TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN

DỰ ĐOÁN GIÁ KIM CƯƠNG

Khoa học dữ liệu

NHÓM 9

Giáp Ngọc Hiệu	102200254
Nguyễn Nho Song Hoàng	102200257
Lê Tự Minh Tuấn	102200292

Mục tiêu và giải pháp

Mục tiêu:

- Đầu vào là giá trị của các đặc trưng trong từng viên kim cương
- Đầu ra là giá của viên kim cương

Giải pháp:

- Sử dụng việc lựa chọn, chuẩn hóa các đặc trưng và sử dụng 2 mô hình Linear Regression và Random Forest Regression để huấn luyện và đưa ra dự đoán về giá trị

https://www.allurez.com/loose-diamonds

Nội dung

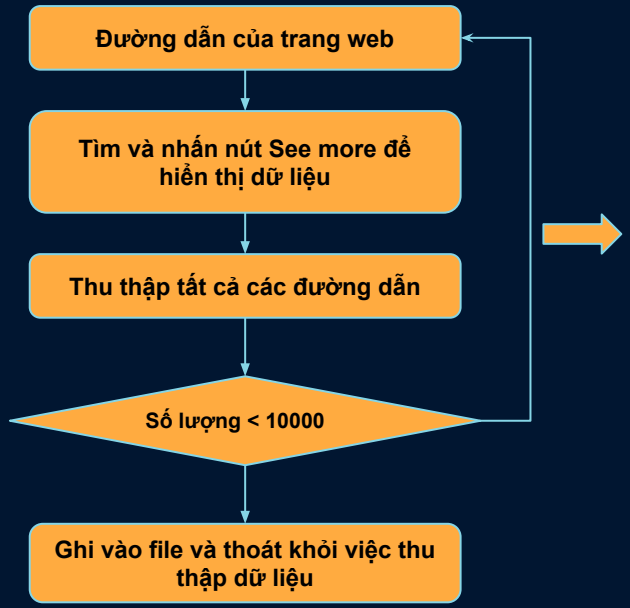
01 Thu thập dữ liệu

02 Trích xuất đặc trưng

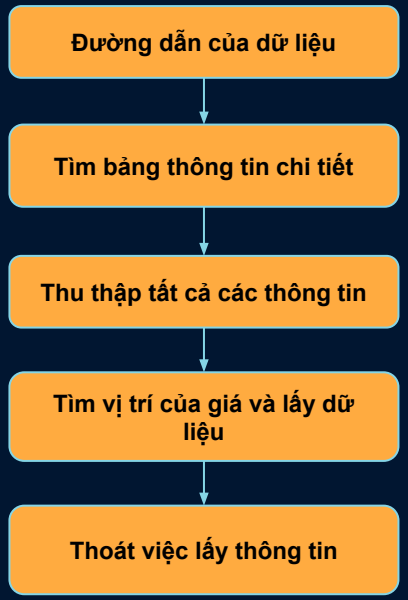
03 Mô hình dự đoán

I. Thu thập dữ liệu

- Việc thu thập dữ liệu được thực hiện thông qua 2 quá trình sau
  - Quá trình 1: thu thập tất cả đường dẫn của dữ liệu
  - Quá trình 2: truy cập từng đường dẫn và lấy dữ liệu theo định dạng của trang web



Cách thức thu thập đường dẫn



Cách thức thu thập dữ liệu

Style Number	10431	non-null	object
Shape	10431	non-null	object
Carat Weight	10431	non-null	float64
Color	10431	non-null	object
Clarity	10431	non-null	object
Graded By	10431	non-null	object
Cut Grade	10431	non-null	object
Fluorescence	10272	non-null	object
Culet	3136	non-null	object
Depth	10279	non-null	float64
Table	10286	non-null	float64
Girdle	8979	non-null	object
Polish	10431	non-null	object
Symmetry	10431	non-null	object
Measurements	10418	non-null	object
Price	10431	non-null	int64

Style Number	0
Shape	0
Carat Weight	0
Color	0
Clarity	0
Graded By	0
Cut Grade	0
Fluorescence	159
Culet	7295
Depth	152
Table	145
Girdle	1452
Polish	0
Symmetry	0
Measurements	13
Price	0

