**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN: KHAI PHÁ DỮ LIỆU**

**MÔ HÌNH DỰ ĐOÁN GIÁ KIM CƯƠNG**

**Nhóm thực hiện: Nhóm 5**

**Thành viên nhóm:**

1. Phạm Tiến Thành - 2251262645
2. Mai Hoàng Tùng - 2251262656
3. Chu Đức Hoàng - 2251262604
4. Triệu Vũ Minh Quân - 2251262629

**Giảng viên hướng dẫn:** Trần Mạnh Tuấn

Hà Nội, 2024

**MỤC LỤC**

LỜI CẢM ƠN

I. Mô tả bài toán..............................................................................................5

1. Lý do chọn đề tài.......................................................................................5

2. Tổng quan bài toán....................................................................................5

3. Quy trình thực hiện....................................................................................6

II. Quy trình khai phá dữ liệu.........................................................................8

1. Đọc dữ liệu................................................................................................8

2. Tiền xử lí dữ liệu .....................................................................................10

2.1. Làm sạch dữ liệu...................................................................................10

2.2. Phân tích dữ liệu thăm dò.....................................................................12

III. Trực quan hóa dữ liệu..............................................................................13

1. Biểu đồ pairplot.....................................................................................13

2. Biểu đồ subplots....................................................................................14

IV. Các biến phân loại...................................................................................16

V. Biến đổi dữ liệu (chuẩn hóa dữ liệu).........................................................17

VI. Đào tạo mô hình......................................................................................20

1. Thuật toán hồi quy tuyến tính................................................................20

1.1. Lý thuyết...........................................................................................20

1.2. Quy trình thực hiện...........................................................................21

1.3. Kết quả thu được...............................................................................22

1.4. Test mô hình.....................................................................................22

1.5. Tạo cửa sổ dự đoán...........................................................................25

2. Thuật toán Random Forest.....................................................................29

2.1. Lý thuyết...........................................................................................29

2.2. Quy trình thực hiện...........................................................................30

2.3. Kết quả thu được...............................................................................31

2.4. Test mô hình.....................................................................................34

2.5. Tạo cửa sổ dự đoán...........................................................................36

VII. So sánh 2 mô hình (hồi quy tuyến tính và Random Forest)........................38

KẾT LUẬN.........................................................................................................40

**LỜI CẢM ƠN**

Chúng em đang sống trong thời kỳ mà công nghệ thông tin phát triển không ngừng, với những ứng dụng vượt bậc trong mọi lĩnh vực từ doanh nghiệp, giáo dục đến đời sống hàng ngày. Việc áp dụng các giải pháp tin học không chỉ giúp tối ưu hóa quy trình làm việc mà còn mở ra nhiều cơ hội sáng tạo mới mẻ, làm cho cuộc sống trở nên tiện lợi và hiệu quả hơn. Để đáp ứng được yêu cầu ngày càng cao của xã hội và thị trường lao động, việc trang bị những kiến thức chuyên môn vững vàng là điều vô cùng cần thiết đối với mỗi sinh viên. Với lý do đó, chúng em quyết định chọn đề tài "Mô hình dự đoán giá kim cương" làm báo cáo cho môn học của mình.

Chúng em xin gửi lời cảm ơn sâu sắc tới thầy Trần Mạnh Tuấn, người đã không ngừng hỗ trợ và truyền đạt kiến thức môn Khai phá dữ liệu trong suốt quá trình học. Sự hướng dẫn tận tâm của thầy không chỉ giúp chúng em nắm bắt các kiến thức cơ bản mà còn thúc đẩy niềm đam mê tìm tòi, học hỏi về công nghệ thông tin.

Dù đã nỗ lực hết mình trong việc thực hiện đề tài, nhưng với kinh nghiệm còn hạn chế và thời gian thực hiện ngắn, không thể tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý, nhận xét từ thầy cô để hoàn thiện hơn. Những đóng góp quý báu của thầy cô sẽ là hành trang quý giá cho chúng em trên con đường học tập và sự nghiệp sau này.

Chúng em xin chân thành cảm ơn!

\* Phân chia công việc:

|  |  |
| --- | --- |
| Tên thành viên | Công việc |
| Triệu Vũ Minh Quân | * Khai phá dữ liệu |
| Chu Đức Hoàng | * Trực quan hóa * Phân loại biến * Chuẩn hóa dữ liệu |
| Phạm Tiến Thành | * Đào tạo mô hình bằng hồi quy tuyến tính |
| Mai Hoàng Tùng | * Đào tạo mô hình bằng Random Forest |

1. **Mô tả bài toán**

Bài toán dự báo giá kim cương là một bài toán thuộc lĩnh vực khai phá dữ liệu và học máy. Mục tiêu của bài toán là xây dựng mô hình dự đoán giá trị của một viên kim cương dựa trên các đặc trưng như trọng lượng (carat), độ cắt (cut), màu sắc (color), độ trong suốt (clarity), và các thuộc tính vật lý khác như kích thước và hình dạng. Dữ liệu về kim cương thường bao gồm nhiều mẫu với giá trị giá cả đã biết, giúp mô hình học hỏi và đưa ra dự đoán cho các viên kim cương mới chưa biết giá. Bài toán này có ý nghĩa thực tiễn trong việc định giá kim cương trên thị trường, hỗ trợ các nhà buôn và người tiêu dùng đánh giá chính xác giá trị viên đá quý dựa trên các yếu tố vật lý của nó.

1. **Lí do chọn đề tài**

Lý do chọn đề tài "Dự báo giá kim cương" xuất phát từ nhu cầu thực tiễn trong thị trường đá quý, nơi giá trị của một viên kim cương phụ thuộc vào nhiều yếu tố phức tạp như trọng lượng, màu sắc, độ trong và độ cắt. Việc dự báo giá chính xác giúp người mua và người bán có cơ sở đánh giá hợp lý hơn, tránh những sai lệch trong quá trình giao dịch. Ngoài ra, với sự phát triển của công nghệ khai phá dữ liệu và học máy, việc áp dụng các mô hình dự báo giá kim cương cũng mở ra tiềm năng phát triển các hệ thống tự động, tăng hiệu quả kinh doanh và giảm thiểu sai sót trong việc định giá.

1. **Tổng quan về bài toán**

Tổng quan về bài toán dự báo giá kim cương tập trung vào việc dự đoán giá trị của một viên kim cương dựa trên các thuộc tính vật lý và chất lượng của nó. Các thuộc tính chính bao gồm trọng lượng (carat), độ cắt (cut), màu sắc (color), độ trong (clarity), và các yếu tố vật lý như kích thước (x, y, z). Mục tiêu là xây dựng mô hình dự báo giá dựa trên dữ liệu đã cho, giúp đưa ra giá trị gần đúng cho các viên kim cương mới.

Dữ liệu đầu vào bao gồm nhiều mẫu kim cương với các thông tin chi tiết về các thuộc tính trên và giá bán tương ứng. Các mô hình khai phá dữ liệu và học máy như hồi quy tuyến tính, cây quyết định, hoặc các thuật toán khác có thể được áp dụng để học từ dữ liệu và đưa ra dự báo.

Bài toán này mang ý nghĩa thực tiễn cao trong thương mại, giúp các bên liên quan dự đoán và định giá kim cương một cách chính xác và khách quan hơn. Việc áp dụng các thuật toán học máy không chỉ giúp cải thiện độ chính xác của việc định giá mà còn tạo ra sự minh bạch và tin cậy trong giao dịch.

1. **Quy trình thực hiện**

### Quy trình thực hiện bài toán dự báo giá kim cương bao gồm các bước chính sau:

#### **3.1. Thu thập dữ liệu**

* Dữ liệu được thu thập từ các nguồn công khai hoặc từ các tập dữ liệu đã có sẵn, như bộ dữ liệu kim cương trên Kaggle hoặc các nguồn thương mại.
* Bộ dữ liệu điển hình sẽ bao gồm nhiều thông tin về từng viên kim cương như trọng lượng (carat), độ cắt (cut), màu sắc (color), độ trong (clarity), giá cả (price) và các kích thước vật lý (x, y, z).

#### **3.2. Tiền xử lý dữ liệu**

* **Xử lý ngoại lệ (outliers)**: Loại bỏ các giá trị bất thường có thể làm sai lệch mô hình, ví dụ như viên kim cương có trọng lượng quá lớn hay quá nhỏ.
* **Chuẩn hóa dữ liệu**: Chuẩn hóa các thuộc tính có giá trị khác nhau để đảm bảo dữ liệu đầu vào cho mô hình có cùng đơn vị hoặc thang đo.
* **Mã hóa dữ liệu phân loại**: Các thuộc tính phân loại như "cut", "color", và "clarity" thường là dạng văn bản. Chúng cần được chuyển thành dạng số bằng các kỹ thuật như One-hot encoding hoặc Label encoding để sử dụng trong mô hình học máy.

#### **3.3. Phân tích dữ liệu**

* **Phân tích thống kê mô tả**: Thực hiện các thống kê mô tả để hiểu rõ hơn về dữ liệu, ví dụ như phân phối của thuộc tính "price", "carat", hay "cut".
* **Trực quan hóa dữ liệu**: Sử dụng biểu đồ để hiển thị mối quan hệ giữa các thuộc tính như biểu đồ phân tán (scatter plot) giữa trọng lượng (carat) và giá (price), hoặc biểu đồ hộp (boxplot) để hiểu thêm về sự biến thiên của từng thuộc tính.
* **Tương quan giữa các thuộc tính**: Sử dụng ma trận tương quan để phân tích mối quan hệ giữa các thuộc tính và xác định những yếu tố nào ảnh hưởng nhiều nhất đến giá của kim cương.

#### **3.4. Xây dựng mô hình dự báo**

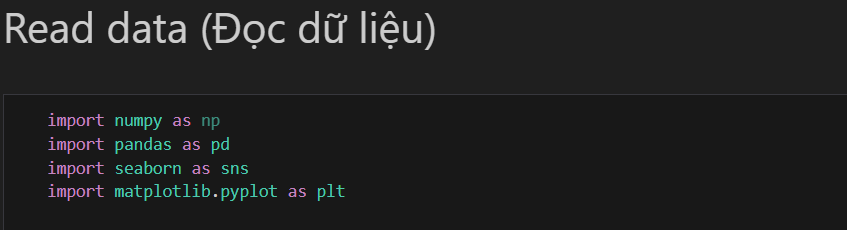
* **Chọn thuật toán học máy**: Dựa vào bản chất của bài toán dự báo giá để chọn ra thuật toán phù hợp và tối ưu.
* **Huấn luyện mô hình**: Sử dụng tập dữ liệu huấn luyện để huấn luyện mô hình. Chia dữ liệu thành hai phần: tập huấn luyện (training set) và tập kiểm thử (test set) để đánh giá mô hình.
* **Tinh chỉnh tham số**: Tinh chỉnh các siêu tham số (hyperparameters) của mô hình bằng cách sử dụng phương pháp tìm kiếm lưới (Grid Search) hoặc tìm kiếm ngẫu nhiên (Random Search) để đạt được kết quả tốt nhất.

#### **3.5. Đánh giá mô hình**

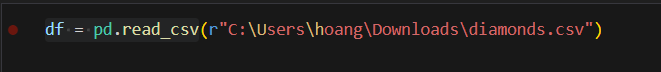
* **Sử dụng các chỉ số đánh giá**:
  + **RMSE (Root Mean Squared Error)**: Đo lường sai số trung bình giữa giá trị dự đoán và giá trị thực tế.
  + **MAE (Mean Absolute Error)**: Tính sai số trung bình giữa các giá trị dự đoán và giá trị thực.
  + **R² (Hệ số xác định)**: Đánh giá mức độ mô hình phù hợp với dữ liệu thực.
* **Kiểm tra trên tập kiểm thử**: Sau khi mô hình được huấn luyện, kiểm tra hiệu suất của mô hình trên tập kiểm thử để đảm bảo mô hình không bị overfitting (quá khớp với dữ liệu huấn luyện).

