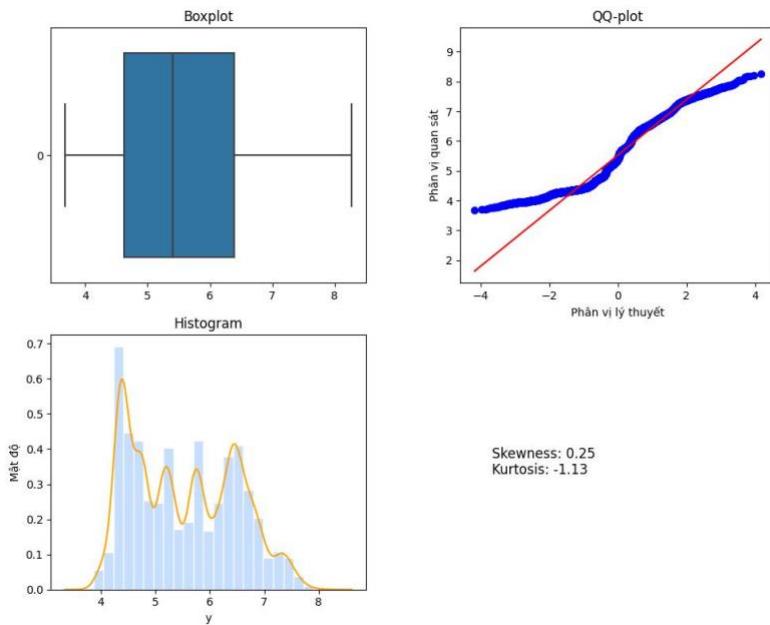
1.3. X



Hình 5.3: Các biểu đồ về hình dạng phân phối của biến X

- Quan sát histogram của biến X ta có thể thấy, chiều dài của kim cương tập trung chủ yếu trong khoảng $(4.25, 4.5) \cup (4.75, 5) \cup (6.5, 6.75)$ mm.
- Boxplot của biến X cho biết, sau khi tiền xử lý dữ liệu, biến không còn xuất hiện outliers.
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $0.25 > 0$ cho biết dữ liệu có xu hướng lệch phải và giá trị kurtosis = $-1.12 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

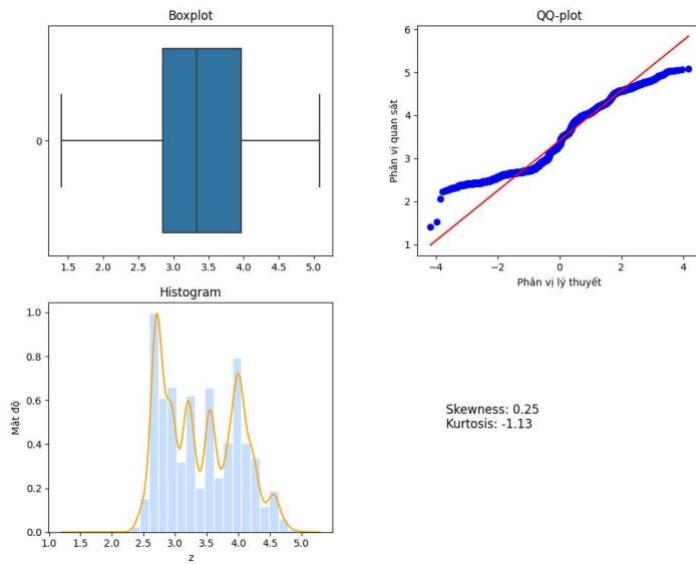
1.4. Y



Hình 5.4: Các biểu đồ về hình dạng phân phối của biến Y

- Quan sát histogram của biến Y ta có thể thấy, chiều rộng của kim cương tập trung chủ yếu trong khoảng $(4.5, 4.75) \cup (5.125, 5.5) \cup (5.75, 6)$ mm.
- Boxplot của biến Y cho biết, sau khi tiền xử lý dữ liệu, biến không còn xuất hiện outliers.
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $0.25 > 0$ cho biết dữ liệu có xu hướng lệch phải và giá trị kurtosis = $-1.13 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

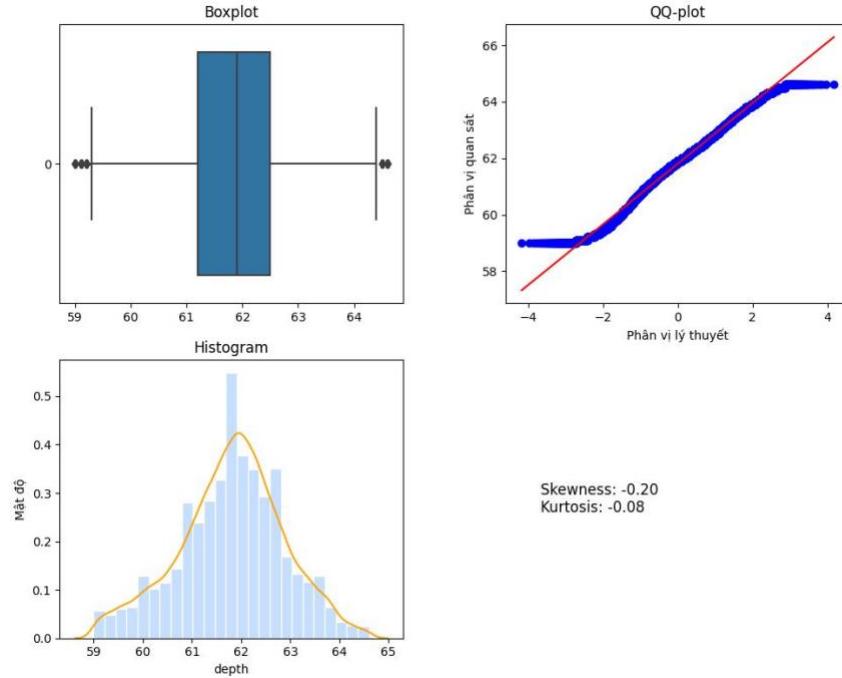
1.5. Z



Hình 5.5: Các biểu đồ về hình dạng phân phối của biến Z

- Quan sát histogram của biến Z ta có thể thấy, chiều cao/sâu của kim cương tập trung chủ yếu trong khoảng $(2.75, 3) \cup (3.25, 3.375) \cup 3.5 \cup 4$ mm.
- Boxplot của biến Z cho biết, sau khi tiền xử lý dữ liệu, biến không còn xuất hiện outliers.
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $0.25 > 0$ cho biết dữ liệu có xu hướng lệch phải và giá trị kurtosis = $-1.13 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

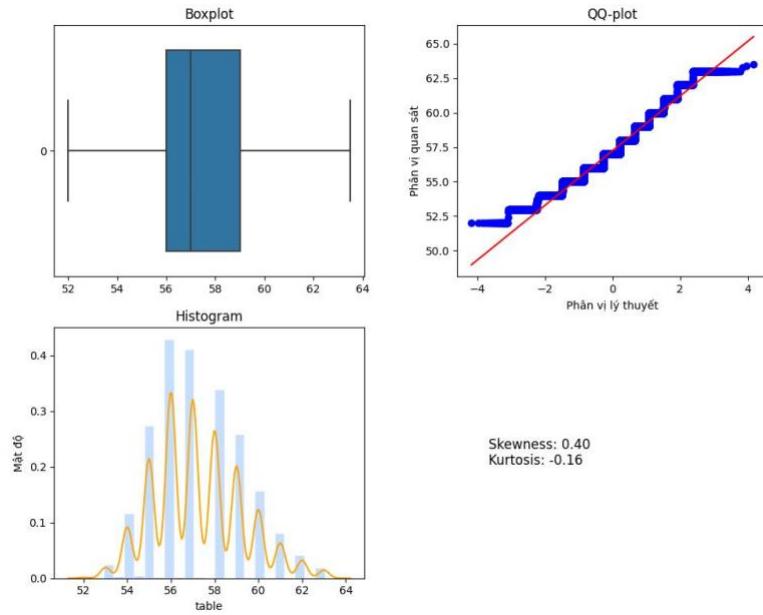
1.6. Depth



Hình 5..6: Các biểu đồ về hình dạng phân phối của biến Depth

- Quan sát histogram của biến Depth ta có thể thấy, chiều cao/sâu của kim cương tập trung chủ yếu trong khoảng (61.25, 62.75) %.
- Boxplot của biến Depth cho biết, sau khi tiền xử lý dữ liệu, outliers của biến là các giá trị lớn hơn 64.5 và nhỏ hơn 59.25
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $-0.2 < 0$ cho biết dữ liệu có xu hướng lệch nhẹ về phía bên trái và giá trị kurtosis = $-0.08 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

1.7. Table

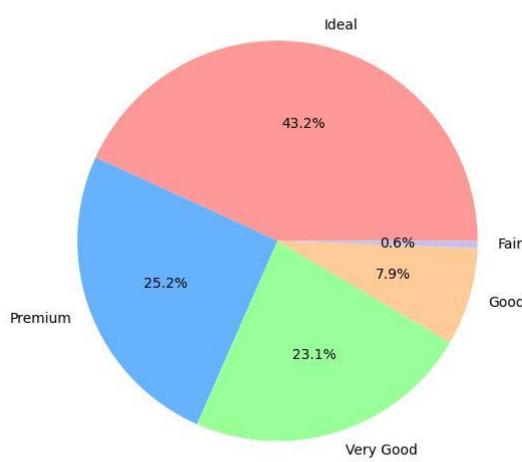


Hình 5.7: Các biểu đồ về hình dạng phân phối của biến Table

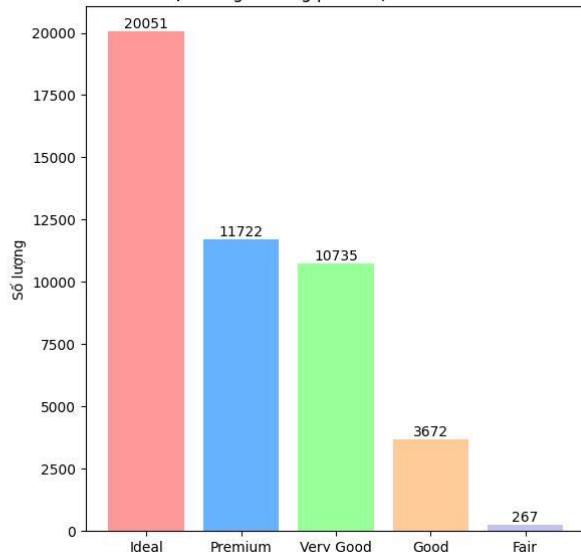
- Quan sát histogram của biến Table ta có thể thấy, độ rộng của phần đỉnh kim cương so với trung bình chiều dài và chiều rộng tập trung chủ yếu trong đoạn [55,59]
- Boxplot của biến Table cho biết, sau khi tiền xử lý dữ liệu, biến không còn xuất hiện outliers.
- Phần đuôi của các điểm dữ liệu trong QQ-plot nằm phía trên đường phân phối chuẩn và phần đầu của các điểm dữ liệu nằm phía dưới đường phân phối chuẩn cho biết tập dữ liệu có phương sai < 1 , thể hiện dữ liệu có xu hướng biến động ít hơn so với phân phối chuẩn.
- Giá trị skewness = $0.4 > 0$ cho biết dữ liệu có xu hướng lệch phải và giá trị kurtosis = $-0.16 < 3$ cho biết đỉnh của phân phối bẹt và thấp hơn so với đỉnh của phân phối chuẩn.

1.8. Cut

Biểu đồ tròn thể hiện tỉ lệ % phân loại vết cắt của biến 'Cut'



Biểu đồ cột thống kê từng phân loại vết cắt của biến 'Cut'

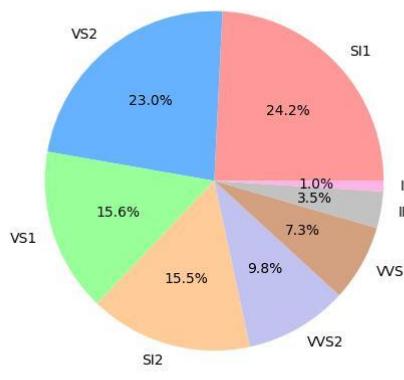


Hình 5.8: Các biểu đồ thể hiện phân phối của từng nhóm vết cắt

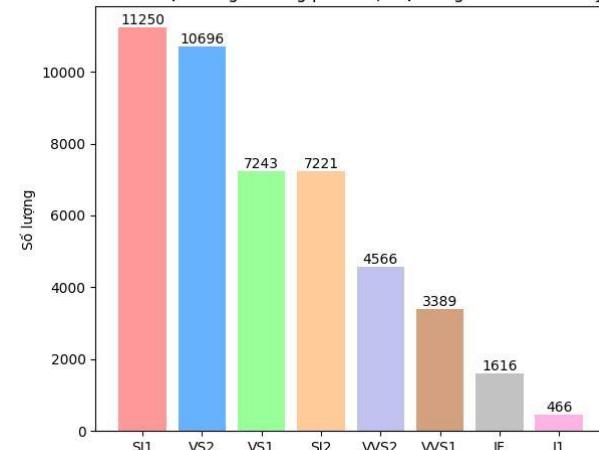
- Quan sát biểu đồ trên, ta có thể thấy, nhóm vết cắt được đánh giá là Ideal chiếm tỷ lệ cao nhất trong số các phân loại (43.2%), theo sau đó lần lượt là nhóm vết cắt Premium và Very Good với tỷ lệ là 25.2% và 23.1%.
- Có thể rút ra kết luận rằng, phần lớn số lượng kim cương được ghi nhận trong bộ dữ liệu có chất lượng vết cắt rất lý tưởng.