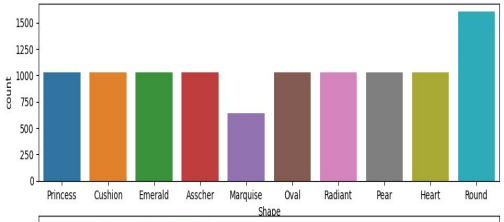
Số lượng dữ liệu và số lượng giá trị NULL (big)

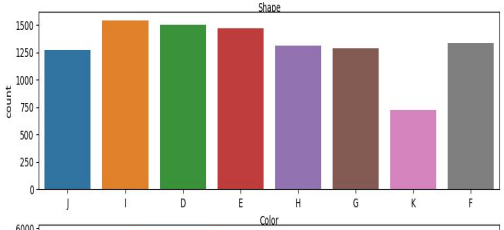
Style Number	1033	non-null	object
Shape	1033	non-null	object
Carat Weight	1033	non-null	float64
Color	1033	non-null	object
Clarity	1033	non-null	object
Graded By	1033	non-null	object
Cut Grade	1033	non-null	object
Fluorescence	1017	non-null	object
Culet	325	non-null	object
Depth	1019	non-null	float64
Table	1019	non-null	float64
Girdle	895	non-null	object
Polish	1033	non-null	object
Symmetry	1033	non-null	object
Measurements	1032	non-null	object
Price	1033	non-null	int64

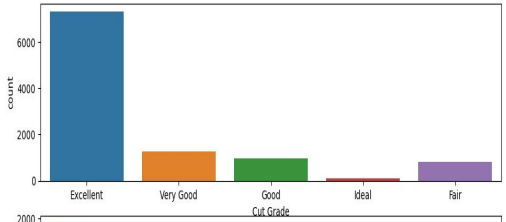
Style Number	0
Shape	0
Carat Weight	0
Color	0
Clarity	0
Graded By	0
Cut Grade	0
Fluorescence	16
Culet	708
Depth	14
Table	14
Girdle	138
Polish	0
Symmetry	0
Measurements	1
Price	0

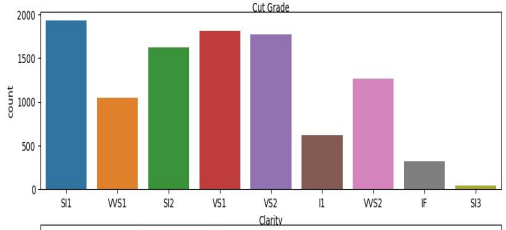
Số lượng dữ liệu và số lượng giá trị NULL (small)