1. **Quy trình khai phá dữ liệu**
2. **Đọc dữ liệu**

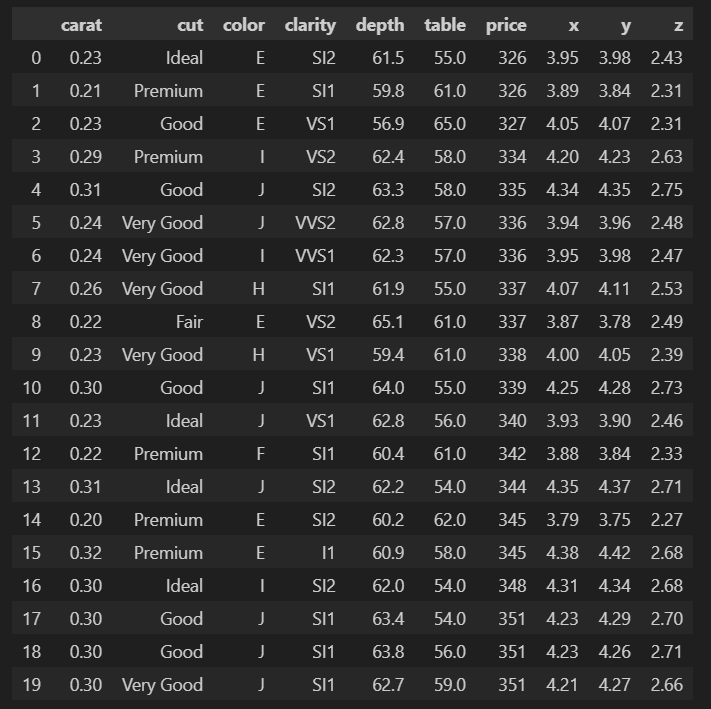
* Triển khai các thư viện sử dụng trong quá trình phân tích dữ liệu, từ việc xử lý và quản lý dữ liệu (với pandas) đến việc phân tích (với NumPy) và trực quan hóa dữ liệu (với Seaborn và Matplotlib).



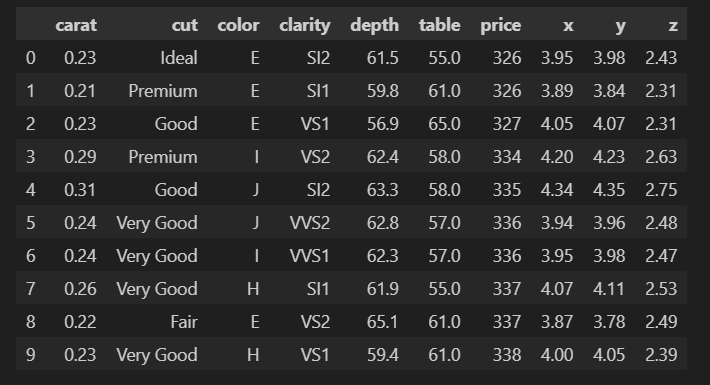
* df = pd.read\_csv: Đọc dữ liệu từ file data định dạng ‘.csv’



* df.head(20) : Gọi ra 20 dòng dầu tiên trong bảng dữ liệu

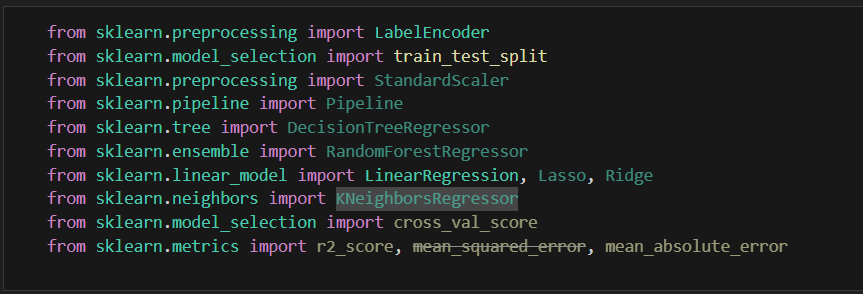


* df1 = df.drop(columns = 'X'): Xóa cột có tên là ‘X’ khỏi bảng dữ liệu. Trong bài toán này xuất hiện cột ‘Unnamed: 0’ không cần thiết cho bài toán nên cần thao tác xóa bỏ và xuất ra bảng mới sau khi xóa

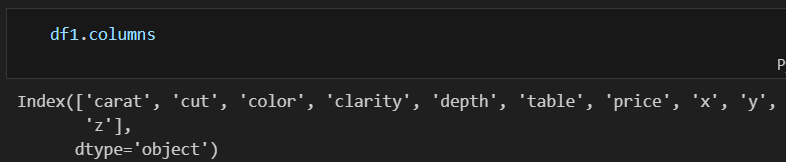


1. **Tiền xử lý dữ liệu**

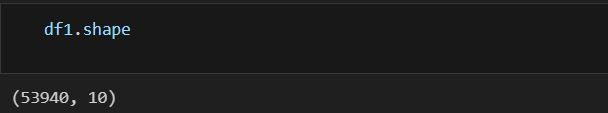
**2.1. Làm sạch dữ liệu**



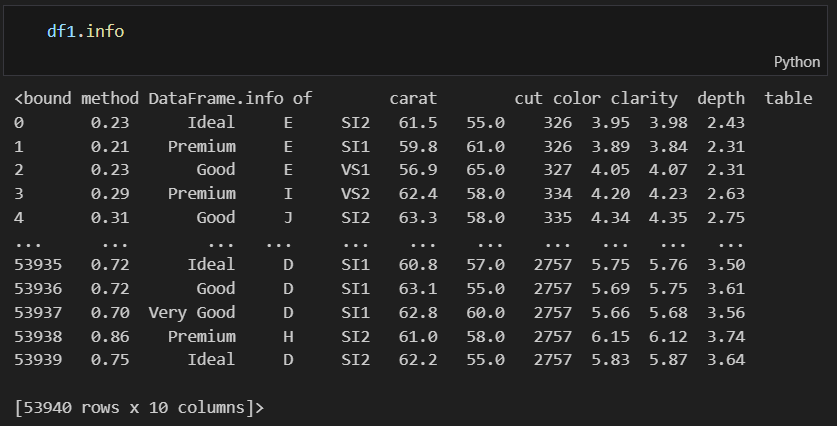
* df1.columns : Gọi ra các thuộc tính của bảng dữ liệu



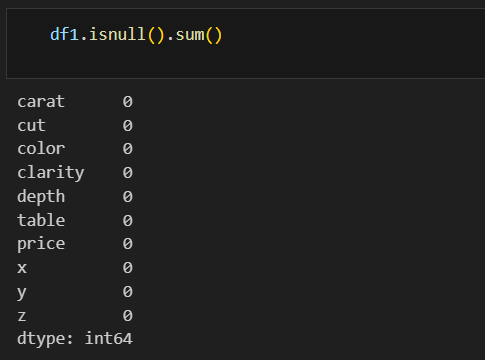
* df1.shape : Xem hình dạng của tập dữ liệu



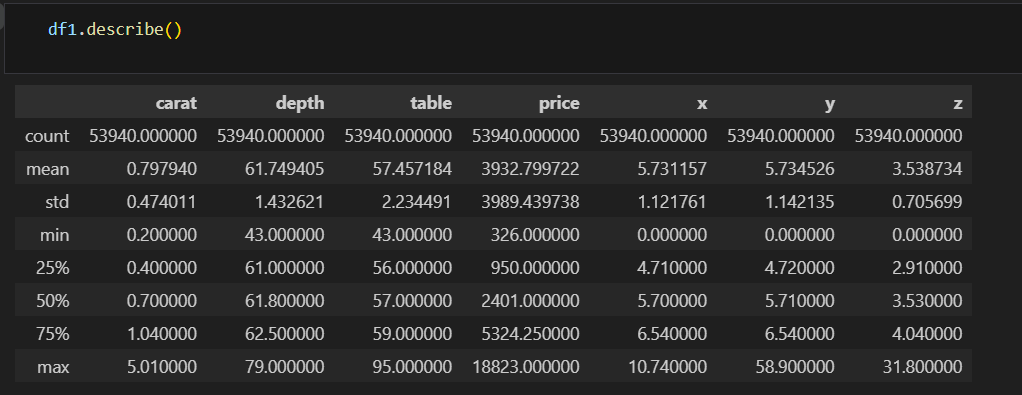
* df1.info: Hiển thị thông tin của bộ dữ liệu



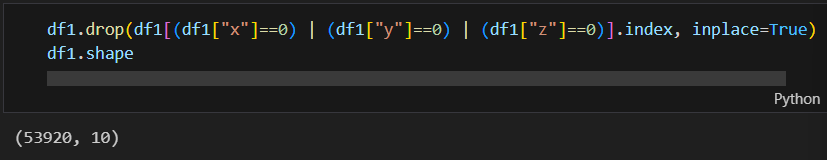
* df1.isnull().sum() : Tính tổng các giá trị bị thiếu ( null )



* df1.describe() : Hiển thị các thông số chi tiết của bộ dữ liệu

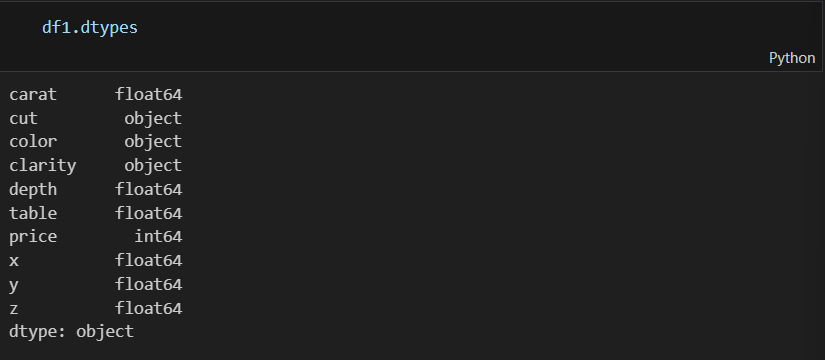


* df1.drop(df1[(df1["x"]==0) | (df1["y"]==0) | (df1["z"]==0)].index, inplace=True) : Loại bỏ những giá trị x, y, z có giá trị bằng 0 vì nếu 1 giá trị x hoặc y hoặc z bằng 0 thì thể tích sẽ bằng 0 :