1.9. Clarity

Biểu đồ tròn thể hiện tỉ lệ % phân loại độ trong của biến 'Clarity'



Biểu đồ cột thống kê từng phân loại độ trong của biến 'Clarity'

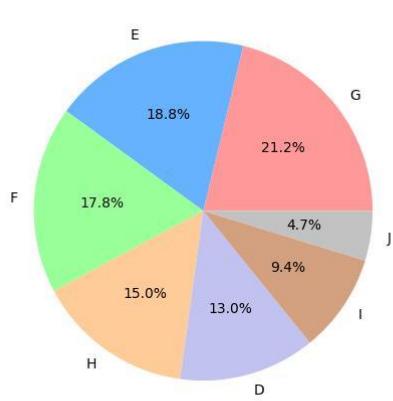


Hình 5.9: Các biểu đồ thể hiện phân phối của từng nhóm phân loại độ tinh khiết

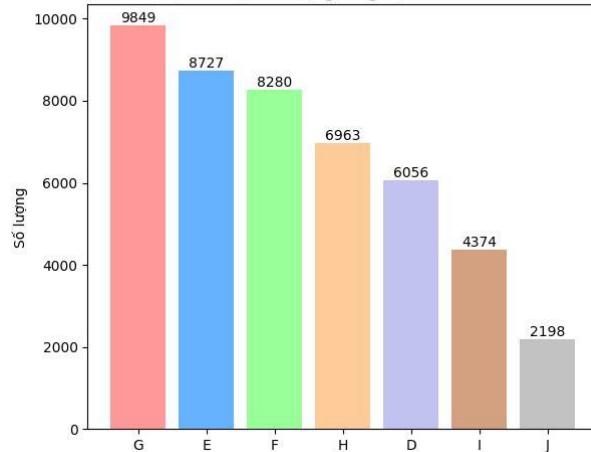
- Quan sát biểu đồ trên, ta có thể thấy, độ tinh khiết cao nhất là SI1 (24.2%), sau đó là VS2 (23.0%).
- Xếp vị trí thứ ba và thứ tư về tỷ lệ phân loại độ tinh khiết của dữ liệu kim cương lần lượt là nhóm phân loại độ tinh khiết VS1 (15.6%) và SI2 (15.5%).

- Hai nhóm phân loại độ tinh khiết chiếm tỷ lệ rất nhỏ trong các nhóm phân loại độ tinh khiết là IF (3.5%) và I1 (1%).
- Có thể rút ra kết luận, độ tinh khiết của các viên kim cương được ghi nhận phần lớn đều thuộc nhóm có tạp chất (SI1, SI2, VS1, VS2). Trên tổng số dữ liệu, số viên kim cương có tạp chất và ảnh hưởng đến tính thẩm mỹ chiếm 39.7%.
- Số lượng kim cương có tạp chất rõ ràng, ảnh hưởng đến tính thẩm mỹ và độ bền chiếm tỷ lệ rất ít trong số dữ liệu được thu thập (I1). Cũng tương tự như vậy, những viên kim cương với độ tinh khiết hoàn hảo, không tạp chất cũng chiếm tỷ lệ khá thấp khi thu thập dữ liệu (IF). Điều này chứng tỏ rằng, những viên kim cương với độ tinh khiết không hoàn hảo kém được ưa chuộng và những viên kim cương với độ tinh khiết hoàn mỹ thì rất hiếm và khó tìm.

Biểu đồ tròn thể hiện tỉ lệ % các loại màu của biến 'Color'



Biểu đồ cột thể hiện số lượng từng loại màu của biến 'Color'



Hình 5.10: Các biểu đồ thể hiện phân phối của từng nhóm phân loại sắc màu

1.10 Color

- Quan sát biểu đồ trên, ta có thể thấy các nhóm phân loại màu phổ biến trong dữ liệu về kim cương được ghi nhận là G (21.2%), E (18.8%), F (17.8%).
- Có thể thấy, phần lớn số kim cương được ghi nhận có nhóm màu thuộc loại trong suốt (E,F) và gần như trong suốt (G), phản ánh chất lượng màu rất tốt.

2. Phân tích 2 biến

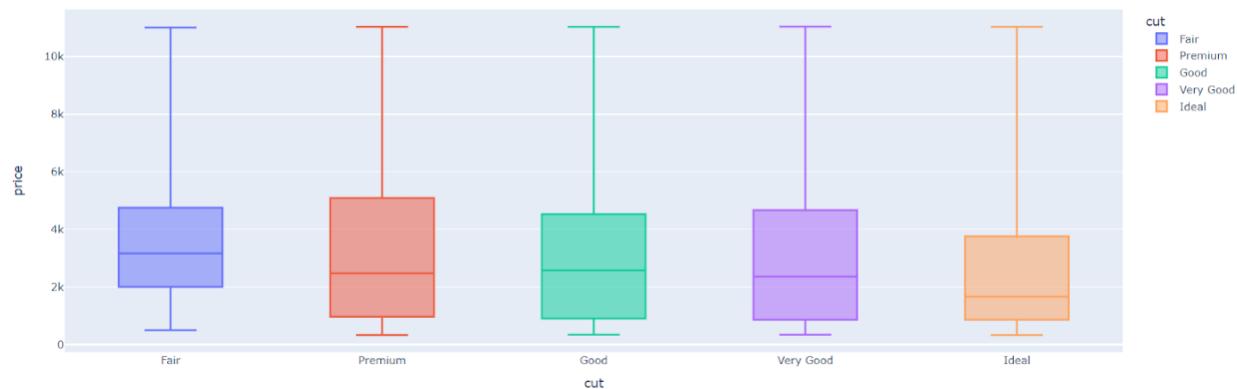
2.1. Các biến categories ảnh hưởng đến giá của kim cương như thế nào?

2.1.1. Vết cắt của viên kim cương ảnh hưởng đến giá kim cương.

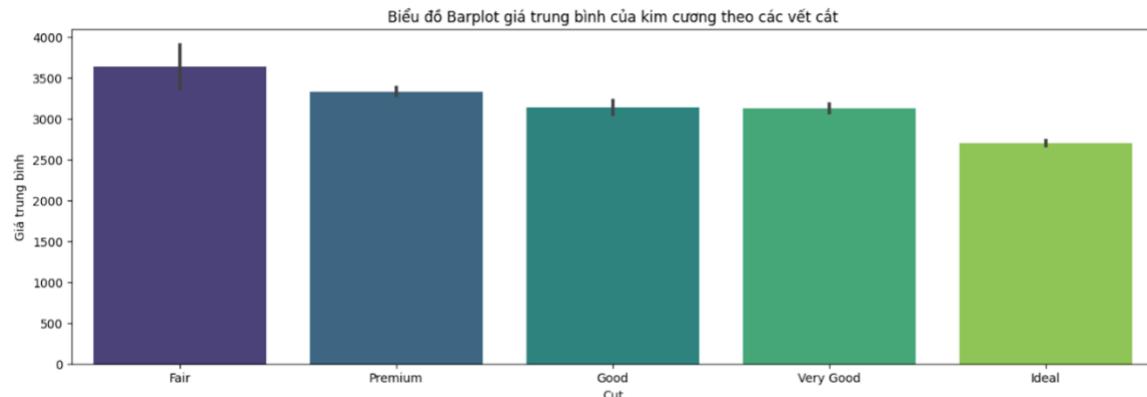
	Diamond Cut	Count	Percentage	Minimum Price	Average Price	Maximum Price
0	Fair	267	0.6	496	3646.28	11007
1	Good	3671	7.9	335	3140.91	11036
2	Ideal	20049	43.2	326	2701.28	11040
3	Premium	11717	25.2	326	3336.49	11037
4	Very Good	10733	23.1	336	3126.03	11039

Bảng 2.1.1: Thông kê vết cắt của kim cương

Biểu đồ Boxplot giá kim cương theo loại vết cắt



Hình 4.11: Biểu đồ boxplot thể hiện giá kim cương theo vết cắt



Hình 4.12: Biểu đồ barplot giá trung bình của kim cương theo vết cắt

Nhận xét:

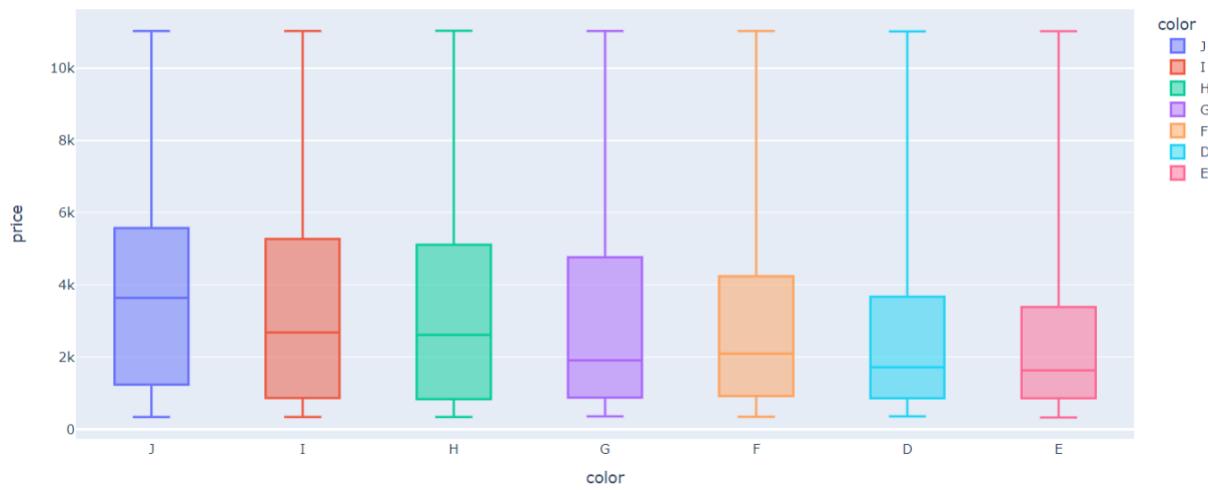
- Các loại vết cắt của viên kim cương được phân loại thành 5 nhóm: Fair, Good, Ideal, Premium, và Very Good. Nhóm kim cương "Ideal" chiếm đa số, với 43.2% số lượng kim cương trong bộ dữ liệu. Nhóm "Fair" có số lượng thấp nhất, chỉ chiếm 0.6% tổng số kim cương. Các nhóm "Good", "Premium" và "Very Good" chiếm phần còn lại, với tỉ lệ tương đối gần nhau.
- Kim cương được phân thành các nhóm với độ chênh lệch về giá khá đáng kể. Ideal là loại có chất lượng vết cắt cao nhất nhưng lại có giá trung bình thấp nhất, trong khi Fair là loại có chất lượng cắt thấp lại có giá trung bình cao nhất.