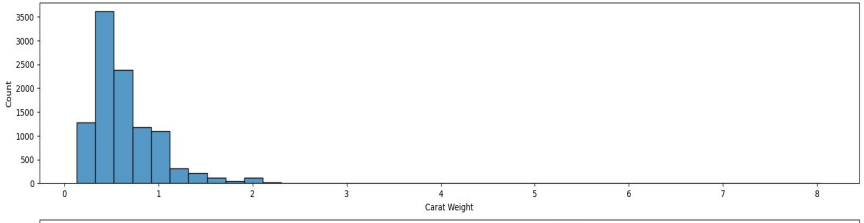
Một số thông tin của dữ liệu

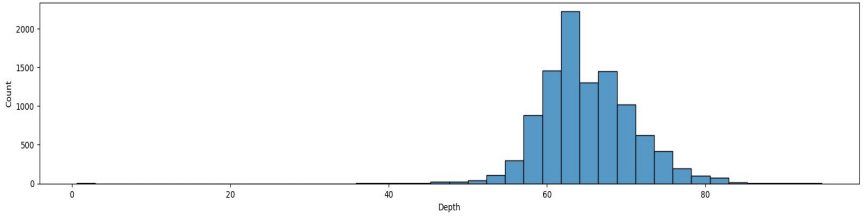


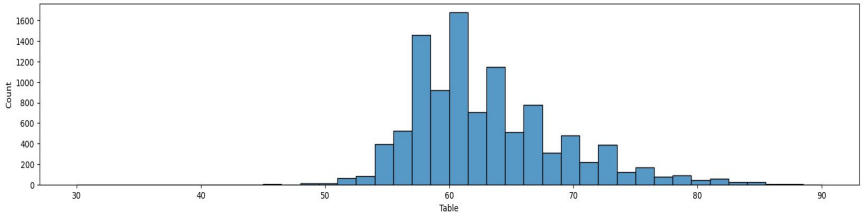










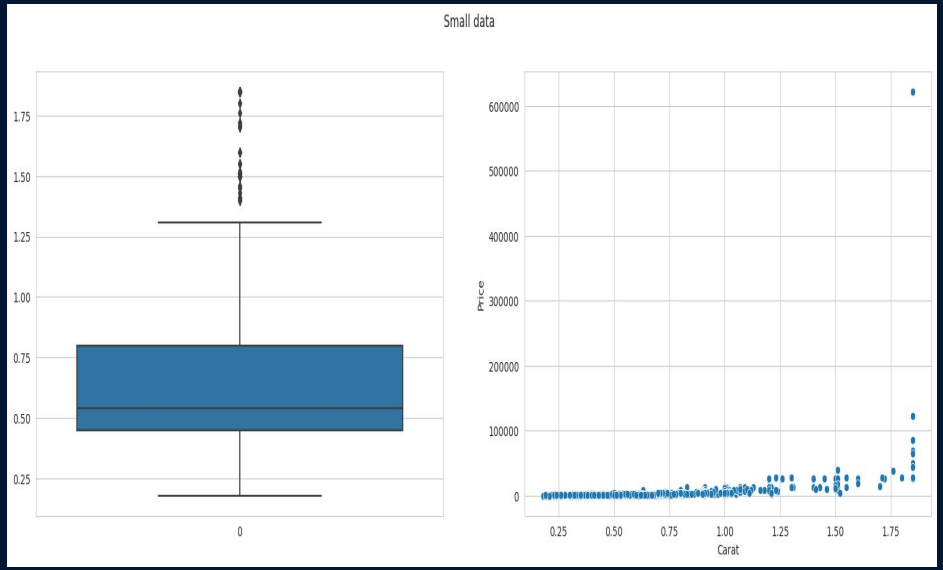


Phân bố dữ liệu của các biến categories trong tập dữ liệu lớn

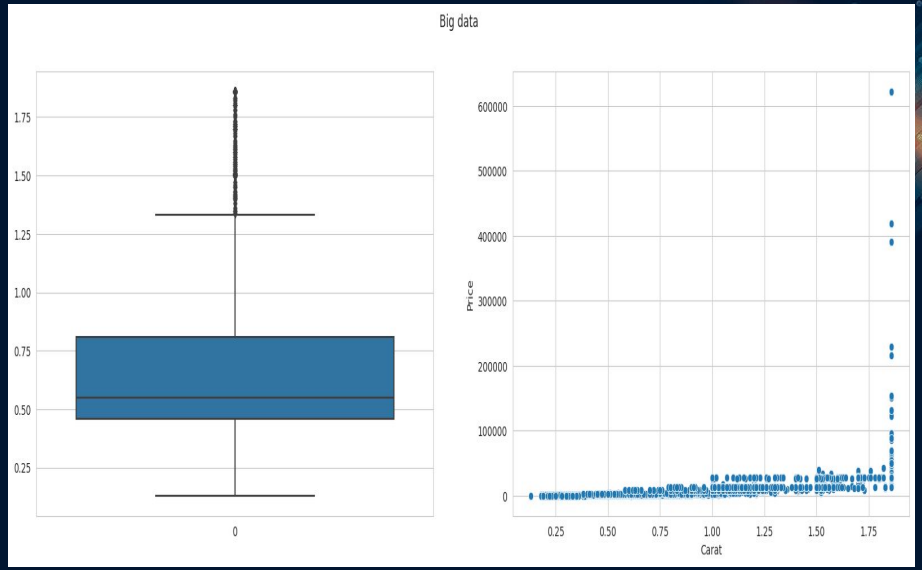
Phân bố dữ liệu của các biến numeric trong tập dữ liệu lớn