**2.2. Phân tích dữ liệu thăm dò**

* Df1.dtypes : Hiển thị kiểu dự liệu của từng thuộc tính



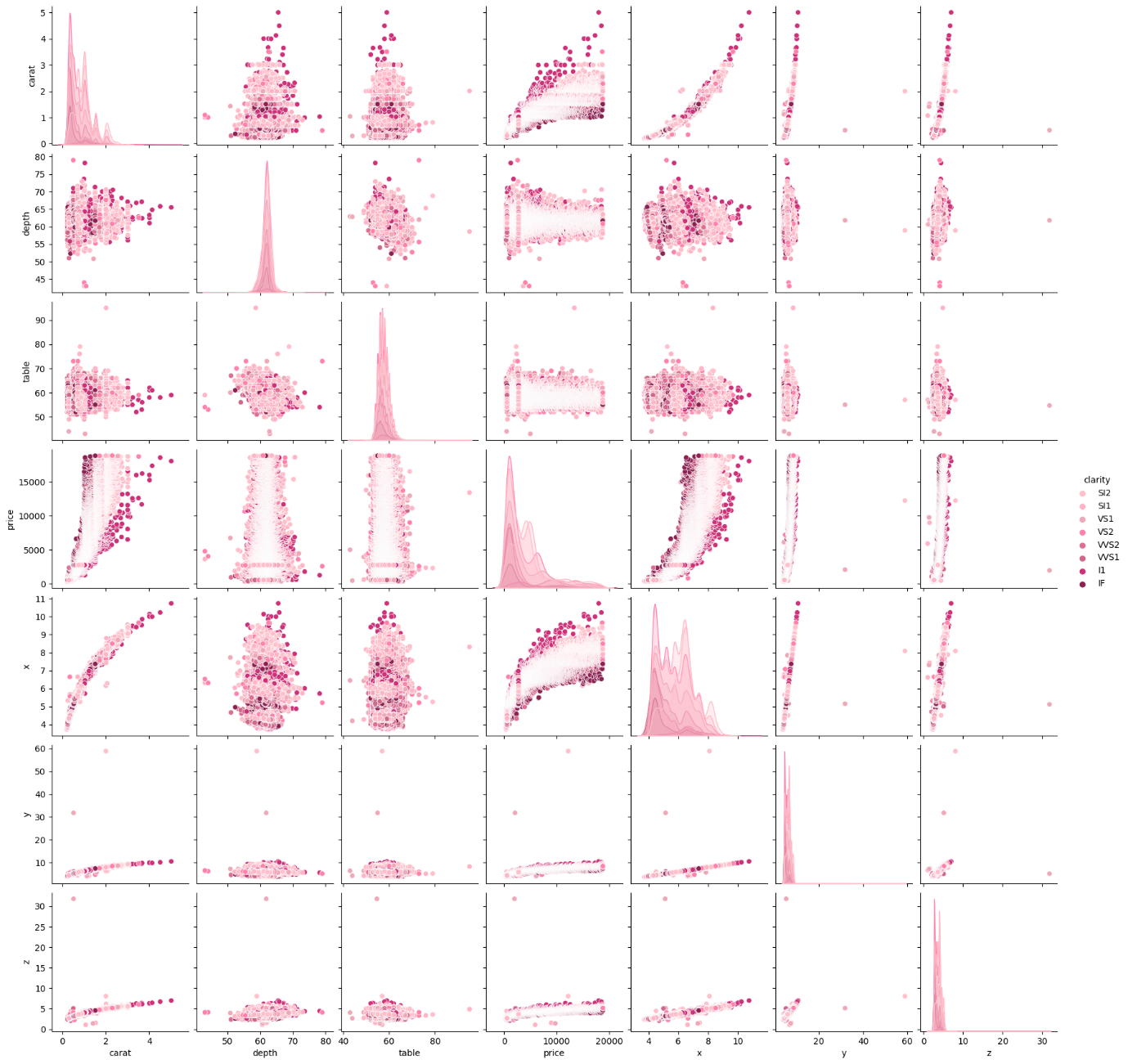
=> Target - Label - Output : Price

Numerical features : carat, depth, table, price, x, y, z -> xử lý null, normalize -> [0,1]

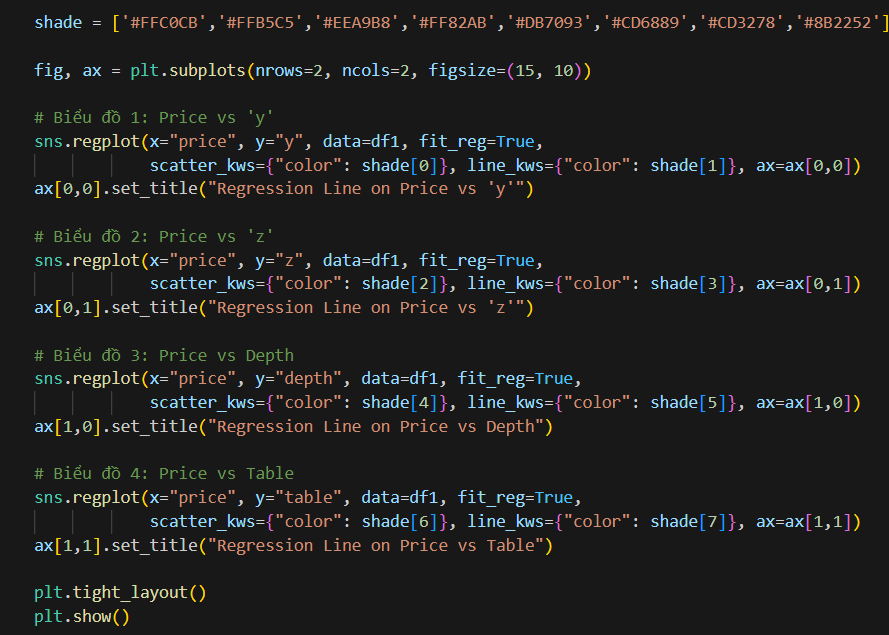
Categorical features : cut, color, clarity -> xử lý null, label encoder/ (value có tính so sánh) or one hot encoder - đưa về các cột = số nhóm, gom nhóm trong trường hợp số lượng nhóm to