2.1.2. Màu sắc của viên kim cương ảnh hưởng đến giá kim cương

Index	Diamond Clarity	Count	Percentage	Minimum Price	Average Price	Maximum Price
0	D	6056	13	357	2570.6	11023
1	E	8726	18.8	326	2491.37	11028
2	F	8280	17.8	342	2955.72	11040
3	G	9845	21.2	354	3092.9	11032
4	H	6960	15	337	3340.5	11039
5	I	4373	9.4	334	3523.24	11040
6	J	2197	4.7	335	3832.48	11036

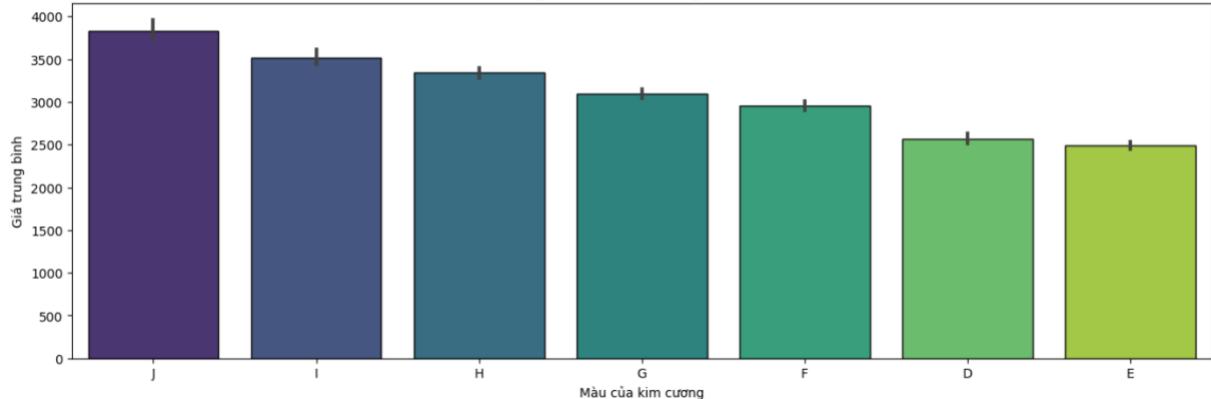
Bảng 2.1.2: Thống kê màu sắc viên kim cương

Biểu đồ Boxplot giá kim cương theo màu sắc kim cương



Hình 4.13: Biểu đồ boxplot giá kim cương theo màu sắc

Giá trung bình của kim cương theo màu sắc



Hình 4.14: Biểu đồ barplot giá trung bình kim cương theo màu sắc

Nhận xét chung:

- Tương tự kim cương có màu J (tệ nhất) lại có giá trung bình cao nhất và kim cương màu D (tốt nhất) lại có giá trung bình thấp hơn.
- Kim cương loại D chiếm 13% tổng số lượng, trong khi kim cương loại J chỉ chiếm 4.7%. Tỷ lệ phân bố này thể hiện độ đa dạng trong thị trường kim cương.

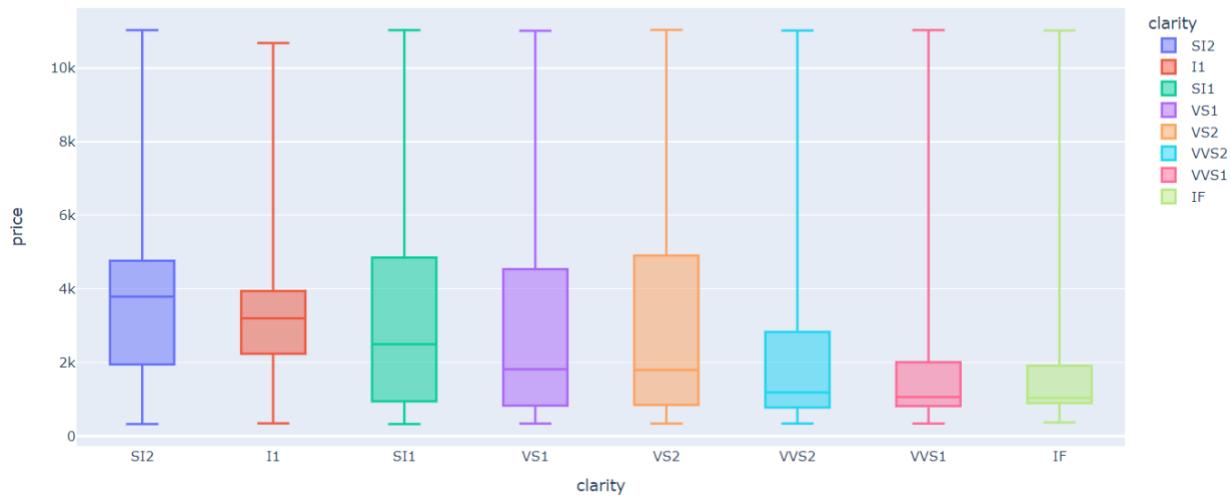
- Kim cương loại D có giá trung bình thấp nhất là 2570.24 đô la, trong khi kim cương loại J có giá trung bình cao nhất là 3835.4 đô la.

2.1.3. Độ trong suốt của viên kim cương ảnh hưởng đến giá kim cương.

Index	Diamond Clarity	Count	Percentage	Minimum Price	Average Price	Maximum Price
0	I1	466	1	345	3292.79	10685
1	IF	1617	3.5	369	1981.15	11025
2	SI1	11244	24.2	326	3131.89	11040
3	SI2	7220	15.5	326	3697.85	11037
4	VS1	7243	15.6	338	3045.36	11019
5	VS2	10693	23	334	2986.24	11039
6	VVS1	3388	7.3	336	1974.11	11033
7	VVS2	4566	9.8	336	2623.33	11040

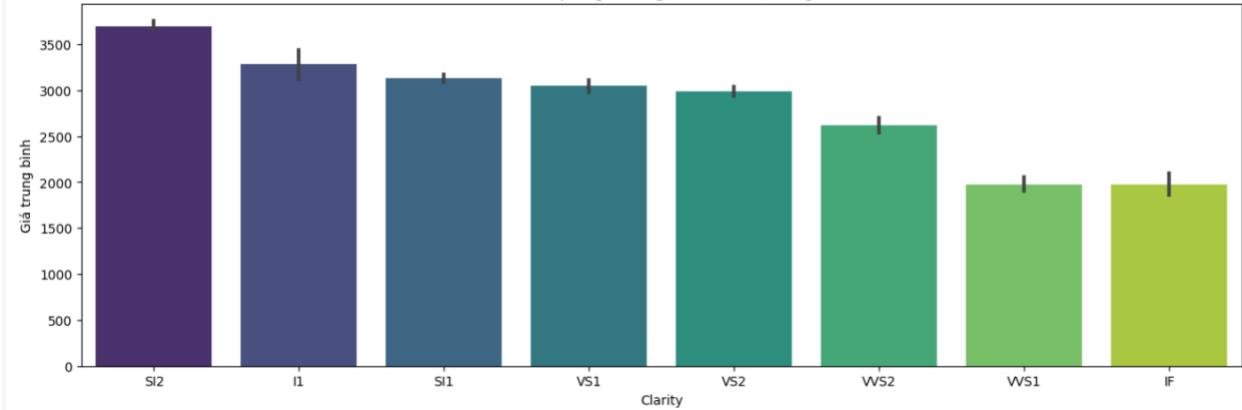
Bảng 2.1.3: Thông kê độ trong suốt của viên kim cương

Biểu đồ Boxplot giá kim cương theo độ trong suốt



Hình 4.15: Biểu đồ Boxplot thể hiện giá kim cương theo độ trong suốt

Biểu đồ Barplot giá trung bình theo độ trong suốt



Hình 4.16: Biểu đồ Barplot thể hiện giá trung bình của kim cương theo độ trong suốt

Nhận xét chung:

- Qua các biểu đồ, khi xem xét độ trong suốt của kim cương theo giá kim cương, độ trong tốt nhất là IF dường như có giá thấp hơn so với I1 (độ trong kém nhất).

Nhận xét về categorical đến price:

Từ việc trực quan các biến categorical đến price, việc các biến bị tương quan nghịch như vết cắt Ideal tốt nhất nhưng có giá trung bình thấp nhất còn vết cắt Fair trung bình lại có giá trung bình cao nhất, kim cương có màu J (tệ nhất) lại có giá trung bình cao nhất và kim cương màu D (tốt nhất) và độ trong tốt nhất là IF dường như có giá thấp hơn so với I1 (độ trong kém nhất) lại có giá trung bình thấp hơn. Có thể khi trực quan từng biến đến giá kim cương chúng ta chưa có đủ thông tin để đưa ra những kết luận chính xác được. Do đó, cần xét các trường hợp khác để có thể biết được tác động đến price như thế nào.

2.2. Các biến numeric (biến số) ảnh hưởng đến giá của kim cương như thế nào?

- Để quan sát mối quan hệ giữa các biến số với biến giá cả, ta tiến hành vẽ heatmap độ tương quan để xem xét độ tương quan giữa các cặp biến.



Hình 5.17: Heatmap thể hiện độ tương quan của các biến số

- Quan sát Heatmap, ta nhận thấy:

Có sự tương quan lớn giữa biến Price và {X, Y, Z, Carat}.

Carat và {X, Y, Z} tương quan nhiều và {X, Y, Z} cũng tương quan nhiều với nhau

Các nhóm biến {Depth, X, Y} không có tương quan tuyến tính.

Các nhóm biến {Depth, Price}, {Depth, Carat}, Table với {Carat, X, Y, Z, Price} tương quan ít.

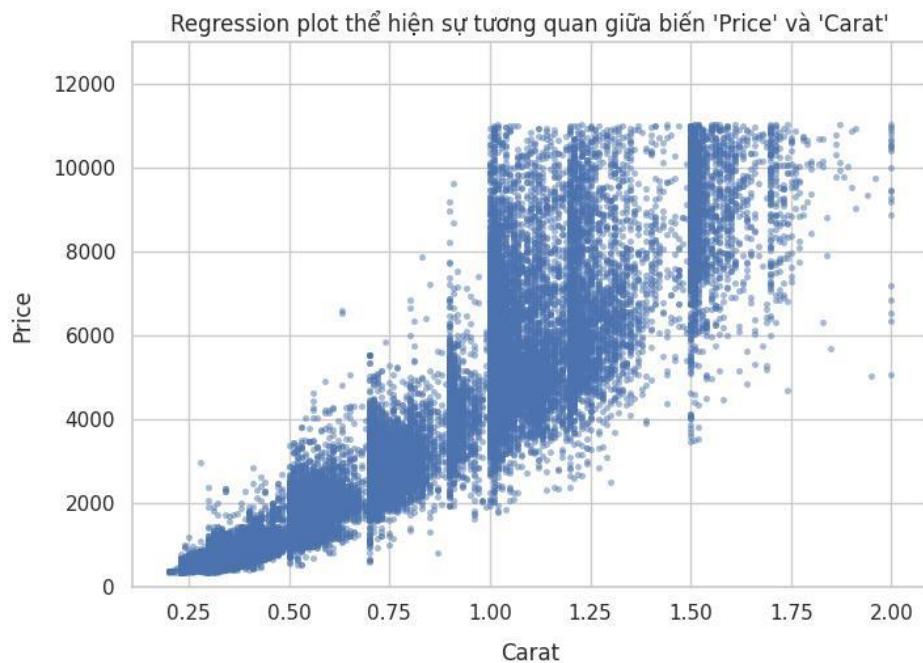
Biến Depth với Table tương quan tuyến tính theo hướng nghịch đảo. Khi một biến tăng, biến còn lại giảm và ngược lại.

- Ta tiến hành trực quan hóa để kiểm tra những nhận định trên.