II. Trích xuất đặc trưng

- Carat và Price
  - Áp dụng xử lý ngoại lệ với phân bố lệch cho Carat

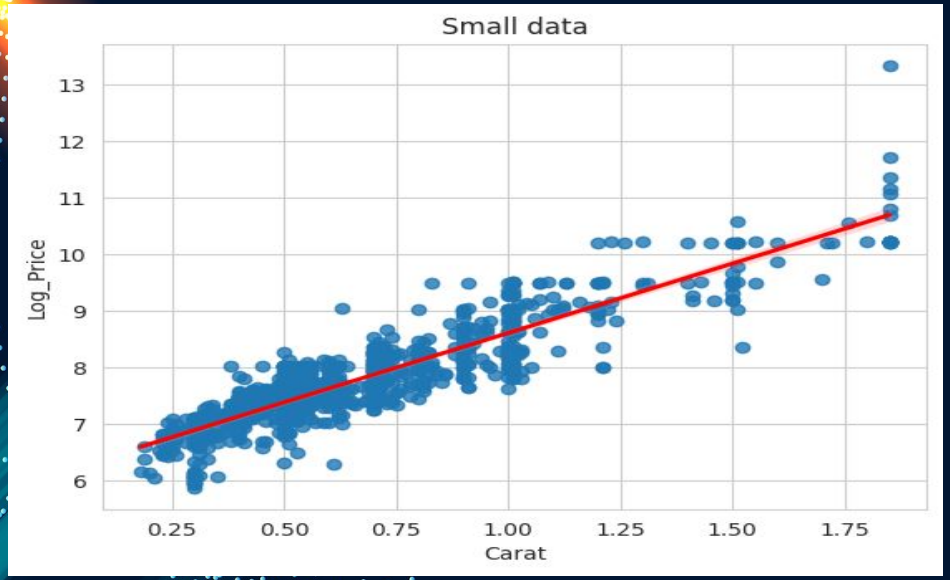


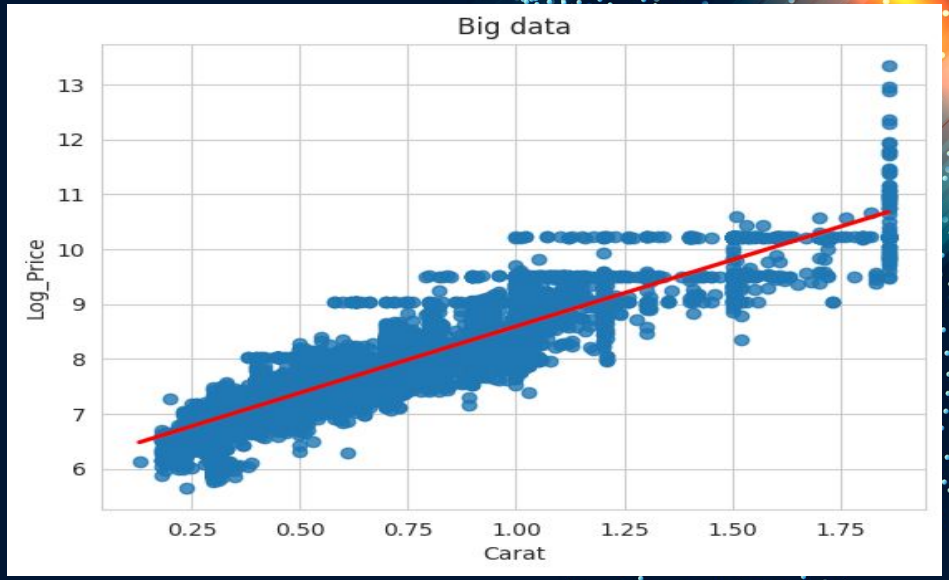
Small data



Big data

- Sau khi xử lý ngoại lệ thì vẫn còn nhiều điểm phân bố ở phía bên phải
- Thử quan sát Carat với Log_Price thì ta thấy Carat có tương quan tuyến tính với Price sau khi được Log