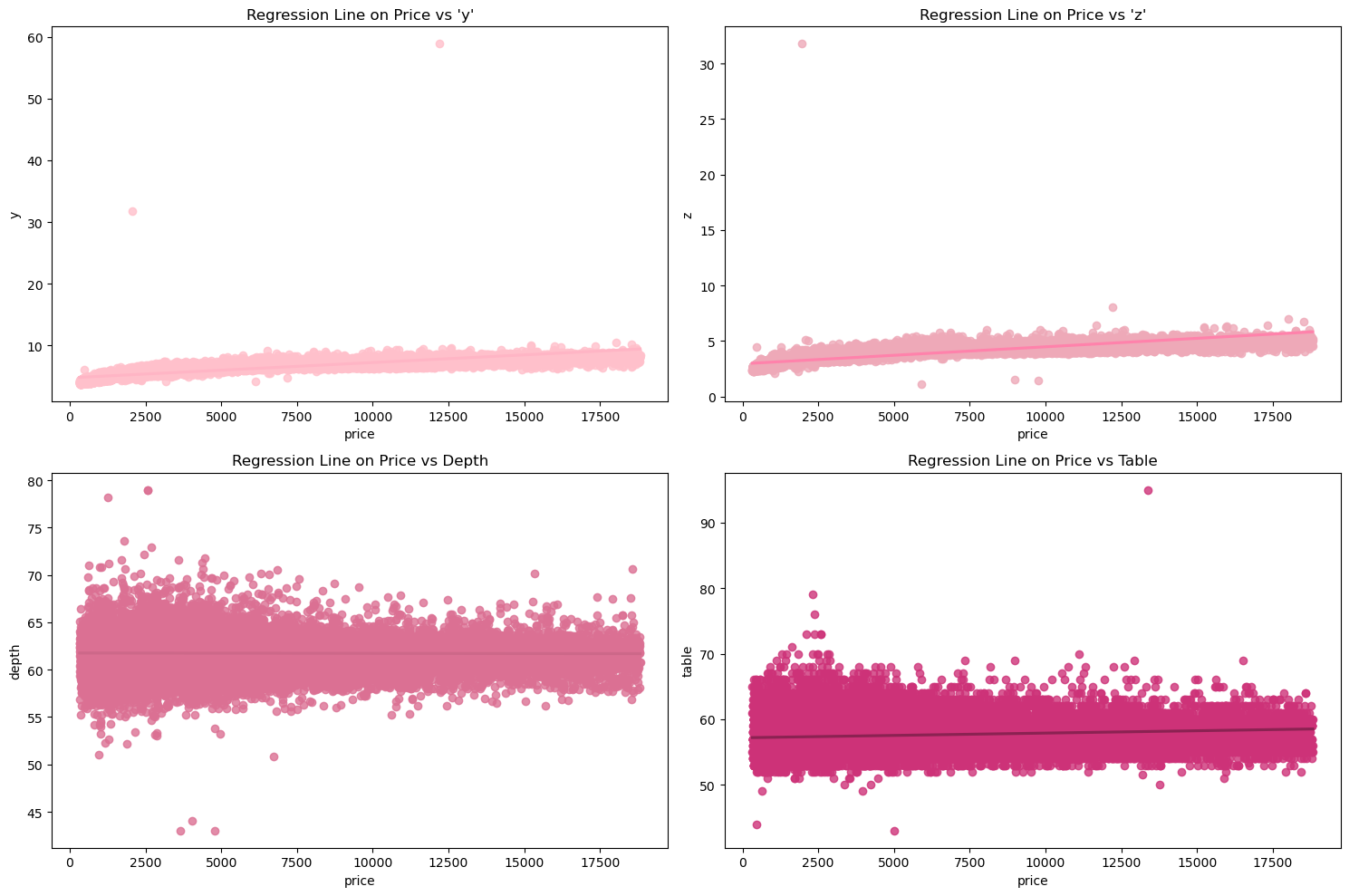
**III. Trực quan hóa dữ liệu**

**1. Biểu đồ pairplot**

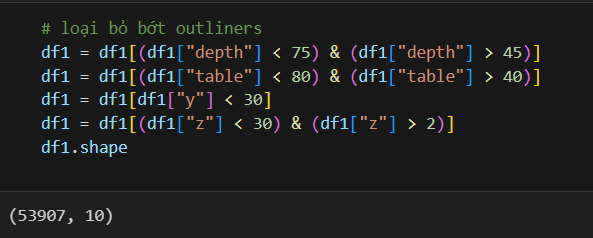


**2. Biểu đồ subplots**

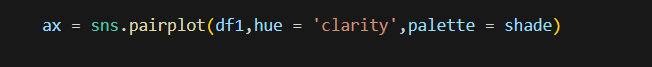


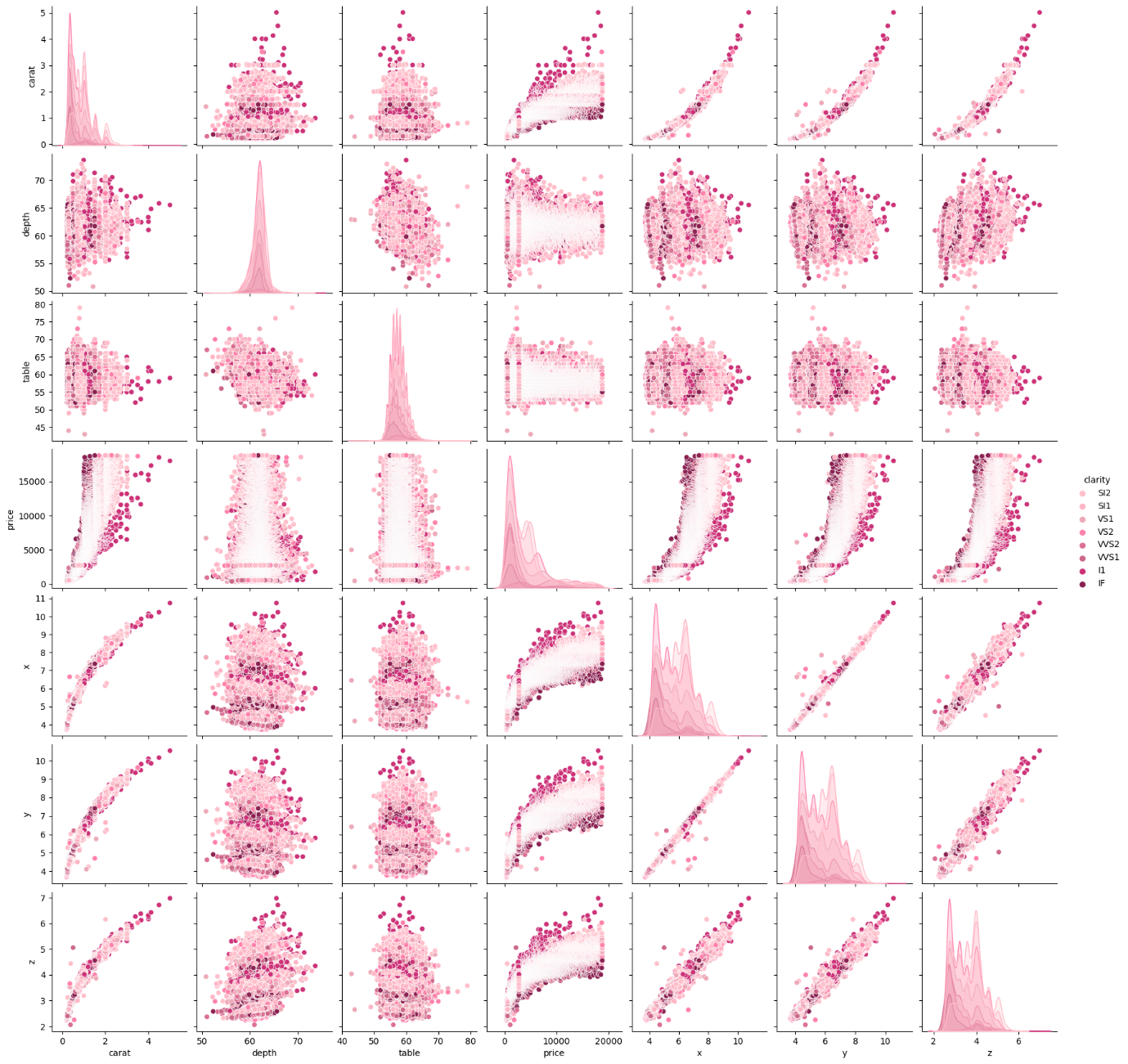


=> Có khá nhiều điểm ngoại lai cần được loại bỏ

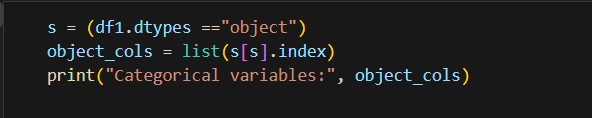
Loại bỏ bớt điểm ngoại lai:

Xem lại bộ dữ liệu bằng biểu đồ pairplots :



=> Bộ dữ liệu đã được làm sạch

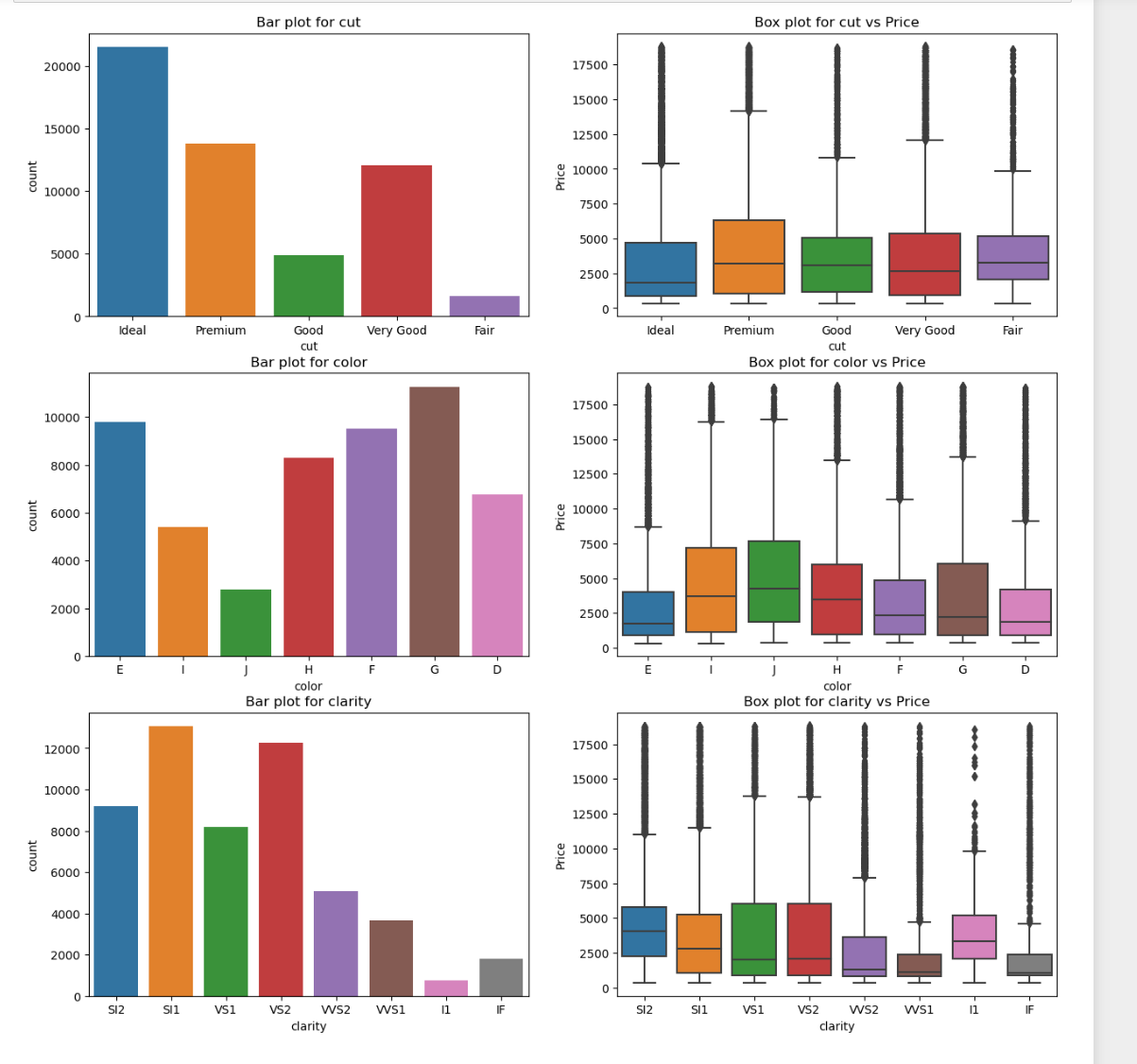
**IV. Các biến phân loại**



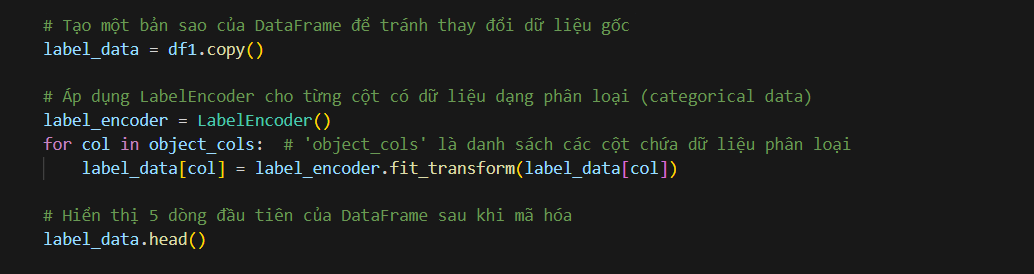
=> Có 3 biến phân loại là 'cut', 'color' và 'clarity'

Phân loại này mô tả mức độ khuyết điểm bên trong viên kim cương. Kim cương càng ít khuyết điểm sẽ có giá trị cao hơn.

Sử dụng biểu đồ subplots để phân tích 3 biến



**V. Biến đổi dữ liệu (chuẩn hóa dữ liệu)**

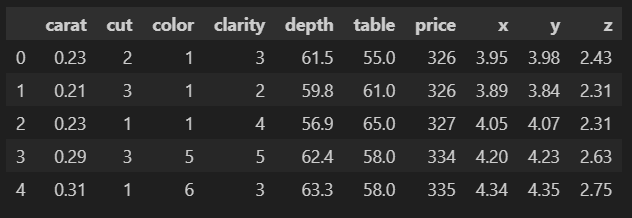


* Trước khi mã hóa dữ liệu cần phải tạo một bản sao Dataframe để xử lí tránh thay đổi dữ liệu gốc.
* Tiếp theo sử dụng class **LabelEncoder()**  và tạo biến lưu động **label\_encoder** được dùng để mã hóa dữ liệu phân loại trong các cột.
* **object\_cols** là danh sách các cột trong DataFrame có dữ liệu dạng phân loại (categorical).
* **label\_encoder.fit\_transform(label\_data[col]):** Phương thức **fit\_transform()** của LabelEncoder thực hiện hai bước:

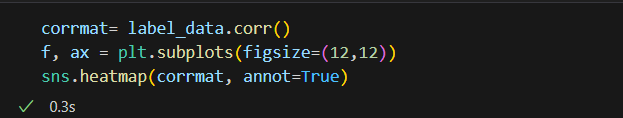
**+ fit():** Xác định các giá trị phân loại duy nhất trong cột và gán mỗi giá trị một số nguyên.

+ **transform():** Chuyển đổi các giá trị phân loại thành số nguyên tương ứng. Sau khi mã hóa, các giá trị phân loại trong cột sẽ được thay thế bằng các số nguyên trong bản sao label\_data.

=> Bảng dữ liệu sau khi được biến đổi:

 Như trên hình các cột dữ liệu như ”cut”, ”color”, ”clarity” đều được chuyển đổi các giá trị phân loại về số nguyên tương ứng.

Tiếp theo để thể hiện được mối tương quan giữa các biến số trong bảng dữ liệu, nhóm bọn em đã sử dụng biểu đồ heatmap để phân tích mối quan hệ giữa các đặc trưng:



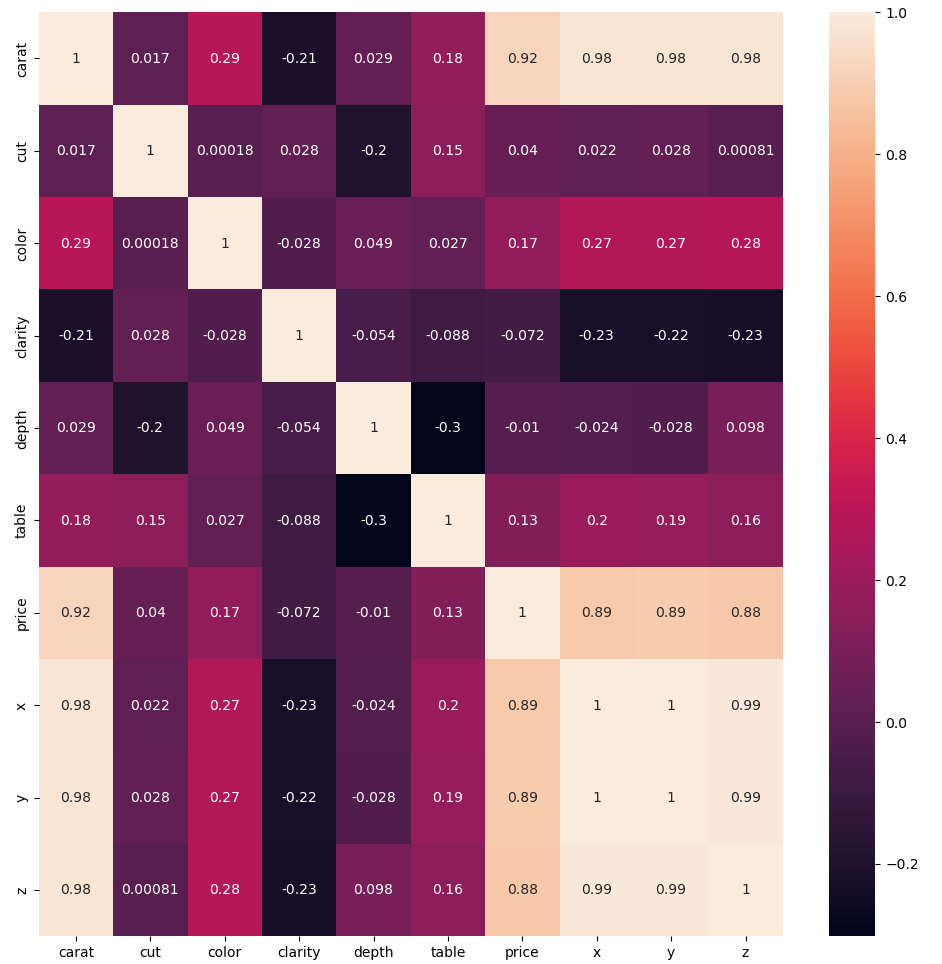
* Đầu tiên tạo một biểu đồ có kích cỡ 12x12 để hiển thị dữ liệu rõ ràng
* corrmat = label\_data.corr(): Hàm sẽ tính toán ma trận tương quan giữa các cột của DataFrame, cho biết mức độ tương quan của các biến số (Rxy)

+ (0;1): Tương quan dương (Khi một biến tăng thì biến kia cũng tăng tương ứng)

+ 0 : Các biến độc lập

+ (-1;0): Tương quan âm (Khi một biến giảm thì biến kia cũng giảm tương ứng)

* sns.heatmap(corrmat, annot=True): Vẽ biểu đồ heatmap và thêm giá trị vào các ô tương ứng

- "x", "y" và "z" có mối tương quan cao với cột mục tiêu. "depth", "cut" và "table" cho thấy mối tương quan thấp.