2.2.1. Các biến có độ tương quan cao với Price

2.2.1.1. Price và Carat

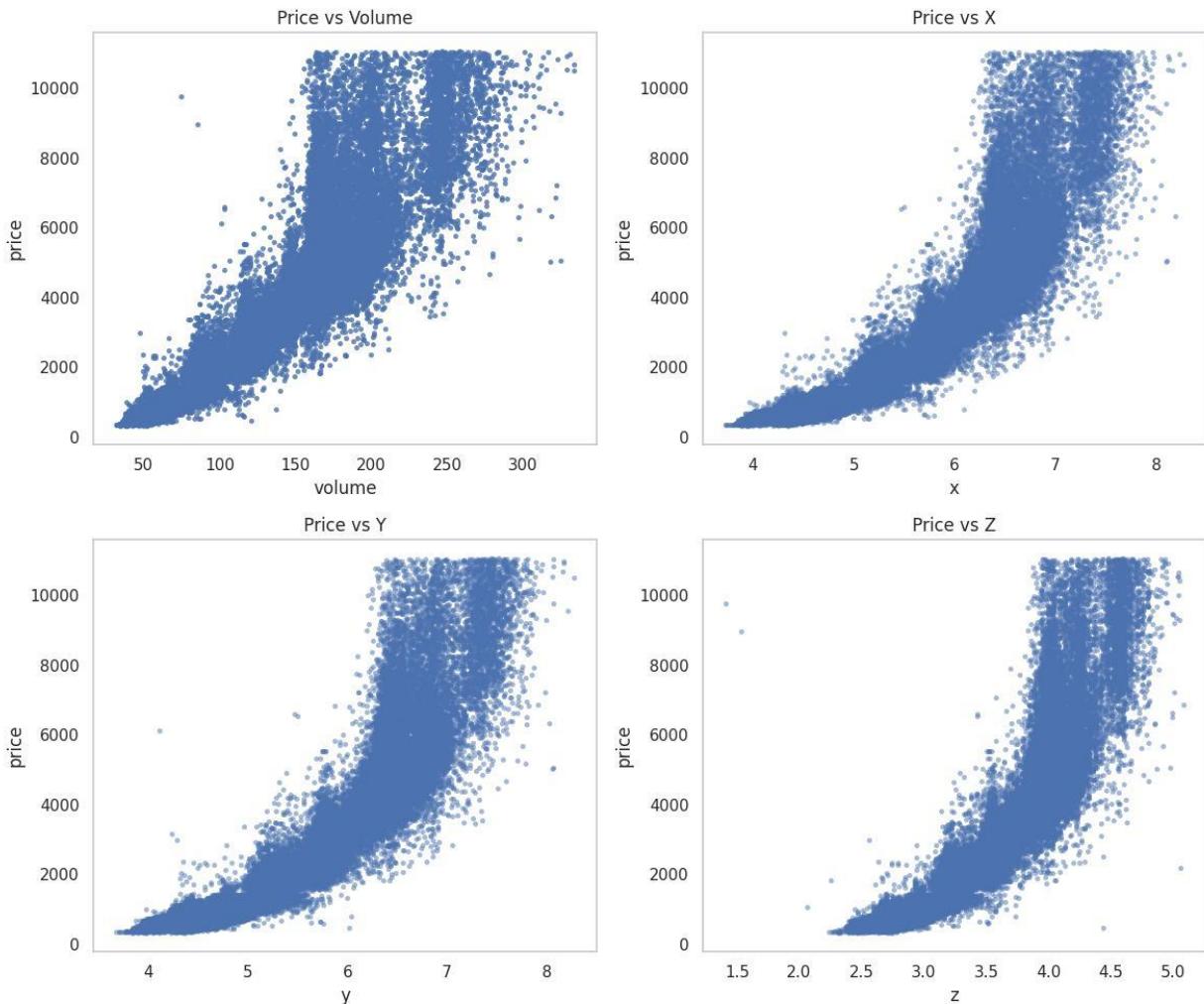
- Quan sát biểu đồ ta có thể thấy, các điểm trong biểu đồ tạo thành đường cong hướng lên trên, tức là khi trọng lượng tăng thì giá trị của kim cương cũng tăng theo.
- Vậy có thể kết luận, biến biến Price và biến Carat có độ tương quan dương cao.



Hình 5.18: Regression plot thể hiện sự tương quan giữa biến Price và Carat

2.2.1.2. Price và X, Y, Z, Volume

- Volume là thể tích của viên kim cương, được tính bằng tích của các biến X, biến Y, và biến Z.
- Quan sát biểu đồ ta có thể thấy biến Price và X, Y, Z, Volume tạo thành đường cong, hướng lên, biểu thị một mối tương quan dương cao.



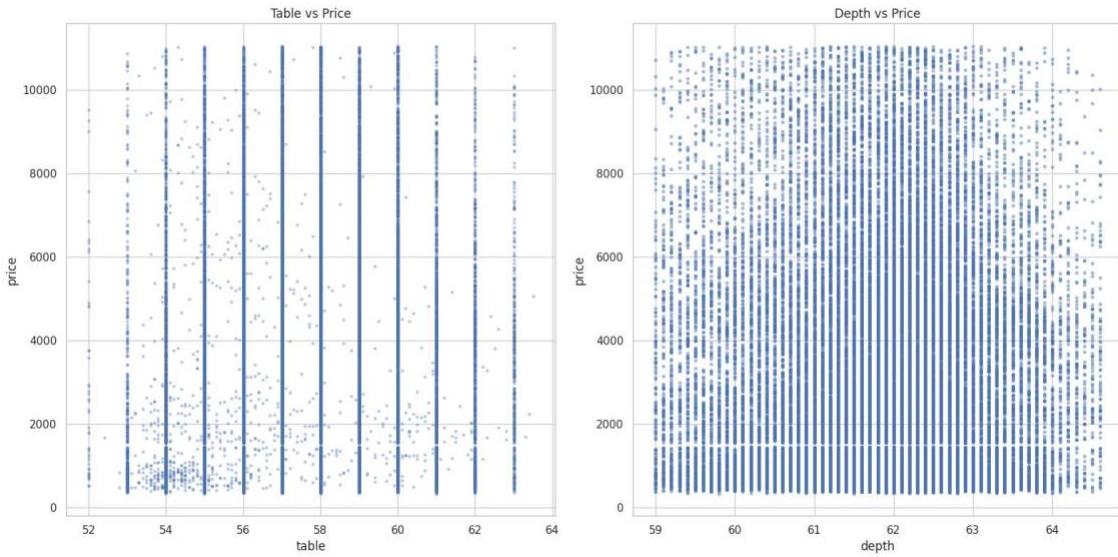
Hình 5.19: Regression plot thể hiện sự tương quan giữa biến Price và X, Y, Z, Volume

➔ Kết luận: Các biến có độ tương quan cao với Price bao gồm: {X, Y, Z, Volume, Carat}.

2.2.2. Các biến có độ tương quan thấp với Price

2.2.2.1. Price và Table, Depth

- Quan sát biểu đồ, ta có thể thấy các điểm dữ liệu trên biểu đồ phân tán không theo quy luật, gần như không thể hiện mối quan hệ nào với nhau.
- Có thể kết luận, biến Table và biến Depth có độ tương quan rất thấp và gần như không tương quan với biến Price.



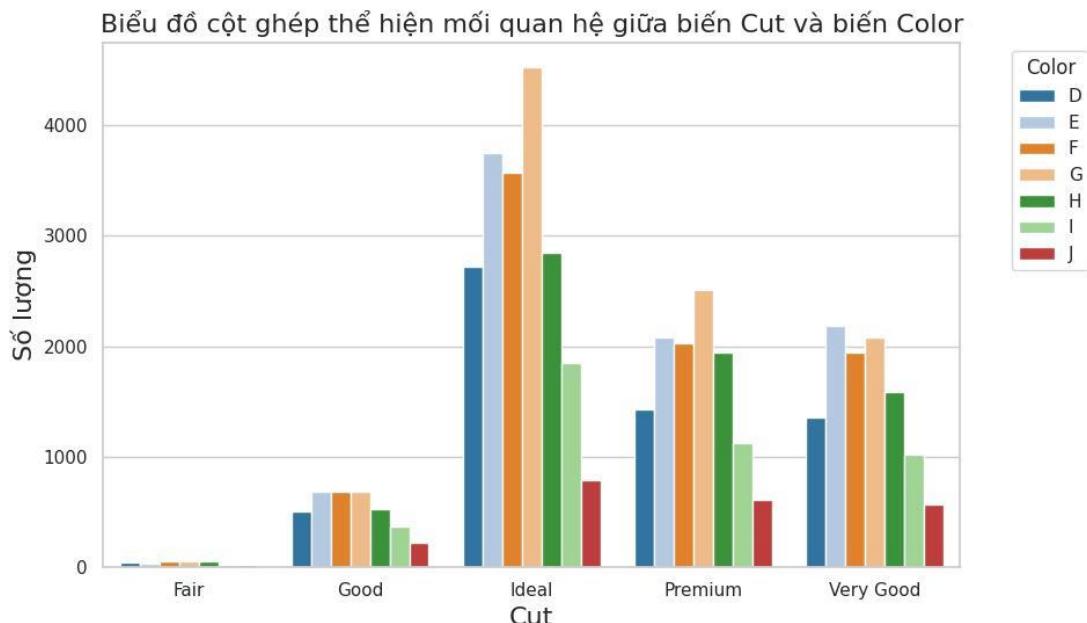
Hình 5.20: Regression plot thể hiện sự tương quan giữa biến Price và Table, Price và Depth

2.3. Bên cạnh mối tương quan với biến Price, giữa các biến còn có sự tương quan nào khác với nhau không?

2.3.1. Mối tương quan giữa các cặp biến phân loại

2.3.1.1. Cut và Color

- Quan sát biểu đồ cột nhóm ta thấy, các nhóm vết cắt được đánh giá là Good, Very Good, Premium, Ideal đều có 3 nhóm màu chiếm tỷ lệ cao nhất là G, F, E. Thứ tự nhóm màu phổ biến ở nhóm vết cắt Ideal và Premium là G, E, F; còn ở nhóm vết cắt Very Good là E, G, F. Nhóm vết cắt Good có tỷ lệ nhóm màu phổ biến G, E, F ngang nhau.
- Nhóm vết cắt Fair thì có những nhóm màu chiếm tỷ lệ cao nhất theo thứ tự lần lượt là G, H. Nhóm màu J xuất hiện rất ít ở nhóm vết cắt này và nhóm màu I gần như không xuất hiện.
- Có thể nhận thấy ở những nhóm vết cắt có chất lượng cao hơn (Good, Very Good, Premium, Ideal) thì sẽ chứa nhiều nhóm màu nằm trong nhóm Colorless (Không màu) hơn (nhóm màu E, F). Còn nhóm vết cắt có chất lượng chấp nhận được (Fair) thì nhóm màu sẽ chủ yếu thuộc nhóm Near Colorless - Gần như không màu - (G, H) với chất lượng màu kém hơn nếu so với nhóm Colorless.

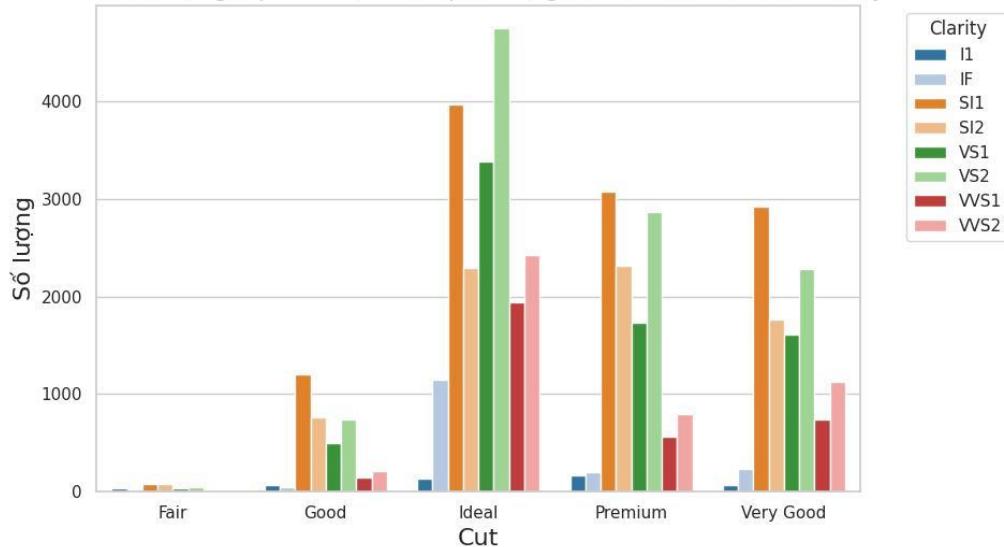


Hình 5.21: Biểu đồ cột nhóm thể hiện mối tương quan của biến Cut và biến Color

2.3.1.2. Cut và Clarity

- Quan sát biểu đồ cột nhóm ta thấy nhóm vết cắt tốt nhất (Ideal) có top 3 tỷ lệ độ tinh khiết phổ biến nhất theo thứ tự là VS2, SI1, VS1.
- Thứ tự này ở nhóm vết cắt Very Good và Premium là SI1, VS2, SI2. Còn ở nhóm Good là SI1, SI2, VS2.
- Tại nhóm vết cắt Fair, top 2 tỷ lệ độ tinh khiết phổ biến nhất lần lượt là SI2, SI1.
- Có thể nhận thấy, ở các nhóm vết cắt có chất lượng tốt (Good, Very Good, Premium, Ideal) thì phần lớn độ tinh khiết của chúng vẫn có tạp chất (SI2, SI1, VS2, VS1 đều thuộc nhóm độ tinh khiết có chứa tạp chất). Tuy nhiên ở nhóm vết cắt với chất lượng lý tưởng nhất (Ideal) thì phần lớn những tạp chất này không ảnh hưởng đến tính thẩm mỹ. Còn ở những nhóm vết cắt với chất lượng tốt nhưng chưa đạt đến lý tưởng (Good, Very Good, Premium) và ở những nhóm vết cắt có chất lượng trung bình (Fair), thì các tạp chất có trong độ tinh khiết chiếm đa số có thể ảnh hưởng đến tính thẩm mỹ của kim cương.