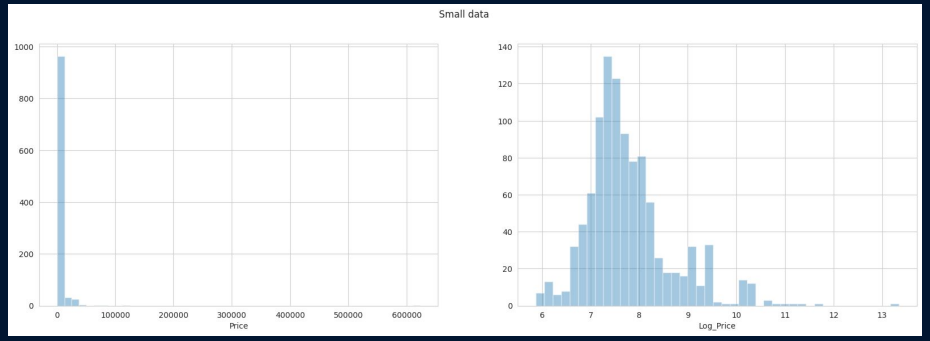


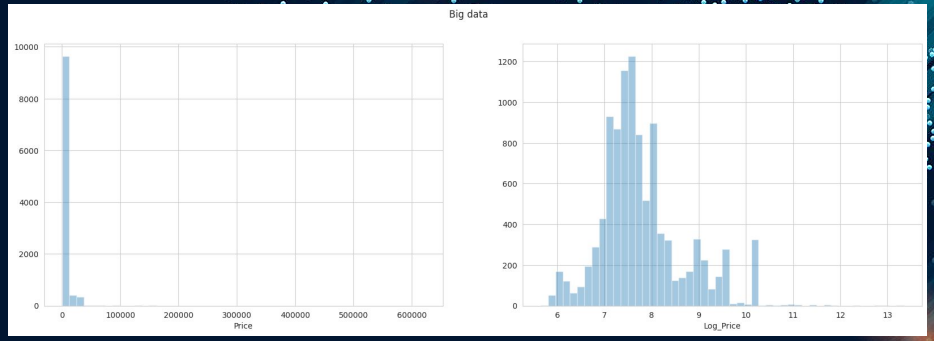


Small data

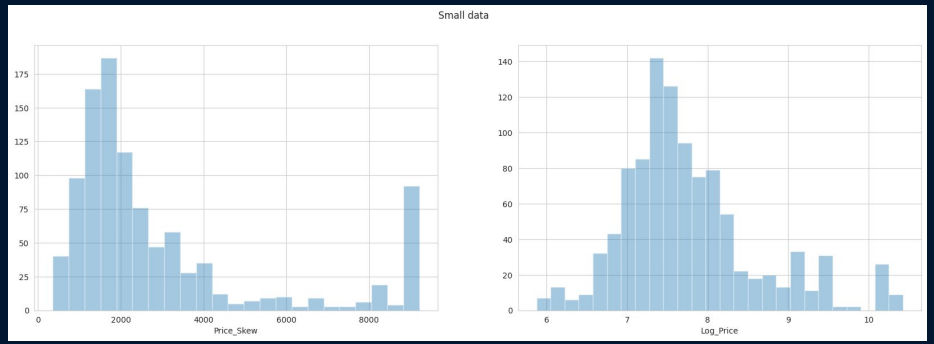
Big data

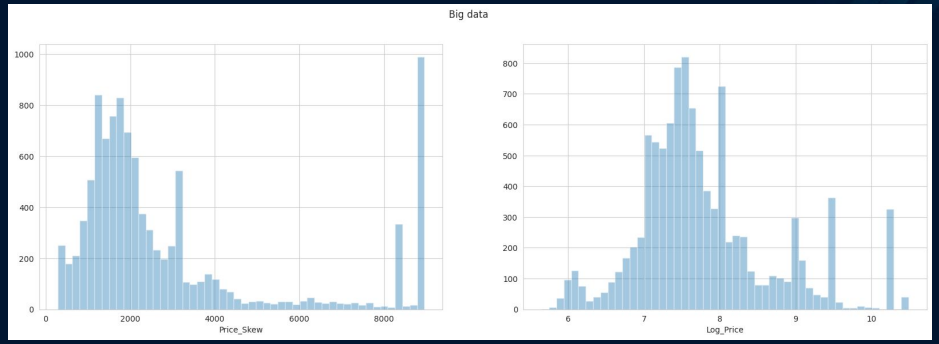