=> Chúng ta có thể xem xét việc loại bỏ những cột này.

* "carat" có mối tương quan cao với "x", "y" và "z".

**VI. Đào tạo mô hình**

**1. Thuật toán hồi quy tuyến tính**

**1.1. Lý thuyết**

* **Hồi quy tuyến tính (Linear Regression)** là một trong những mô hình học máy cơ bản và phổ biến nhất, dùng để dự đoán giá trị của một biến phụ thuộc (biến mục tiêu) dựa trên một hoặc nhiều biến độc lập (biến đầu vào). Nó thể hiện mối quan hệ giữa các biến này bằng một phương trình tuyến tính.
* Công thức tổng quát :
* Trong mô hình hồi quy tuyến tính, sẽ có công thức dự đoán là:

***Y = intercept+coef1.x1 +coef2.x2 +…+ coefn.xn***

* Trong đó:
* + Intercept: Là một hằng số đại diện cho giá trị dự đoán khi tất cả các biến độc lập (biến đầu vào) bằng 0
* + Coefficients: Là hệ số hồi quy cho từng biến đầu vào (còn gọi là trọng số) trong phương
* => Đối với bộ dữ liệu này chúng em sẽ chọn 6 thuộc tính quan trọng tương ứng với 6 biến đầu vào trong phương trình đó là “carat”, “cut”, “color”, “x”, “y”, “z”. Từ đó sau quá trình huấn luyện sẽ xuất ra được phương trình tương đương với bộ dữ liệu đã cung cấp.
* Mục tiêu của hồi quy tuyến tính là tìm ra một **đường thẳng** (hoặc mặt phẳng nếu nhiều biến độc lập) tốt nhất để dự đoán giá trị của biến phụ thuộc từ các biến độc lập.
* Hồi quy tuyến tính thường được dùng trong:

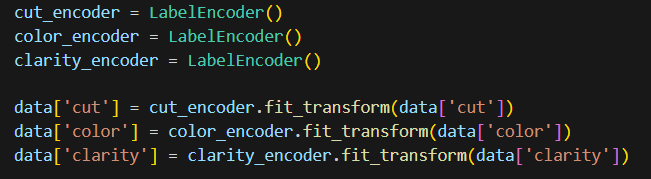
**+ Dự đoán xu hướng**: Ví dụ, dự đoán giá nhà dựa trên các yếu tố như diện tích, số phòng, v.v.

**+Phân tích quan hệ giữa các biến**: Xác định mức độ ảnh hưởng của các yếu tố đầu vào (biến độc lập) đến kết quả (biến phụ thuộc).

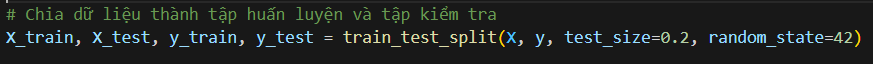
**+Kinh tế học**: Đo lường tác động của các yếu tố kinh tế (như lãi suất, lạm phát) lên các chỉ số kinh tế.

**1.2. Quy trình thực hiện**

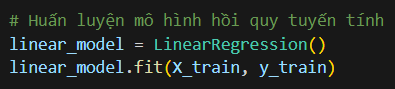
* Thực hiện mã hóa các cột “cut", “color", và “clarity" bằng cách sử dụng LabelEncoder.



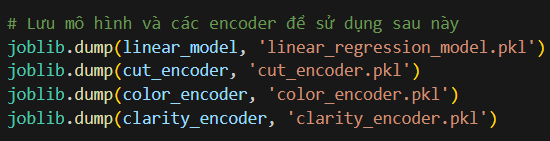
* Chọn ra những cột dữ liệu quan trọng nhất để đưa vào mô hình.
* Dữ liệu được chia thành hai phần: tập huấn luyện (*train set*) và tập kiểm tra (*test set*).

random\_state là một tham số dùng để kiểm soát tính ngẫu nhiên của thuật toán

* Tiến hành huấn luyện mô hình hồi quy tuyến tính (Linear - *Regression*).



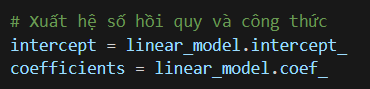
* Lưu lại mô hình cùng với các LabelEncoder đã sử dụng.



=> Huấn luyện mô hình thành công

**1.3. Kết quả thực hiện**

* Sau khi hoàn thành quá trình huấn luyện, chúng em đã xuất mô hình cùng với các encoder để sử dụng cho những lần sau.



* Cuối cùng thì chúng em in ra công thức hồi quy tuyến tính tổng quát cho bộ dữ liệu này.

print(f"y = {intercept:.2f} + {coefficients[0]:.2f}\*carat + {coefficients[1]:.2f}\*cut + {coefficients[2]:.2f}\*color + {coefficients[3]:.2f}\*clarity + {coefficients[4]:.2f}\*x + {coefficients[5]:.2f}\*y + {coefficients[6]:.2f}\*z")

Trong mô hình hồi quy tuyến tính, sẽ có công thức dự đoán là:

Y = intercept+coef1.x1 +coef2.x2 +…+ coefn.xn

Trong đó:

+ Intercept: Là một hằng số đại diện cho giá trị dự đoán khi tất cả các biến độc lập (biến đầu vào) bằng 0

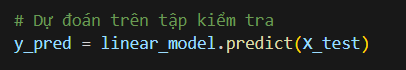
+ Coefficients: Là hệ số hồi quy cho từng biến đầu vào (còn gọi là trọng số) trong phương

=> Đối với bộ dữ liệu này chúng em sẽ chọn 6 thuộc tính quan trọng tương ứng với 6 biến đầu vào trong phương trình đó là “carat”, “cut”, “color”, “x”, “y”, “z”. Từ đó sau quá trình huấn luyện sẽ xuất ra được phương trình tương đương với bộ dữ liệu đã cung cấp.

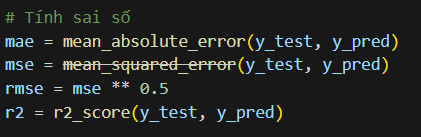
* Công thức hồi quy tuyến tính sẽ là:

**1.4 Test mô hình**

* Phương thức predict dùng để dự đoán các giá trị đầu ra cho các mẫu trong X\_test, dựa trên các trọng số (coefficients) và intercept mà mô hình đã học được từ dữ liệu huấn luyện.
* y\_pred: Đây là một mảng chứa các giá trị dự đoán của biến phụ thuộc từ mô hình, tương ứng với mỗi hàng trong X\_test

**.**

* Chúng em tính toán các sai số giữa giá trị dự đoán của mô hình và giá trị thực tế trong tập kiểm tra. Các sai số này được đo bằng nhiều chỉ số khác nhau để đánh giá mô hình từ nhiều khía cạnh, bao gồm:



* **MAE (Mean Absolute Error - Sai số tuyệt đối trung bình):** Đây là giá trị trung bình của độ lệch giữa dự đoán và giá trị thực tế, không quan tâm đến dấu của sai số (quá cao hay quá thấp). MAE giúp chúng ta hiểu được, trung bình, mô hình dự đoán sai bao nhiêu.

Công thức tính toán: MAE=

*Chú thích*

*n : Số lượng mẫu*

*: Giá trị thực tế*

*: Giá trị dự đoán*

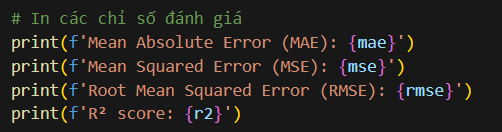
* **MSE (Mean Squared Error - Sai số bình phương trung bình):** Đây là trung bình của bình phương các sai số giữa giá trị dự đoán và giá trị thực. Vì sai số được bình phương, MSE đặc biệt nhạy cảm với những sai số lớn, giúp phát hiện các dự đoán có độ lệch lớn.

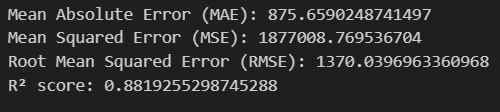
Công thức tính toán: MSE =

* **RMSE (Root Mean Squared Error - Căn bậc hai của sai số bình phương trung bình):** Là căn bậc hai của MSE, RMSE đưa kết quả về cùng đơn vị với biến mục tiêu (giá kim cương). Chỉ số này đại diện cho độ lệch chuẩn của các sai số dự đoán và cho chúng ta biết mô hình có dự đoán ổn định hay không.

Công thức tính toán: RMSE = =

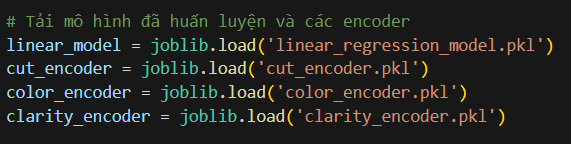
* **R² (R-squared - Hệ số xác định):** R² thể hiện mức độ mô hình có thể giải thích được sự biến thiên của biến phụ thuộc (giá kim cương). R² có giá trị từ 0 đến 1, với 1 là mức độ giải thích hoàn hảo. Giá trị cao cho thấy mô hình có khả năng dự đoán tốt.
* **Công thức tính toán :**
* In ra các kết quả