Biểu đồ cột ghép thể hiện mối quan hệ giữa biến Cut và biến Clarity

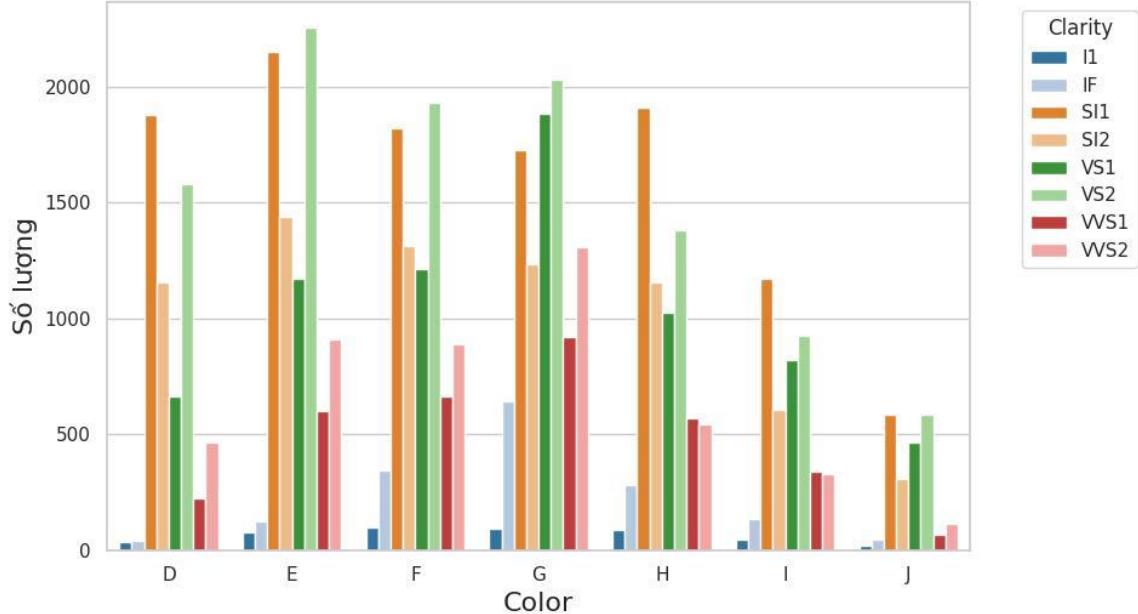


Hình 5.22: Biểu đồ cột nhóm thể hiện mối tương quan của biến Cut và biến Clarity

2.3.1.3. Color và Clarity

- Quan sát biểu đồ cột nhóm ta thấy 4 nhóm độ tinh khiết phổ biến nhất ở các nhóm màu là SI1, SI2, VS1, VS2.
- Các nhóm màu D, H, I, J đều có top 2 tỷ lệ nhóm độ tinh khiết theo thứ tự là SI1, VS2.
- Tỷ lệ này ở nhóm màu E, F là VS2, SI1 và ở nhóm màu G là VS2, VS1.
- Có thể nhận thấy, các viên kim cương với nhóm màu D (xếp hạng cao nhất trong nhóm Colorless) và các viên kim cương với nhóm màu H, I, J (lần lượt là thứ hạng từ 2-4 trong nhóm Near Colorless) trong bộ dữ liệu thu thập được tuy có màu sắc khá lý tưởng nhưng độ tinh khiết vẫn còn chứa tạp chất và vẫn còn nhiều viên kim cương có tính thẩm mỹ bị ảnh hưởng bởi những tạp chất này.
- Những viên kim cương với nhóm màu E, F (thứ hạng 2,3 trong nhóm Colorless) và G (thứ hạng cao nhất trong nhóm Near Colorless) có tỷ lệ độ tinh khiết lẩn tạp chất nhưng không ảnh hưởng đến tính thẩm mỹ cao hơn tỷ lệ độ tinh khiết lẩn tạp chất và ảnh hưởng đến tính thẩm mỹ.

Biểu đồ cột ghép thể hiện mối quan hệ giữa biến Color và biến Clarity

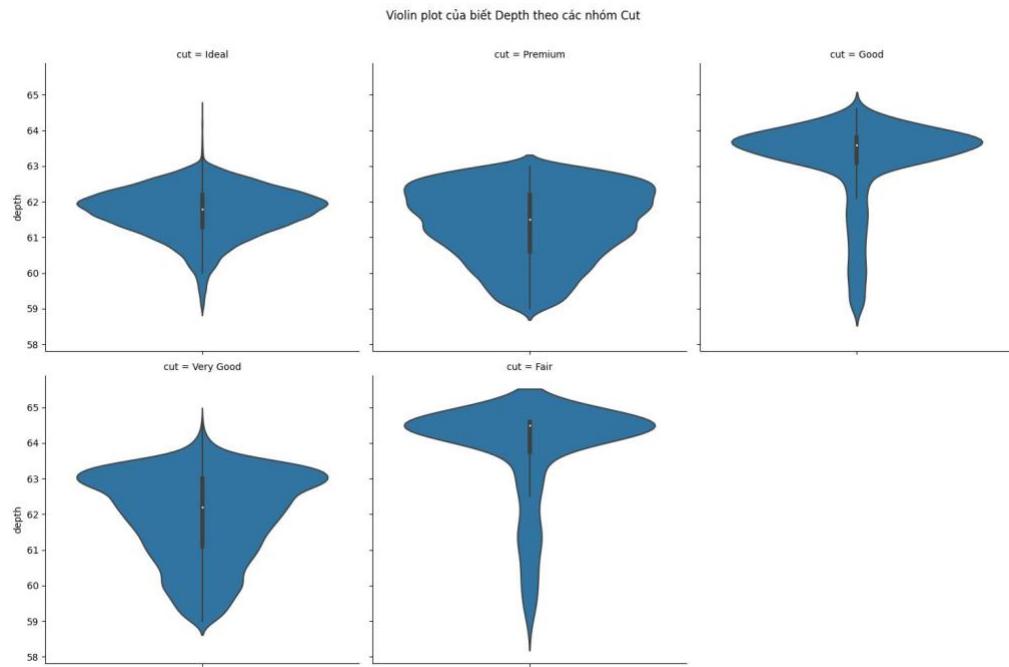


Hình 5.23: Biểu đồ cột nhóm thể hiện mối tương quan của biến Color và biến Clarity

2.3.2. Mối tương quan giữa các cặp biến số và biến phân loại

2.3.2.1. Depth Và Cut

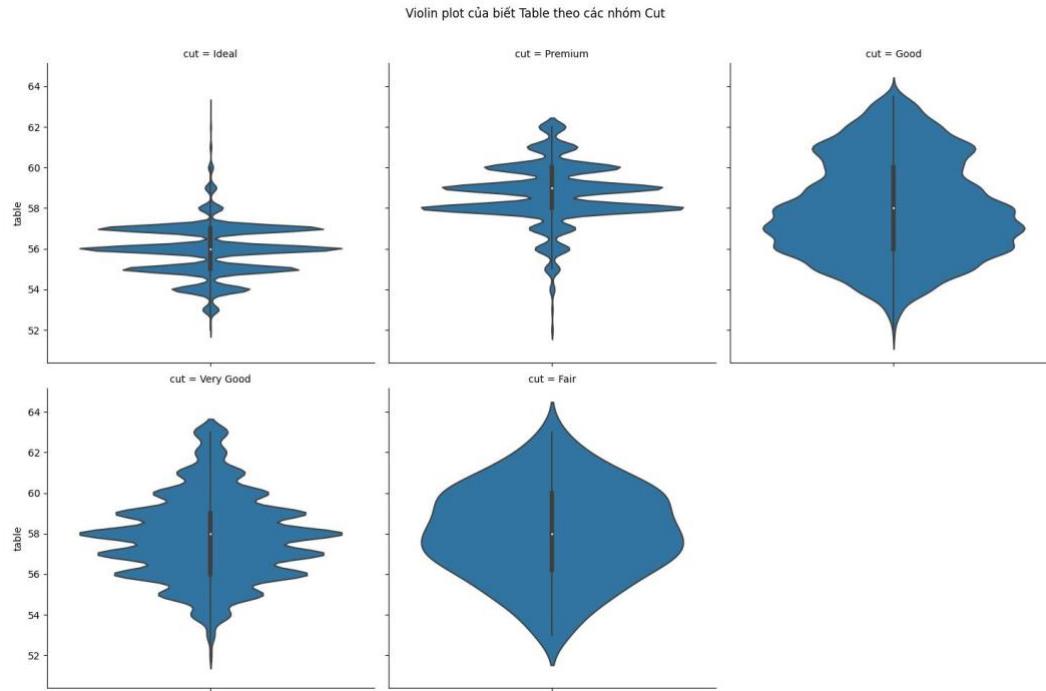
- Quan sát biểu đồ, ta có thể thấy, tỷ lệ chiều cao của kim cương so với chiều dài và chiều rộng của các nhóm vết cắt tập trung chủ yếu trong các đoạn dưới đây:
 - Ideal ∈ [61.7, 62]
 - Premium ∈ [61.5, 62.5]
 - Very Good ∈ [62.8, 63.2]
 - Good ∈ [63.4, 64]
 - Fair ∈ [64, 65]
- Có thể thấy, chất lượng vết cắt càng giảm thì tỷ lệ chiều cao của kim cương so với chiều dài và chiều rộng càng tăng.



Hình 5.24: Biểu đồ Violin thể hiện mối tương quan giữa biến Depth và Cut

2.3.2.2. Table và Cut

- Quan sát biểu đồ, ta có thể thấy, độ rộng của phần đỉnh kim cương so với trung bình chiều dài và chiều rộng của các nhóm vết cắt tập trung chủ yếu trong các đoạn dưới đây:
 Ideal $\in [55.8, 56] \cup [55.4, 55.6]$
 Premium: 58
 Very Good [57.8, 58]
 Good $\in [56, 58] \cup [60.8, 61.2]$
 Fair $\in [57, 60]$
- Có thể thấy, khi độ rộng của phần đỉnh kim cương so với trung bình chiều dài và chiều rộng tăng dần thì chất lượng của vết cắt cũng giảm dần.

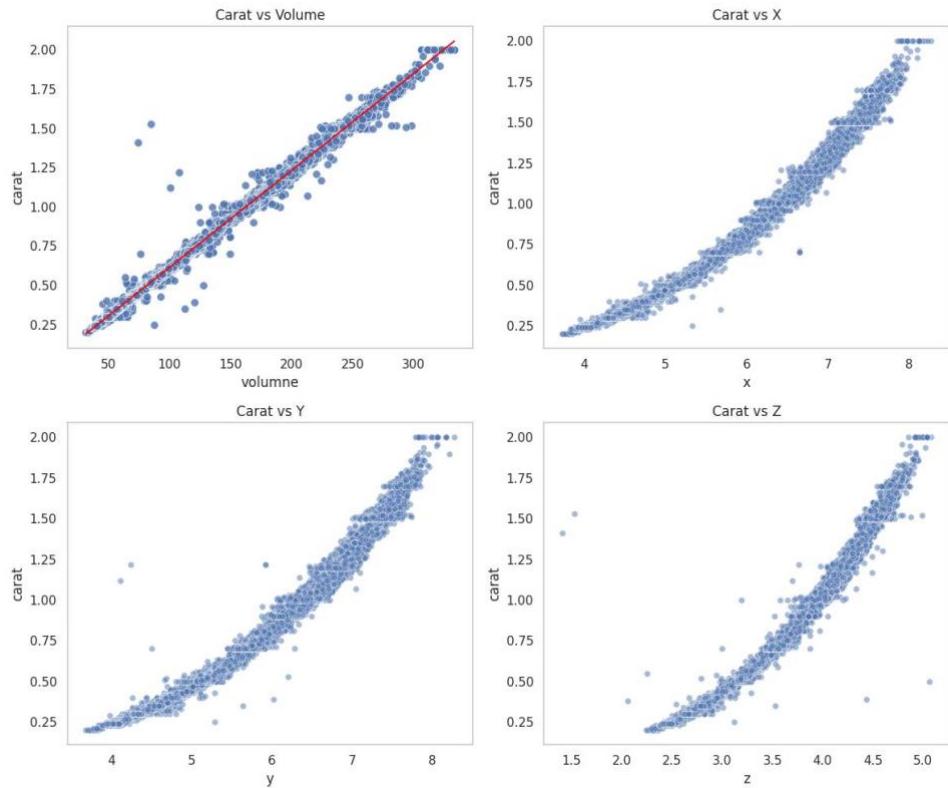


Hình 5.25: Biểu đồ Violin thể hiện mối tương quan giữa biến Table và Cut

2.3.3. Mối tương quan giữa các cặp biến số

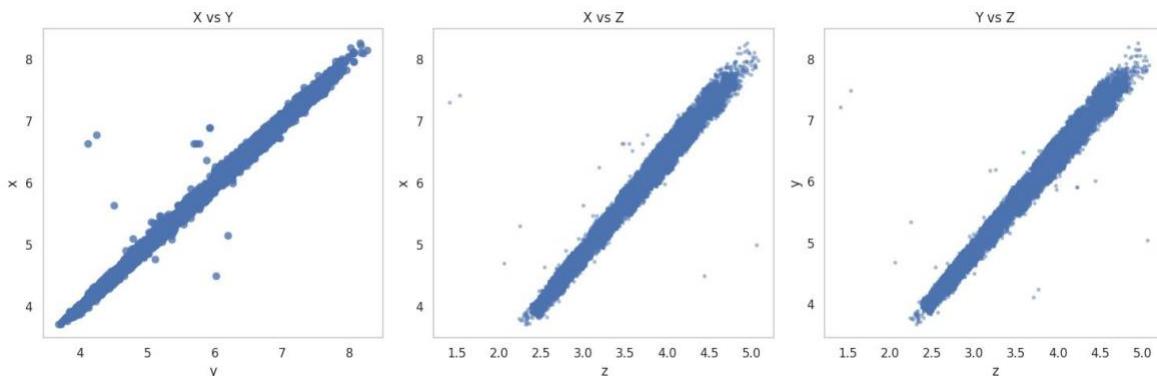
2.3.3.1. Các cặp biến số có độ tương quan cao với nhau