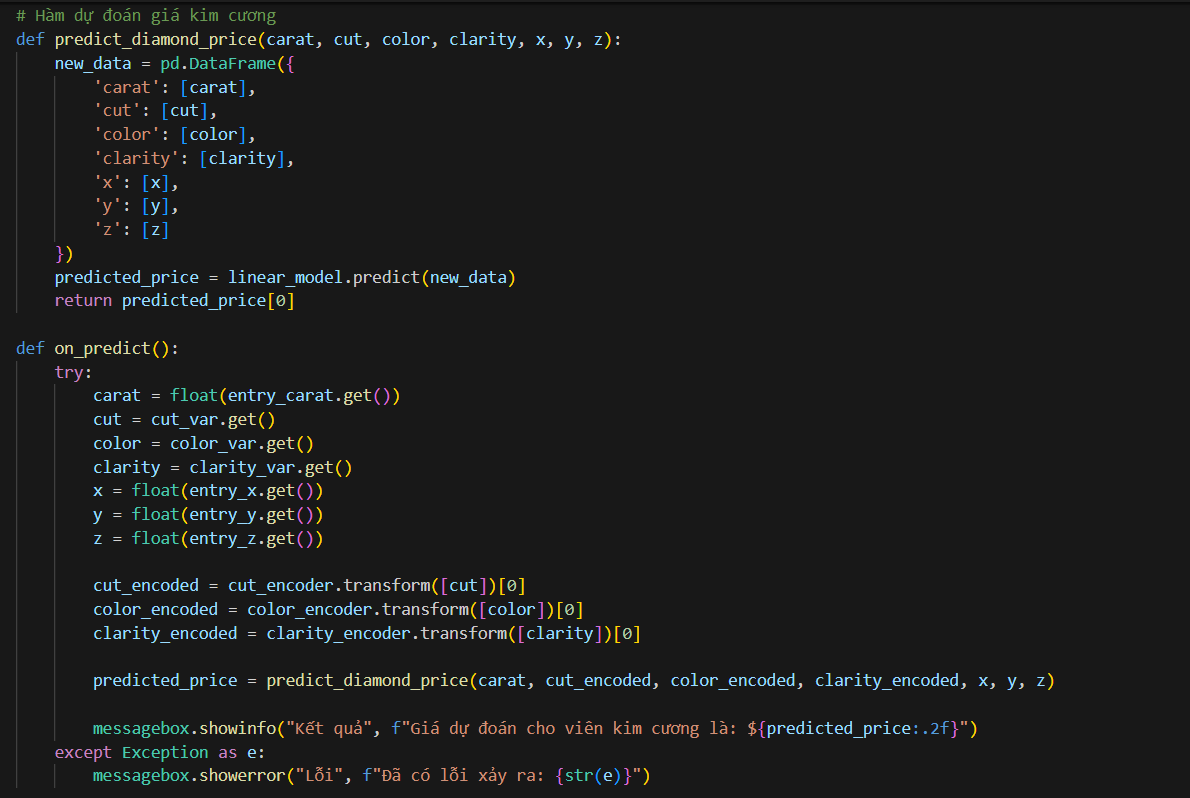
* Chúng ta sẽ được kết quả:
* **MAE (Mean Absolute Error):** 875.66. Sai số này đại diện cho khoảng cách trung bình giữa giá trị dự đoán và giá trị thực tế.
* **MSE (Mean Squared Error):** 1,877,008.77. Giá trị MSE lớn cho thấy rằng mô hình có một vài sai số khá lớn, bởi vì MSE phạt nặng các sai số lớn hơn do bình phương sai số.
* **RMSE (Root Mean Squared Error):** 1,370.04. RMSE đưa MSE về cùng đơn vị với biến mục tiêu và dễ hiểu hơn. Điều này cho thấy độ lệch chuẩn trung bình của các sai số dự đoán là 1,370 đơn vị tiền tệ.
* **R² (R-squared):** 0.8819. Hệ số R² gần bằng 0.88, tức là mô hình có thể giải thích khoảng 88.19% sự biến thiên trong giá kim cương dựa trên các đặc trưng đầu vào (như carat, cut, color, clarity, kích thước kim cương). Đây là một kết quả khá tốt, cho thấy mô hình có khả năng dự đoán tương đối chính xác.

**1.5.Tạo cửa sổ dự đoán**

* Tải mô hình hồi quy tuyến tính đã được huấn luyện từ trước, cùng với các encoder đã sử dụng để mã hóa các cột phân loại (*cut*, *color*, *clarity*)



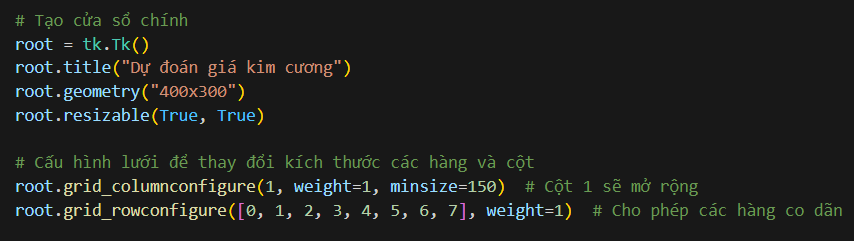
* Xây dựng một hàm dự đoán giá kim cương dựa trên các thông số đầu vào. Hàm này sẽ lấy dữ liệu nhập vào, chuyển đổi các giá trị phân loại thành số sử dụng các encoder đã lưu và sử dụng mô hình để đưa ra dự đoán giá cuối cùng. Kết quả dự đoán sẽ được trả về cho giao diện hiển thị.

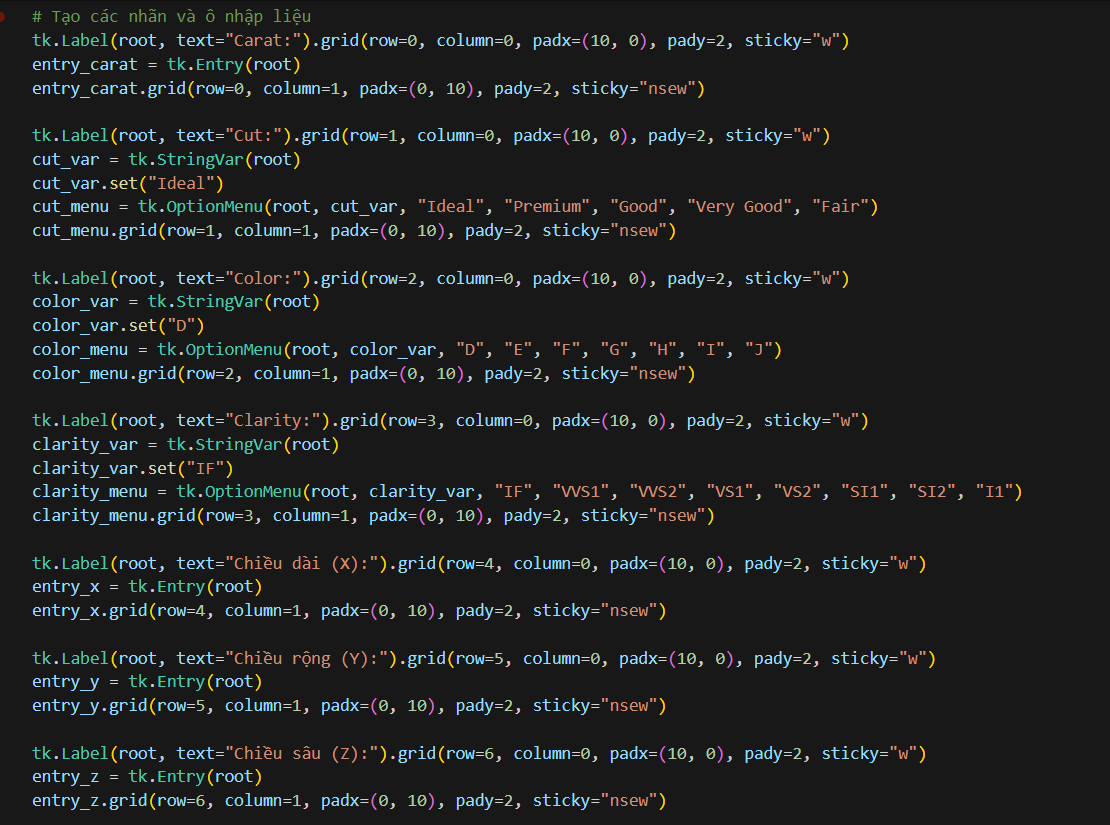
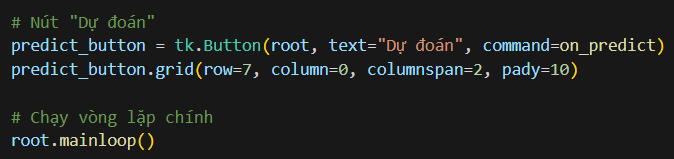
+ def predict\_diamond\_price: Khai báo hàm dự báo với các tham số đặc trưng. Mỗi đặc trưng sẽ được truyền vào dưới dạng một danh sách (value)

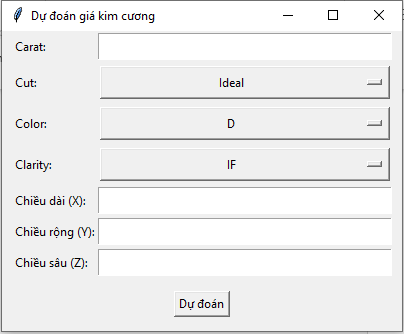
+ Lấy giá trị từ các trường nhập liệu đối với “carat”, “x”, “y”, “z” và biến chúng thành số thực. Lấy giá trị từ biến lựa chọn đối với “cut”, “color”, “clarity”.

+ Tiếp tục gọi lại hàm predict\_diamond\_price với các giá trị được mã hóa và các kích thước để nhận lại giá dự đoán cho viên kim cương. Sau đó in giá trị đã dự đoán ra ngoài màn hình.

* Tạo giao diện người dùng bằng cách thiết kế cửa sổ chính có thể kéo thả.



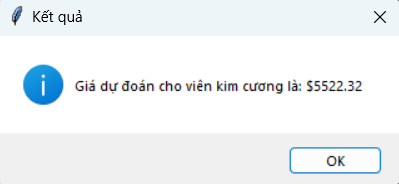
* Trong cửa sổ chính, chúng em thêm các nhãn (*labels*) để hướng dẫn người dùng nhập thông tin 
* Thêm một nút "Dự đoán". Khi người dùng nhấn nút này, hàm dự đoán giá kim cương sẽ được gọi, sử dụng dữ liệu đầu vào từ các ô nhập liệu, và hiển thị kết quả giá kim cương dự đoán trên giao diện.
* Người dùng có thể nhập dữ liệu liên quan đến viên kim cương của họ và nhấn nút "Dự đoán". Hệ thống sẽ xử lý dữ liệu và hiển thị giá dự đoán ngay trên cửa sổ giao diện.



* Khi chúng ta nhập dữ liệu như hình



* Thì sẽ ra được kết quả



=> Kết quả dự đoán cho ta thấy sai số giữa giá trị thực với giá trị dự đoán chênh nhau giao động trong khoảng từ 200-600$. Đây là một mức sai số cao chiếm khoảng 12% giá trị thực của viên kim cương. Đối với những viên nhỏ, ít giá trị thì sai số không ảnh hưởng nhiều. Tuy nhiên khi lên tới những viên tầm trung hoặc cao cấp thì sai số ảnh hưởng lớn tới khả năng dự đoán thực tế.

**2. Thuật toán Random Forest**

**2.1. Lý thuyết**

**Random Forest** là một thuật toán học máy sử dụng **ensemble learning** (học ghép) để tạo ra một mô hình dự đoán mạnh mẽ. Random Forest hoạt động bằng cách xây dựng nhiều cây quyết định (**decision trees**) từ các tập dữ liệu mẫu khác nhau và lấy kết quả dự đoán bằng cách **bỏ phiếu đa số** (đối với phân loại) hoặc **trung bình** (đối với hồi quy).

### **Đặc điểm chính:**

1. **Bootstrap Aggregating (Bagging):** Mỗi cây trong rừng được huấn luyện trên một mẫu ngẫu nhiên của dữ liệu gốc (có thể lặp lại).
2. **Random Subspace:** Tại mỗi nút của cây, chỉ một tập con của các đặc trưng được chọn ngẫu nhiên để tìm thuộc tính tốt nhất chia tách dữ liệu.
3. **Tính ổn định cao:** Giảm thiểu vấn đề **overfitting** vì dựa trên nhiều cây thay vì chỉ một.
4. **Độ chính xác tốt:** Hoạt động tốt với cả dữ liệu phân loại và hồi quy, kể cả khi dữ liệu nhiễu.

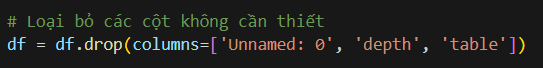
**Ưu điểm:**

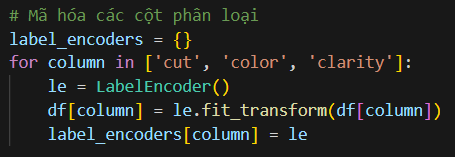
* Khả năng chống overfitting tốt.
* Hiệu quả với dữ liệu lớn và nhiều biến.
* Tự động xử lý dữ liệu thiếu.

**Nhược điểm:**

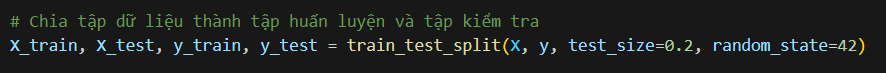
* Có thể chậm khi số lượng cây rất lớn.
* Mất khả năng giải thích trực quan do sử dụng nhiều cây.

**2.2. Quy trình thực hiện**

* Đầu tiên chúng em sẽ xóa các cột không cần thiết 
* Tiếp theo chúng em sẽ mã hóa các cột danh mục

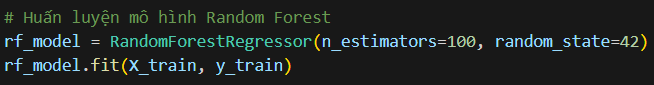


* Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra

=> Tập dữ liệu được chia làm hai phần: 80% dùng để huấn luyện và 20% còn lại để kiểm tra hiệu suất của mô hình

random\_state=42 đảm bảo tính tái lập khi chạy lại mô hình

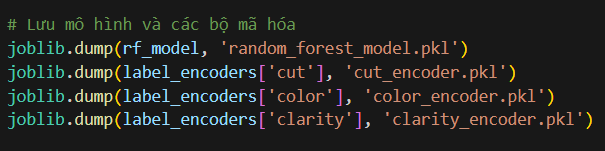
* **Huấn luyện mô hình Random Forest**



* Một mô hình **Random Forest** với 100 cây (trees) được khởi tạo. Mỗi cây sẽ học từ các mẫu dữ liệu khác nhau và đưa ra dự đoán riêng của nó. Với càng nhiều cây thì độ chính xác mô hình càng cao nhưng đồng thời thời gian chạy sẽ tăng lên.
* Mô hình được huấn luyện trên tập dữ liệu huấn luyện (X\_train, y\_train)

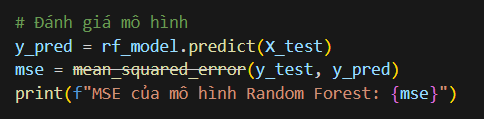
**=> Random Forest** là một thuật toán mạnh trong học máy dựa trên việc tạo ra nhiều cây quyết định (decision trees) để đưa ra các dự đoán, sau đó lấy trung bình (hoặc bỏ phiếu) từ các cây này để đưa ra kết quả cuối cùng, giúp giảm thiểu vấn đề quá khớp (overfitting)

* Lưu mô hình và các bộ mã hóa



- Sau khi huấn luyện, mô hình và các bộ mã hóa được lưu trữ thành tệp (.pkl) để có thể sử dụng lại mà không cần phải huấn luyện lại từ đầu

- Cuối cùng là đánh giá mô hình



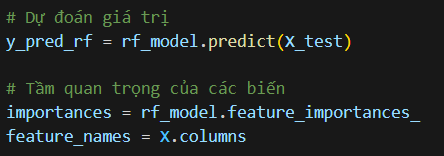
- Mô hình được kiểm tra bằng cách dự đoán giá trị price trên tập dữ liệu kiểm tra (X\_test)

- Chúng ta được kết quả:



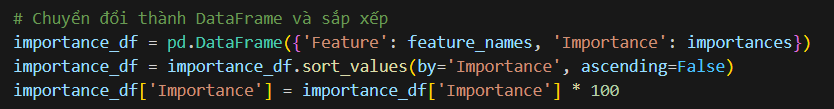
**Mean Squared Error (MSE)** cho thấy mức độ sai lệch giữa giá trị thực và giá trị dự đoán.

**2.3. Kết quả thu được**



+ y\_pred\_rf lưu trữ kết quả dự đoán của mô hình

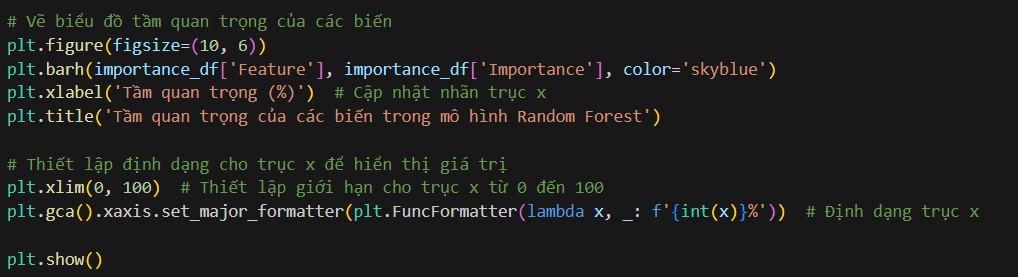
+ Hàm predict() của mô hình sẽ tạo ra các giá trị dự đoán dựa trên các biến đầu vào

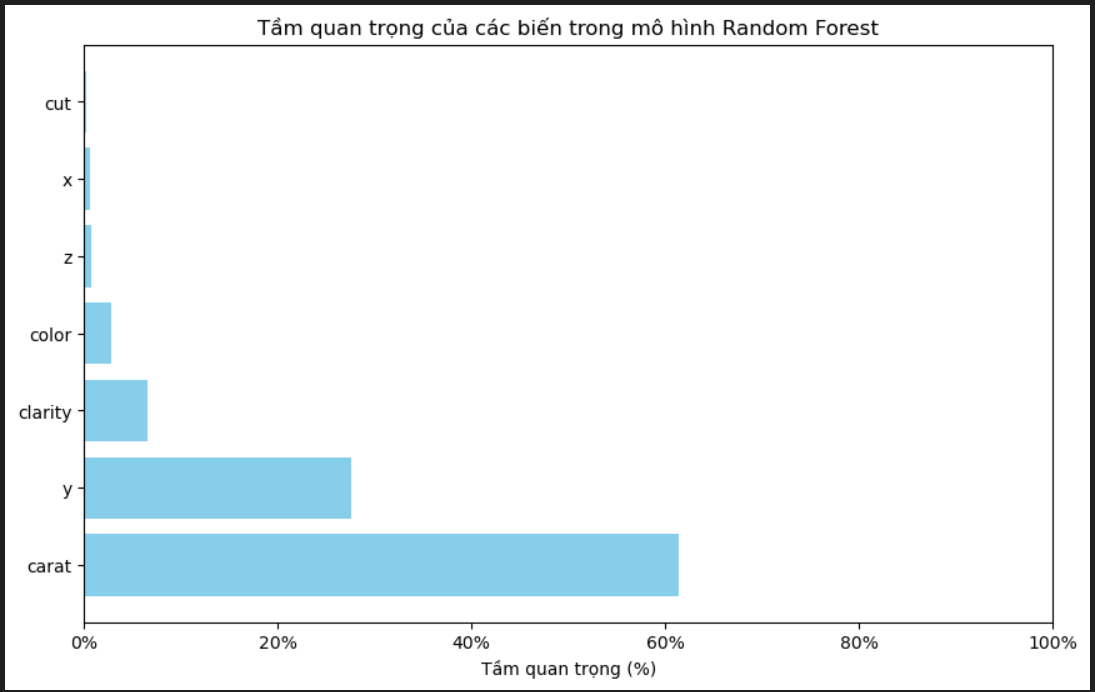
- Đầu tiên tạo một DataFrame có tên là importance\_df với hai cột:

+ **Feature**: Chứa tên các biến (đặc trưng) trong mô hình.

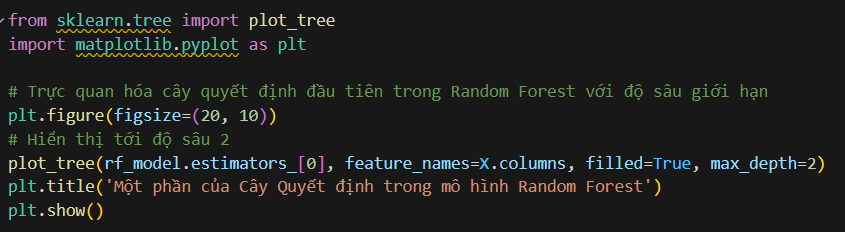
+ **Importance**: Chứa giá trị tầm quan trọng tương ứng của các biến.

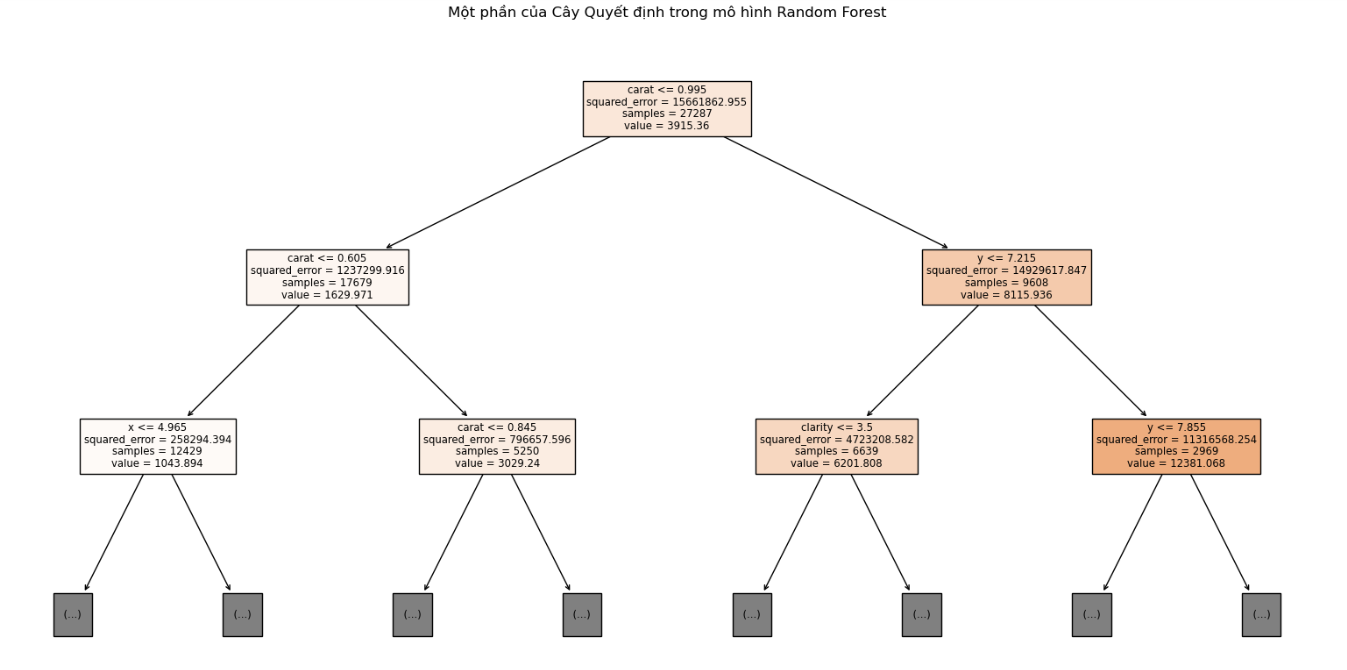
- Sau đó sắp xếp DataFrame theo thứ tự giảm dần rồi chuyển đổi tỉ lệ sang phần trăm.

Đây là bước vẽ biểu đồ để thể hiện mức độ quan trọng, ảnh hưởng của thuộc tính đối với khả năng dự . Cuối cùng sẽ xuất ra một biểu đồ cột ngang như dưới hình.



=> Như biểu đồ trên ta thấy mức độ ảnh hưởng của ”carat” đối với khả năng dự đoán là cao nhất chiếm tới hơn 60%. Các thuộc tính còn lại ảnh hưởng ở mức trung bình-thấp.