2.3.3.1.1. Carat và X, Y, Z



Hình 5.26: Regression plot thể hiện các cặp biến số có độ tương quan cao với nhau (Carat và X, Y, Z)

2.3.3.1.2. X và Y, Z

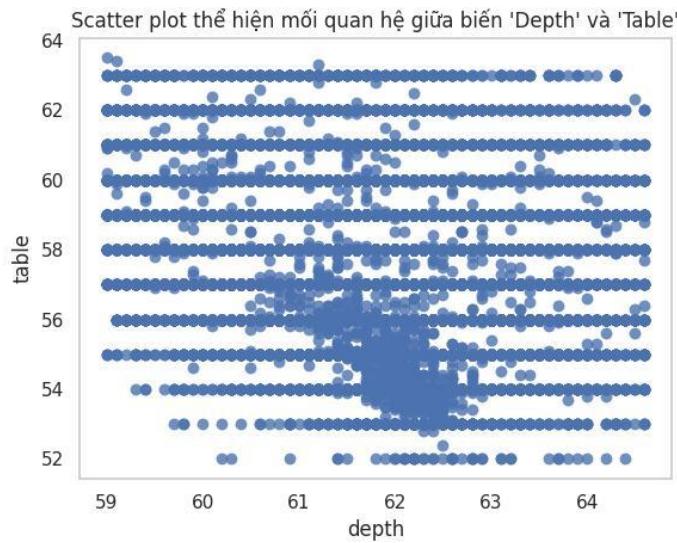


Hình 5.27: Regression plot thể hiện các cặp biến số có độ tương quan cao với nhau (X với Y và Z)

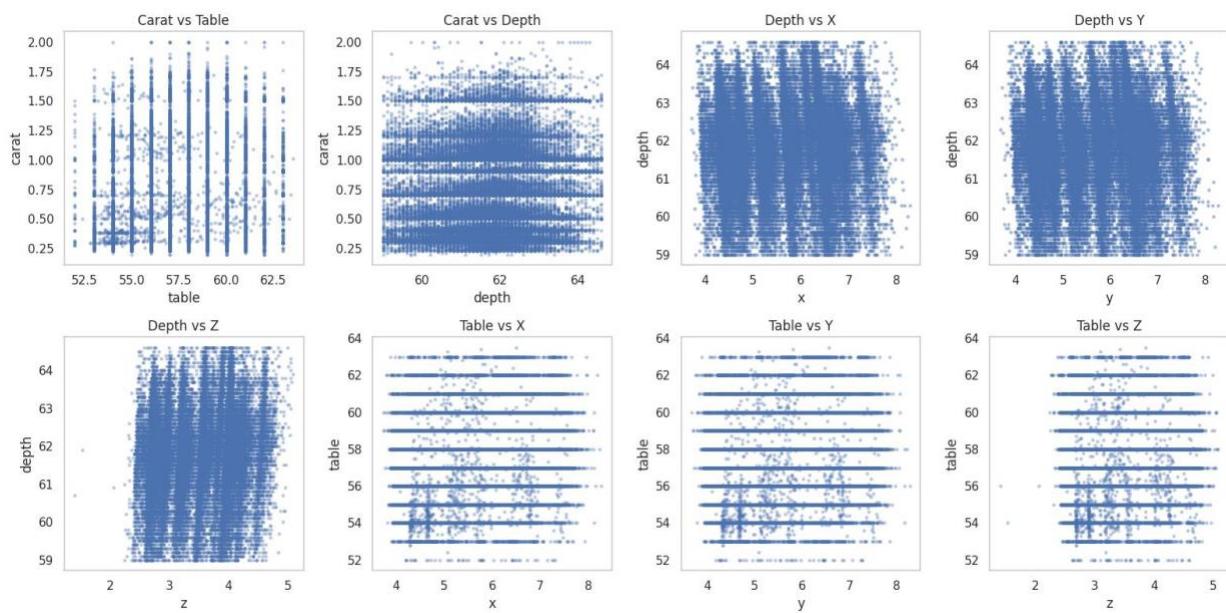
- Kết luận: Quan sát biểu đồ ta có thể thấy, các cặp biến {Carat, Volume}, {X,Y}, {X,Z}, {Y,Z} tạo thành một đường tuyến tính dương hoàn hảo. Điều này biểu thị một mối tương quan dương. Tức là khi giá trị của Volume tăng thì trọng lượng của kim cương cũng tăng. Tương tự như vậy với các cặp biến {X,Y}, {X,Z}, {Y,Z}.

2.3.3.2. Các cặp biến số có độ tương quan thấp hoặc không tương quan

- Quan sát biểu đồ ta có thể thấy, các điểm trong biểu đồ của các cặp biến phân bố ngẫu nhiên không theo quy luật, chứng tỏ giữa các cặp biến này không có mối liên hệ rõ ràng hay không có tương quan tuyến tính hoặc tương quan tuyến tính rất thấp.



Hình 5.28: Scatter plot thể hiện mối quan hệ giữa biến Depth và Table



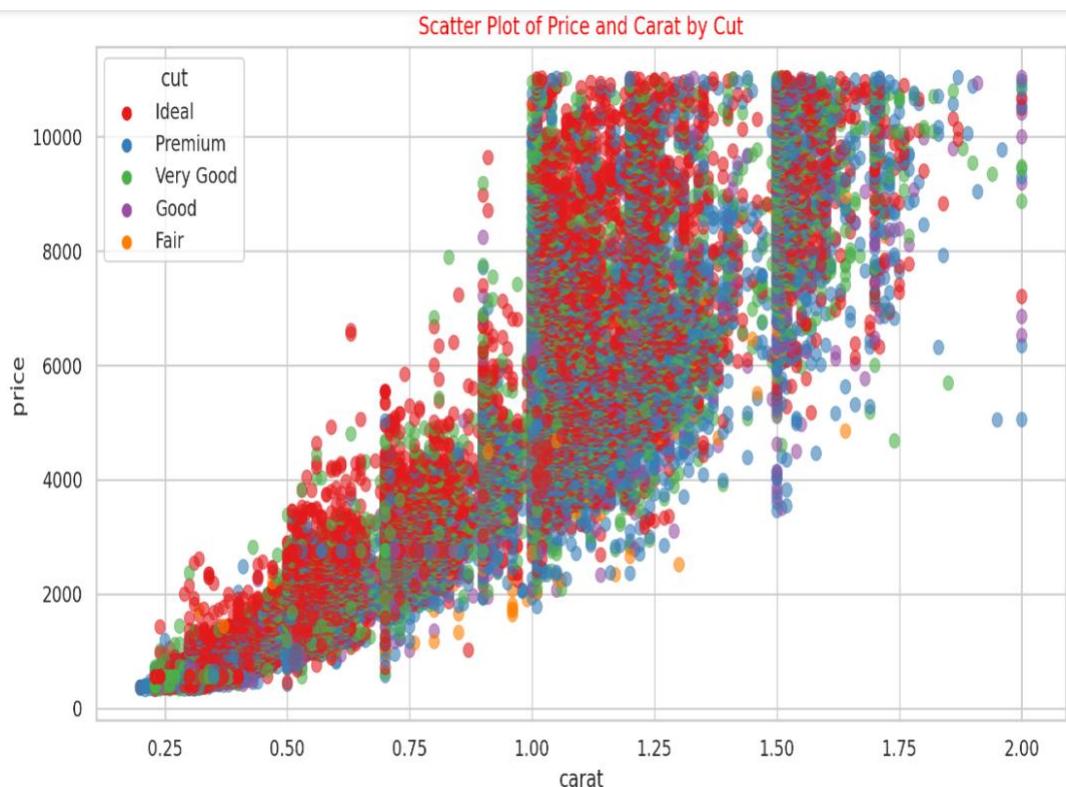
Hình 5.29: Regression plot thể hiện các cặp biến số có độ tương quan thấp hoặc không tương quan

3. Phân tích 3 biến

- Biểu diễn tương quan giữa các biến phân loại (cut, clarity, color) với 2 biến price và carat. Sử dụng scatter plot để thấy mối quan hệ của biến target ‘price’ và biến carat so với các

biến Categorical (cut, clarity, color) để xem các biến phân loại này có mối quan hệ như thế nào đến giá của kim cương (price) và trọng lượng của chúng (carat).

- Biểu đồ mối quan hệ giữa biến cut so với biến carat và price:



Hình 5.17: Scatterplot phân tích mối quan hệ giữa biến cut so với biến carat và price.

- Nhận xét:

- Chất lượng của vết cắt đóng vai trò quan trọng trong việc đánh giá giá trị của viên kim cương. Tuy nhiên, khi quan sát biểu đồ, chúng ta thấy rằng đa số các viên kim cương có chất lượng vết cắt ở mức Fair và Very Good thường xuất hiện ở khoảng giá dưới 6000 và trọng lượng từ 0.5 đến 1.75 carat. Trái ngược với đó, các viên kim cương có chất lượng vết cắt Ideal và Premium phân bố rộng rãi ở mọi phân khúc giá và trọng lượng từ 0.2 đến 1.5 carat.
- Nhìn chung, biểu đồ cho thấy rằng đa số người tiêu dùng không dành quá nhiều quan tâm đến chất lượng vết cắt. Họ có xu hướng chọn viên kim cương phù hợp với túi tiền mà không quá quan tâm đến chi tiết về chất lượng vết cắt. Chỉ một số ít người có độ nhạy bén và hiểu biết sâu sắc về kim cương mới có khả năng phân biệt được chất lượng vết cắt tốt, với sự chăm sóc tỉ mỉ và độ trong suốt ở mức đạt đến đỉnh cao. Những khách hàng này có thể được coi là những "outlier" trên biểu đồ, đại diện cho nhóm những người chọn lựa kim cương với tiêu chí cao cấp và trọng lượng lớn.

- Biểu đồ mối quan hệ giữa biến clarity so với biến carat và price:

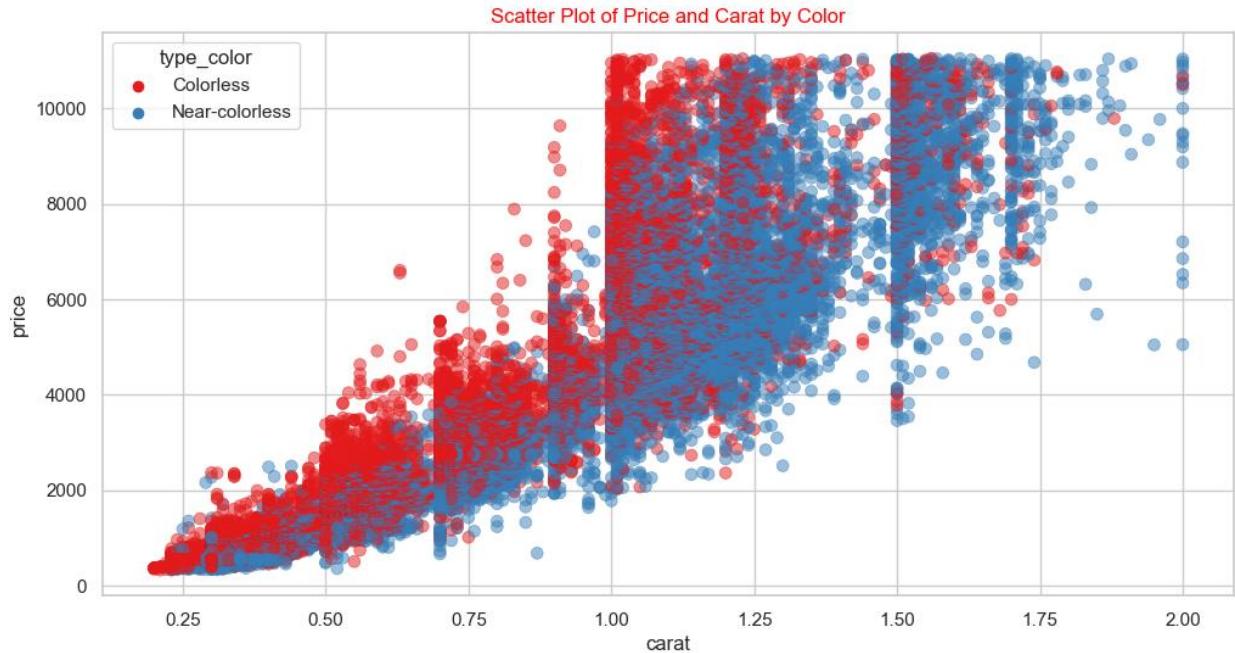


Hình 3.2: Scatterplot phân tích mối quan hệ giữa biến clarity so với biến carat và price.

- **Nhận xét:**

+ Qua việc phân tích biểu đồ, chúng ta có thể dễ dàng nhận thấy rằng kim cương có độ trong suốt kém, như I1, chiếm đa số ở các phân khúc giá bình dân từ 500 đến khoảng 6000 USD, và xuất hiện trong khoảng trọng lượng từ 0,5 đến 1,75 carat. Ngược lại, các viên kim cương có độ trong suốt cao như IF, VVS, VS tập trung ở mức giá cao hơn, nằm trong khoảng từ 6000 USD trở lên, với trọng lượng chủ yếu từ 0,2 đến 1,5 carat.

+ Sự quý giá của kim cương trong suốt thuần túy được thể hiện rõ, khi một viên kim cương nhỏ với độ trong suốt cao có giá có thể cao gấp nhiều lần so với các viên cùng trọng lượng nhưng độ trong suốt kém hơn. Điều này phản ánh xu hướng tiêu dùng, cho thấy đa số người tiêu dùng thích mua kim cương có độ trong suốt thấp vì giá thành rẻ, có thể mua được nhiều viên. Hơn nữa, khi đeo, ít người có thể phân biệt chất lượng trong suốt của kim cương, điều này cũng đóng góp vào quyết định mua kim cương với độ trong suốt thấp hơn.



Hình 3.3: Scatterplot phân tích mối quan hệ giữa biến color so với biến carat và price.

- **Nhận xét:**

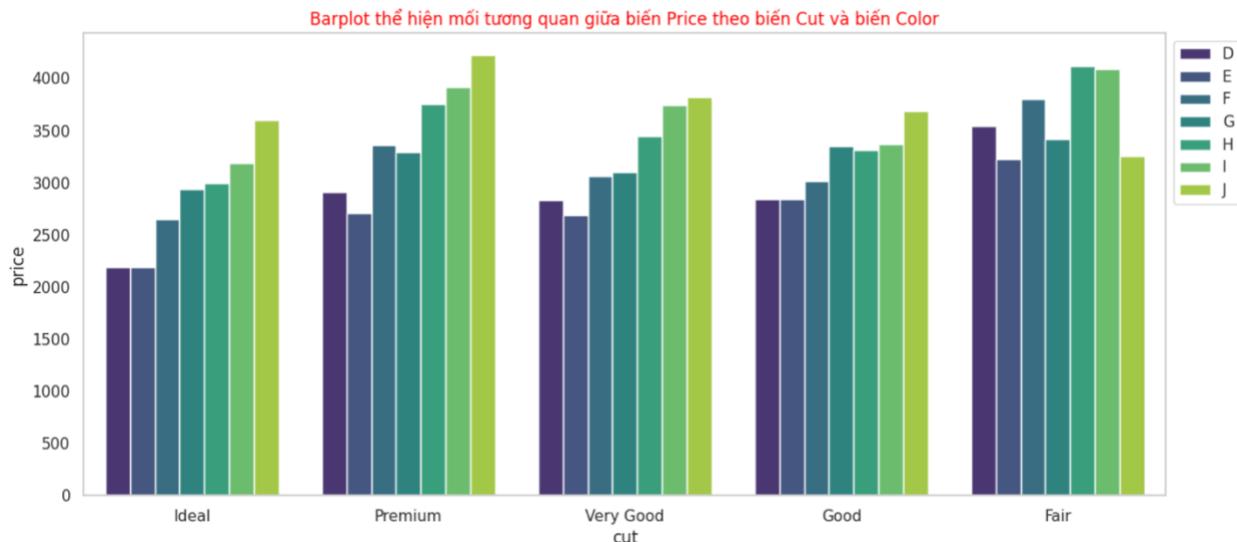
+ Quan sát biểu đồ, ta nhận thấy rằng kim cương màu Colorless (trong suốt) được coi là quý hiếm, dẫn đến việc trọng lượng của chúng trong mỗi giao dịch thường tập trung trong khoảng từ 0.3 đến 1.5 carat. Điều này cũng làm tăng giá trị của chúng theo từng carat (khối lượng). Ngược lại, các viên kim cương có màu sắc khác (Near-colorless), mặc dù không quý hiếm nhưng lại có sắc lấp lánh và thường có trọng lượng lớn hơn.

+ Mặc dù không phải là kim cương quý hiếm màu Colorless, các viên kim cương khác vẫn thu hút sự chú ý do sự độc đáo của màu sắc và trọng lượng lớn. Do đó, giá của chúng cũng tăng theo trọng lượng và sự độc đáo của màu sắc. Một điều cần lưu ý là nhiều người có thể hiểu lầm rằng chỉ những viên kim cương có màu sắc đặc biệt mới có giá trị cao.

Tuy nhiên, so sánh giữa một viên kim cương màu D và một viên màu J có giá trên 10000, nhưng trọng lượng của chúng lại chênh lệch đáng kể (D: 1 carat, J: 1.75 carat), cho thấy sự quý hiếm không chỉ đến từ màu sắc mà còn từ trọng lượng đáng kể.

- Sử dụng bar plot để thấy mối quan hệ của biến target price so với các biến Categorical (cut, clarity, color) để xem các biến phân loại này có mối quan hệ như thế nào đến giá của kim cương (price).

- Biểu đồ mối quan hệ giữa 2 biến cut và color so với biến price:



Hình 3.4: Barplot phân tích mối quan hệ giữa biến Price với Cut và Color.

- **Nhận xét:**

Từ biểu đồ trên, ta có thể thấy:

- + Các viên kim cương có cùng một kiểu vết cắt trong năm kiểu (Ideal, Premium, Good, Very Good, Fair) thì viên kim cương nào có màu sắc là các màu G, H, I, J (Near-colorless) thì sẽ có mức giá cao hơn so với các viên kim cương có các màu D, E, F (Colorless)
- + Từ đó, có thể thấy giá của viên cương không chỉ phụ thuộc vào chất lượng của vết cắt mà còn phụ thuộc vào màu sắc mà nó có.

Chương 6: Mô hình ML

1. Định nghĩa:

Mô hình k-NN (k-Nearest Neighbors) là một mô hình học máy thuộc lớp học máy có giám sát (supervised-learning), được sử dụng chủ yếu trong 2 bài toán hồi quy và phân loại. Mô hình này dựa trên nguyên tắc rằng những điểm dữ liệu gần nhau trong không gian đặc trưng có xu hướng tương tự nhau. Mô hình tìm k láng giềng gần nhất và sử dụng giá trị trung bình (đối với hồi quy) hoặc số lớp (đối với phân loại) để dự đoán giá trị mới.

2. Ưu nhược điểm của mô hình:

2.1. Ưu điểm:

- Đơn giản và dễ giải thích.
- Hoạt động tốt trong trường hợp phân loại với nhiều lớp .
- Sử dụng được trong cả phân loại và hồi quy.

2.2. Nhược điểm:

- Bởi vì mô hình cần lưu trữ tất cả các điểm dữ liệu nên sẽ trở nên rất chậm khi số lượng điểm dữ liệu tăng lên.
- Tốn bộ nhớ.
- Nhạy cảm với các dữ liệu bất thường (nhiều).

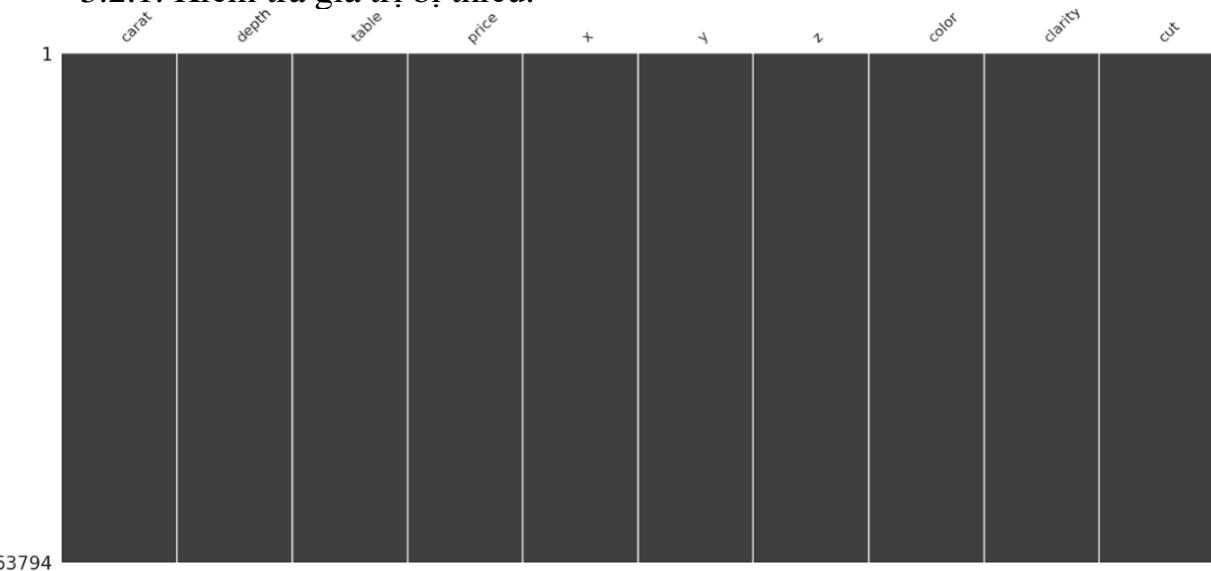
3. Xây dựng mô hình:

3.1. Import thư viện:

```
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import r2_score
from sklearn.metrics import accuracy_score, mean_squared_error,
mean_absolute_error
from sklearn import preprocessing, metrics
from math import sqrt
from plotly.offline import plot
from IPython.display import HTML
import plotly.graph_objs as go
import plotly.io as pio
import os
import warnings
warnings.filterwarnings('ignore')
```

3.2. Tiền xử lý dữ liệu cho mô hình:

3.2.1. Kiểm tra giá trị bị thiếu.



Hình 6.1: Biểu đồ heatmap thể hiện dữ liệu bị thiếu.

3.2.2. Gom 3 biến x, y, z .

```
processed_df['volume'] = processed_df['x']*processed_df['y']*processed_df['z']
```

Qua quá trình trực quan hóa dữ liệu trước đó, thực hiện gom nhóm 3 biến x, y, z nhằm hạn chế dữ liệu x, y, z bị đa cộng tuyến khi xây dựng mô hình dẫn đến kết quả không chính xác.

3.2.3. Chuyển đổi các biến categorical.

Trong quá trình tiền xử lý dữ liệu, kiểm tra và xử lý các biến phân loại để chuẩn bị cho quá trình xây dựng mô hình dự đoán giá kim cương. Đầu tiên, cần xác định danh sách các biến phân loại bằng cách sử dụng vòng lặp để kiểm tra kiểu dữ liệu của từng cột trong tập dữ liệu. **Các biến 'color', 'clarity', và 'cut' được xác định là các biến phân loại.**

```
categorical_col = list()
for col in processed_df.dtypes.index:
    if processed_df.dtypes[col] == 'object':
        categorical_col.append(col)
```

Tiếp theo, để có thể sử dụng thông tin từ các biến này trong mô hình, chúng ta áp dụng kỹ thuật mã hóa. **Đối với 'color', 'clarity' và 'cut'** có nhiều loại khác nhau, sử dụng kỹ thuật **one-hot encoding** để chuyển đổi chúng thành các biến nhị phân (0 và 1) để có thể sử dụng trong mô hình.

```
# One-hot encoding cho 'color' and 'clarity'
```

```
processed_df = pd.get_dummies(processed_df, columns=['color', 'clarity', 'cut'], drop_first=True)
```

Sau quá trình tiền xử lý này, tập dữ liệu đã được chuẩn bị sẵn sàng cho việc xây dựng mô hình, và các biến phân loại đã được chuyển đổi thành dạng số để có thể được sử dụng hiệu quả trong quá trình huấn luyện mô hình dự đoán giá kim cương.