 Đoạn code trên nhằm trực quan hóa bằng cây quyết định có độ sâu bằng 2. Sau đây là cây hiển thị:



**- Chú thích:**

**+ Carat:** đơn vị đo khối lượng kim cương

**+ squared\_error:** Tổng bình phương sai số - Là một chỉ số đo lường sai lệch giữa giá trị thực tế và giá trị dự đoán.

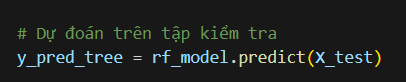
=> **Công thức tổng quát:** squared\_error = (Y\_thực tế - Y\_Tổng quát)^2

**+ samples:** Số lượng mẫu dữ liệu (điểm dữ liệu) có mặt tại một node nhất định trong cây quyết định.

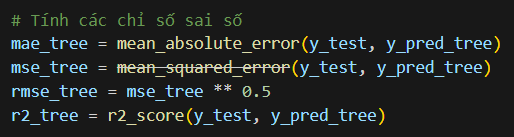
**+ Value:** Biểu thị giá trị trung bình của biến mục tiêu (target) cho các mẫu thuộc node đó.

**2.4 Test mô hình**

* Dự đoán trên tập kiểm tra

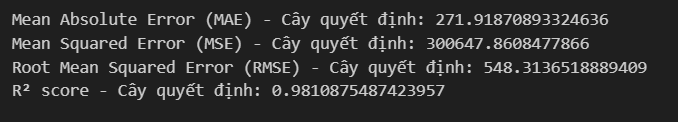


* Tính các chỉ số sai số

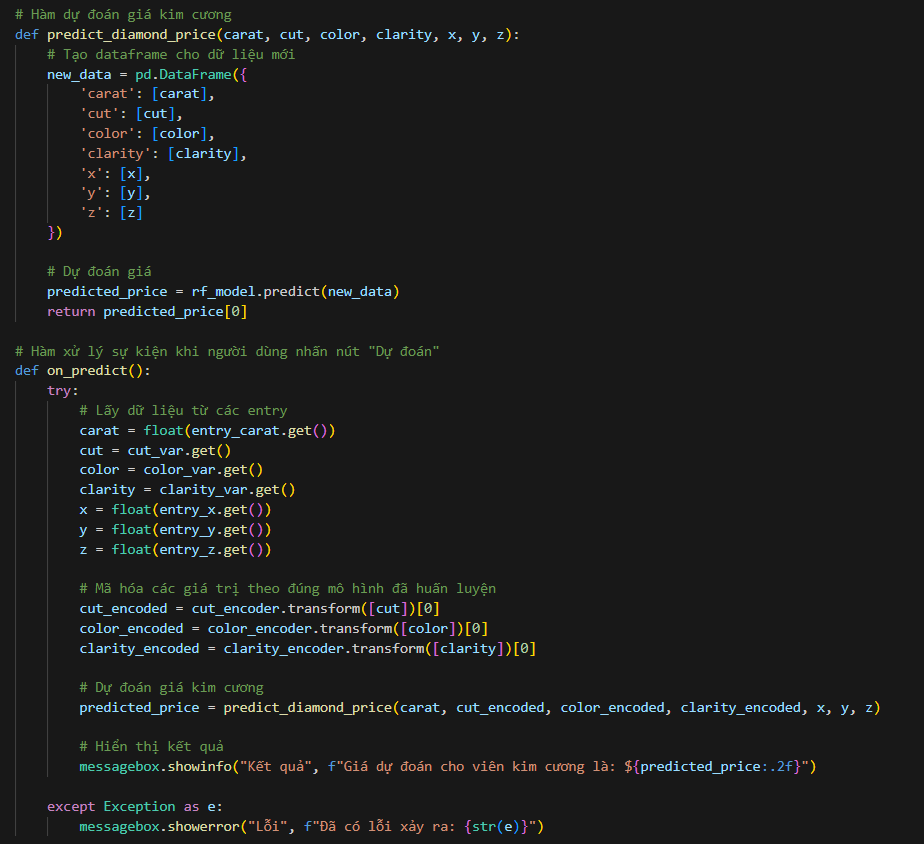


Các chỉ số đánh giá hiệu suất được tính toán để so sánh giữa các giá trị thực tế **(y\_test)** và các giá trị dự đoán **(y\_pred\_tree)**

* Kết quả:

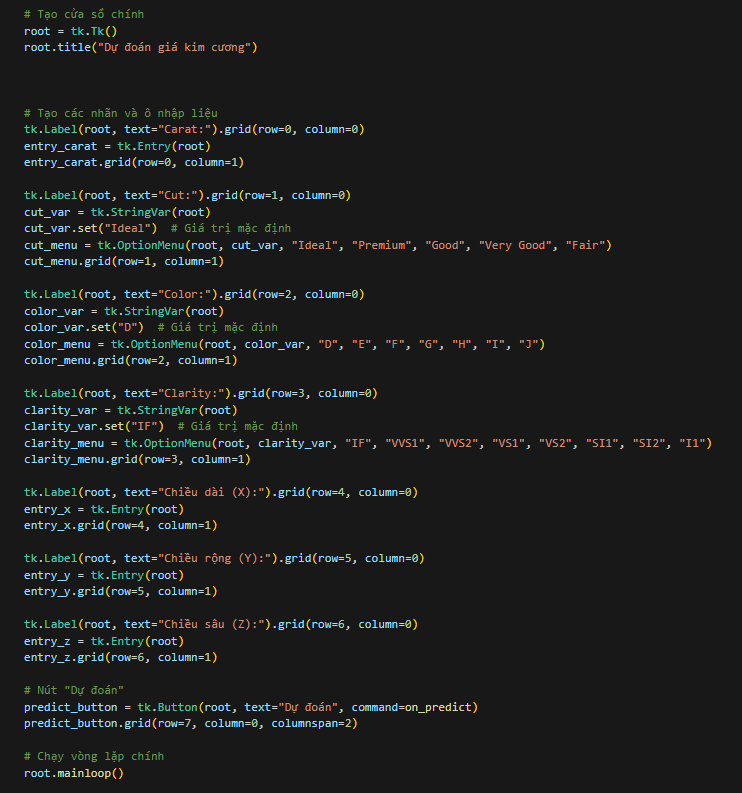


- Tải mô hình đã huấn luyện và viết hàm dự đoán giá kim cương:

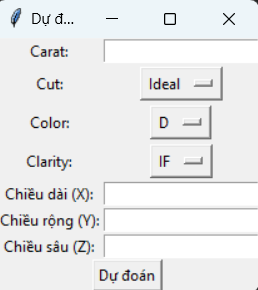


**2.5 Tạo cửa sổ dự đoán**

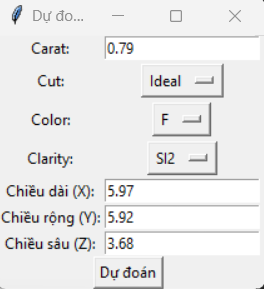
- Chương trình tạo giao diện nhập dữ liệu:



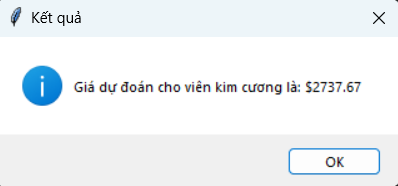
- Cửa sổ dự đoán



- Nhập dữ liệu



- Kết quả thu được



=> Kết quả dự đoán cho ta thấy sai số giữa giá trị thực với giá trị dự đoán chênh nhau giao động trong khoảng từ 5-80$. Đây là một mức sai số rất thấp chiếm khoảng 1,9% giá trị thực của viên kim cương. Mức sai số này có thể chấp nhận được. Vì thế việc sử dụng thuật toán Random Forest cho mô hình trên là thích hợp hơn.

**VII. So sánh 2 thuật toán (Linear Regression và Random Forest)**

|  |  |  |
| --- | --- | --- |
|  | **Hồi quy tuyến tính (Linear Regression)** | **Random forest** |
| **Cách thức hoạt động** | -Đây là mô hình đơn giản, giả định mối quan hệ tuyến tính giữa biến đầu vào (đặc trưng) và đầu ra (kết quả).  -Mô hình cố gắng tìm một đường thẳng (hoặc mặt phẳng trong không gian nhiều chiều) phù hợp nhất với dữ liệu, để tối thiểu hóa tổng bình phương sai số giữa các giá trị thực tế và dự đoán. | -Random Forest là một mô hình phức tạp hơn, dựa trên một tập hợp các cây quyết định. Nó tạo nhiều cây quyết định ngẫu nhiên trên các tập con khác nhau của dữ liệu và sử dụng kết quả trung bình từ các cây này để đưa ra dự đoán cuối cùng. |
| **Độ chính xác** | -R² (R-squared): 88,19%  -sai số giữa giá trị thực với giá trị dự đoán chênh nhau giao động trong khoảng từ 200-600$. | - R² (R-squared):98,1%  - sai số giữa giá trị thực với giá trị dự đoán chênh nhau giao động trong khoảng từ 50-100$. |
| **Hiệu suất tính toán** | -Tốc độ huấn luyện nhanh hơn và yêu cầu ít tài nguyên tính toán do tính đơn giản của mô hình. | -Yêu cầu tài nguyên tính toán lớn hơn, do cần huấn luyện nhiều cây quyết định và tính trung bình kết quả từ nhiều cây. |
| **Thời gian huấn luyện** | Nhanh chóng, ít yêu cầu tính toán | Chậm hơn, cần nhiều thời gian hơn khi số lượng cây lớn |
| **Thời gian dự đoán** | Nhanh, đặc biệt là khi xử lý dữ liệu lớn | Chậm hơn do phải tổng hợp dự đoán từ nhiều cây |

=> **Random Forest** vượt trội hơn về độ chính xác và khả năng xử lý các đặc trưng phù hợp hơn với “Mô hình dự đoán giá kim cương”

**KẾT LUẬN**

Sau khi hoàn thành đề tài “Mô hình dự đoán giá kim cương**”,** nhóm em đã đạt được kết quả sau:

* **Xử lý và trực quan hóa dữ liệu**: Sử dụng Python, nhóm đã tiến hành tiền xử lý dữ liệu một cách chi tiết và trực quan hóa các yếu tố quan trọng để làm nền tảng cho mô hình dự đoán.
* **Áp dụng mô hình**: Chúng em đã thử nghiệm hai mô hình Hồi quy tuyến tính và Random Forest. Kết quả cho thấy Random Forest có độ chính xác cao hơn, thích hợp với các đặc trưng trong dữ liệu của nhóm đã đưa ra, giúp cho mô hình dự đoán sát với thực tế nhất.

Trong quá trình hoàn thành đề tài, dù nhóm em đã cố gắng tìm hiểu và thực hiện bài toán nhưng vẫn không thể tránh những thiếu sót, khiếm khuyết. Do vậy, rất mong nhận được sự đóng góp ý kiến của thầy cô để chúng em rút kinh nghiệm, cải thiện kỹ năng và tích lũy kiến thức trong môn học.

Chúng em xin chân thành cảm ơn thầy !

à hồi quy tuyến tính để các mô hình có thể dự đoán chính xác mức giá của một viên kim cương dựa trên những thuộc tính cụ thể.

-So sánh kết quả của 2 thuật toán để thu được kết quả tốt nhất

Trong quá trình hoàn thành đề tài, dù nhóm em đã cố gắng tìm hiểu và thực hiện bài toán nhưng

vẫn không thể tránh những thiếu sót, khiếm khuyết. Do vậy, rất mong nhận được sự đóng góp ý

kiến của thầy cô để chúng em rút kinh nghiệm, cải thiện kỹ năng và tích lũy kiến thức trong môn

học.