3.3. Cài đặt thuật toán:

3.3.1. Tách tập dữ liệu:

Trong quá trình phát triển mô hình, bước quan trọng là tách tập dữ liệu thành tập train và tập test để đánh giá hiệu suất của mô hình trước khi đưa vào dữ liệu mới. Thực hiện loại bỏ các cột không cần thiết như các biến không tương quan hoặc ít tương quan được xác định thông qua quá trình phân tích và trực quan hóa (**depth**, **table**) và loại bỏ biến **price** mục tiêu ở tập train để tránh làm cho dữ liệu không chính xác. Sau đó, sử dụng **train_test_split** để chia tập dữ liệu thành tập huấn luyện và tập kiểm thử với tỷ lệ **70-30**. Điều này giúp mô hình được đánh giá trên dữ liệu độc lập và có khả năng tổng quát hóa tốt hơn.

```
x, y = processed_df.drop(["price", "table", "depth"], axis=1), processed_df.price  
xtrain , xtest , ytrain , ytest = train_test_split(x , y , test_size =0.3 , random_state = 42)
```

3.3.2. Xác định k láng giềng bằng phương pháp cross_val_score:

Trước khi xây dựng mô hình k-NN, cần xác định giá trị tối ưu cho tham số k - số lượng láng giềng. Quá trình này thường được thực hiện thông qua kiểm định chéo bằng phương pháp cross_val_score để đánh giá hiệu suất của mô hình với các giá trị k khác nhau.

```
neighbors = np.arange(1, 20, 2)  
scores = []  
for k in neighbors:  
    clf = KNeighborsRegressor(n_neighbors = k, weights = 'distance', p=1)  
    clf.fit(xtrain, ytrain)  
    score = cross_val_score(clf, xtrain, ytrain, cv = 10)  
    scores.append(score.mean())  
mse = [1-x for x in scores]
```

3.3.3. Xem xét lỗi kiểm định chéo và lựa chọn giá trị k phù hợp:

"CV error" thường được hiểu là "cross-validation error" (lỗi kiểm định chéo). Vẽ đồ thị để kiểm tra lỗi kiểm định chéo (cross-validation error) là một phương pháp đánh giá hiệu suất của mô hình bằng cách biểu diễn sự biến thiên của lỗi này theo giá trị K, và giá trị tối ưu của K được in ra màn hình.

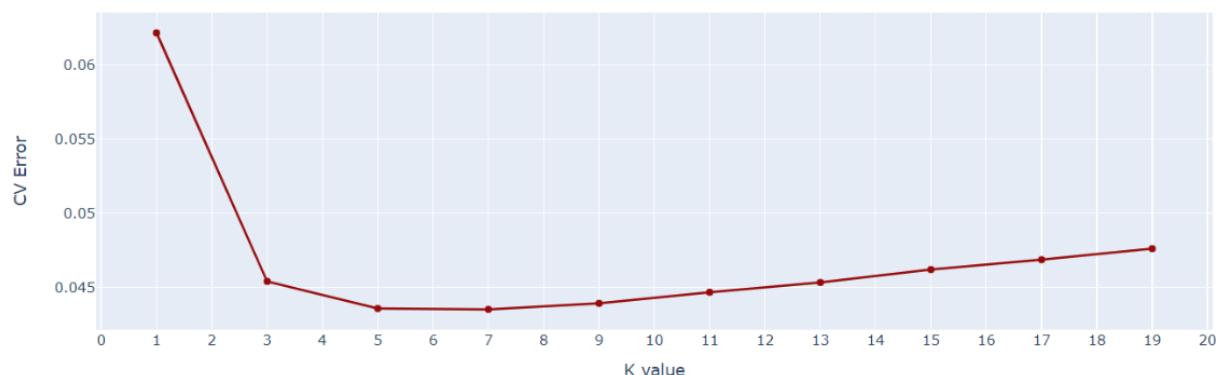
Lợi ích chính của việc sử dụng lỗi kiểm định chéo: Lỗi kiểm định chéo cung cấp một ước lượng tổng quát về hiệu suất của mô hình đồng thời giúp phát hiện và kiểm soát tình trạng overfitting. Ngoài ra, việc sử dụng lỗi kiểm định chéo còn để tối ưu hóa các tham số của mô hình và đảm bảo khả năng tổng quát tốt nhất.

```

trace0 = go.Scatter(
    y=mse,
    x=neighbors,
    mode='lines+markers',
    marker=dict(
        color='rgb(150, 10, 10)')
)
layout = go.Layout(
    title="",
    xaxis=dict(
        title='K value',
        tickmode='linear'
    ),
    yaxis=dict(
        title='Cross-Validation Error',
    )
)
fig = go.Figure(data=[trace0], layout=layout)
plot(fig, filename='basic-line')

html_path = '/content/basic-line.html'
HTML(filename=html_path)

```



Hình 6.2: Biểu đồ thể hiện lỗi kiểm định chéo để tìm k láng giềng

```
optimal_k = neighbors[mse.index(min(mse))]
```

```
print("Optimal K: ", optimal_k)
```

Optimal K: 9

Từ đồ thị, ta có thể thấy giá trị K = 9 là giá trị có ít lỗi kiểm định chéo nhất được xem là giá trị tham số tối ưu phù hợp cho việc xây dựng mô hình.

```
knn = KNeighborsRegressor(n_neighbors=optimal_k)  
knn
```

```
▼ KNeighborsRegressor  
KNeighborsRegressor(n_neighbors=9)
```

Hình 6.3: Hình ảnh về mô hình K-NN với giá trị k = 9

4. Đánh giá mô hình:

4.1. Đánh giá mô hình bằng các chỉ số:

Mô hình KNN được đánh giá dựa trên các chỉ số như Train_score, Test_score, Mean Squared Error (MSE), Mean Absolute Error (MAE) và Root Mean Squared Error (RMSE). Các chỉ số này cung cấp cái nhìn tổng quan về khả năng giải thích và độ chính xác của mô hình.

```
knn = KNeighborsRegressor(n_neighbors=optimal_k)  
knn  
ypred = knn.fit(xtrain , ytrain).predict(xtest)  
  
train_score = knn.score(xtrain , ytrain)  
test_score = knn.score(xtest , ytest)  
  
mse = mean_squared_error(ytest , ypred)  
mae = mean_absolute_error(ytest , ypred)  
rms = sqrt(mean_squared_error(ytest, ypred))  
evaluation_results = pd.DataFrame(  
    { "KNN": [train_score, test_score, mse, mae, rms]},  
    index=["train_score", "test_score", "MSE", "MEA", "RMSE"])  
evaluation_results
```

KNN

train_score	0.9526
test_score	0.9397
MSE	408515.3904
MEA	352.0048
RMSE	639.1520

Bảng 6.1: Bảng kết quả đánh giá mô hình k-NN qua các chỉ số

4.1.1. Train_score: 95.26%

Mô hình KNN đã đạt được độ chính xác cao lên đến 95.26% trên tập dữ liệu huấn luyện. Điều này cho thấy mô hình có khả năng khá tốt trong việc mô phỏng và từ dữ liệu huấn luyện.

4.1.2. Test_score: 93.97%

Trên tập dữ liệu kiểm thử, mô hình KNN vẫn giữ được độ chính xác cao với giá trị là 93.97%. Điều này cho thấy mô hình không bị overfitting từ tập dữ liệu huấn luyện.

4.1.3. Mean Squared Error (MSE): 408515.3904

MSE là một chỉ số đo lường sự chênh lệch giữa giá trị dự đoán và giá trị thực tế. Trong trường hợp này, MSE đạt giá trị là 408515.3904, cho thấy mức độ chênh này vẫn ở mức chấp nhận được.

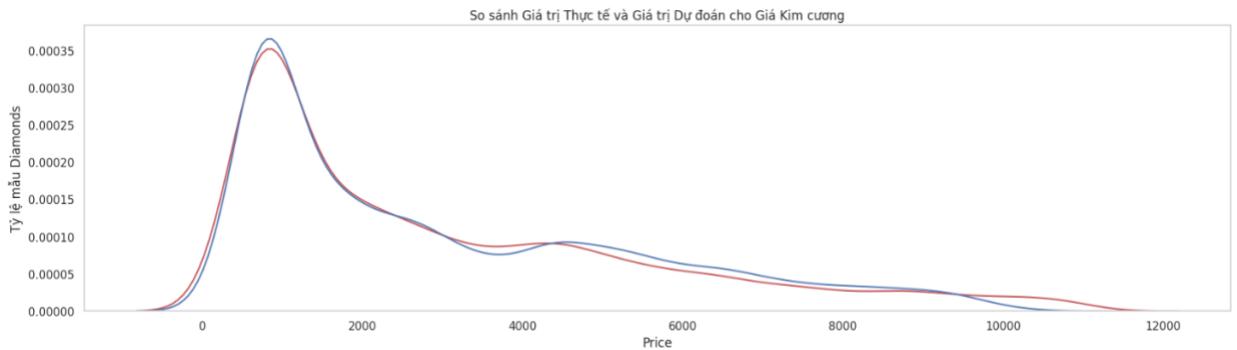
4.1.4. Mean Absolute Error (MAE): 352.0048

MAE là trung bình độ lớn tuyệt đối của sai số giữa giá trị dự đoán và giá trị thực tế. Mô hình KNN có MAE là 352.0048, đặc điểm này chỉ ra rằng mô hình có khả năng dự đoán giá kim cương với độ chính xác cao.

4.1.5. Root Mean Squared Error (RMSE): 639.1520

RMSE là căn bậc hai của MSE, đo lường độ lớn của sai số giữa giá trị dự đoán và giá trị thực tế.

4.2. Biểu đồ thể hiện giá trị dự đoán so với giá trị ban đầu



Hình 6.4: Biểu đồ so sánh giá trị dự đoán so với giá trị ban đầu

Chương 7: Kết luận

Qua quá trình phân tích, kiểm định và biểu diễn cho thấy những yếu tố ảnh hưởng đến giá kim cương. Kết quả phân tích trên giúp nhóm xác định được tầm quan trọng của các yếu tố, từ đó loại bỏ những biến không cần thiết hoặc gây nhiễu, nâng cao hiệu quả xây dựng mô hình máy học. Mô hình KNN do nhóm xây dựng có độ chính xác 0.95 trong dự đoán giá kim cương. Kết quả này cho thấy hiệu quả của việc lựa chọn biến dựa trên kết quả phân tích, góp phần cải thiện độ chính xác của mô hình.

Tài liệu tham khảo

1. <https://www.machinelearningplus.com/machine-learning/mice-imputation/>
2. <https://www.kaggle.com/code/ahmedgadoo/diamond-more-and-more-eda-visualizations#install-Libraries>
3. <https://www.kaggle.com/code/ranjeet013/predicting-diamond-price-acc-0-99>
4. <https://www.kaggle.com/code/surajjha101/regression-models-diamond-price-prediction>
5. <https://www.kaggle.com/code/joysonprincealvares/eda-4-diamond-sales-analysis>
6. <https://www.kaggle.com/code/emrekorkmazpy/prediction-diamonds-r-score-0-98>
7. <https://www.kaggle.com/code/daisyamber/diamond-prices-log-log-linear-regression>
8. <https://www.kaggle.com/code/deborshibanerjee/diamond-price-prediction-0-98>

Bảng phân công

Thành viên	Phân công	Đánh giá
Đinh Trọng Hữu	Tiền xử lý dữ liệu, Phân tích đa biến (2 biến), Mô hình k-NN, Slide	100%
Phạm Minh Hiền	Mô tả dữ liệu, Các thuộc tính bộ dữ liệu, phân tích đa biến (1, 2 biến), kiểm định giả thuyết (Chi bình phương)	100%
Nguyễn Như Hoàng	Mô tả bộ dữ liệu, Phân tích đa biến (3 biến), Anova (1 chiều)	100%
Đặng Nhật Huy	Tổng hợp word, Tổng quan đề tài, Xử lý các outliers, Anova (2 chiều)	100%