- Price và Log_Price





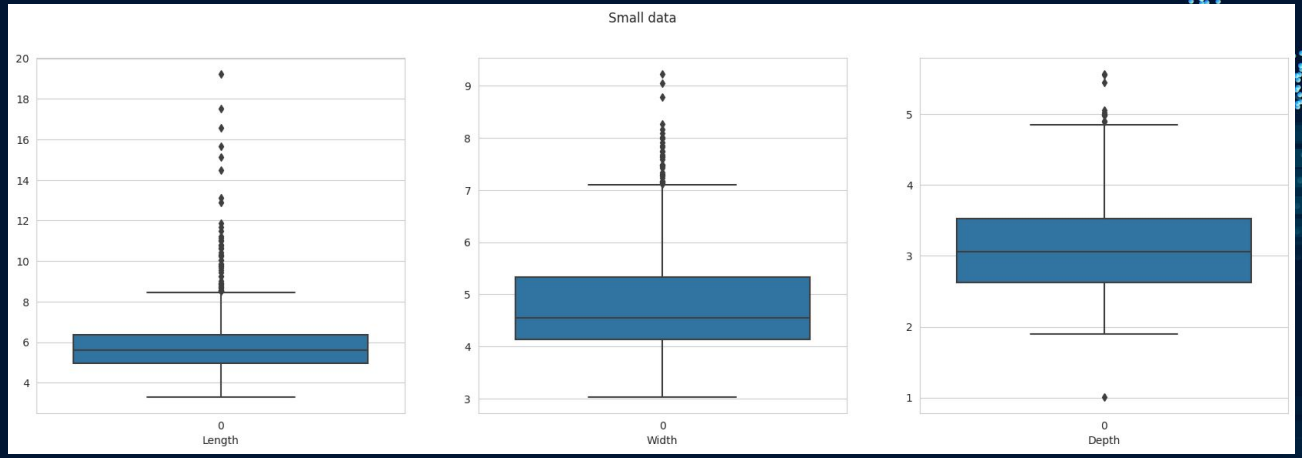
- Áp dụng xử lý với phân bố lệch cho Price và phân bố chuẩn cho Log_Price

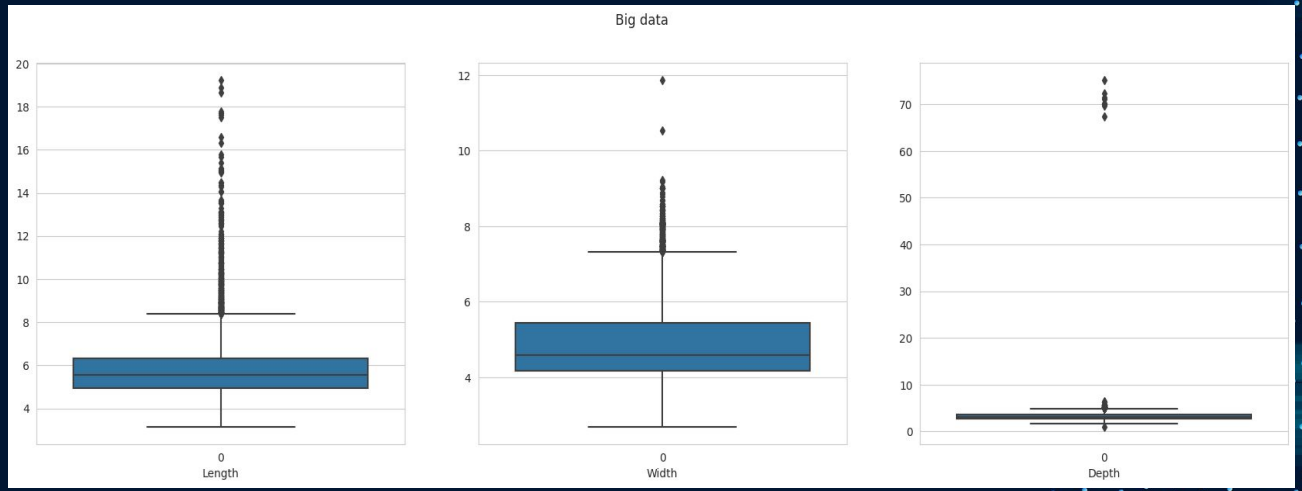




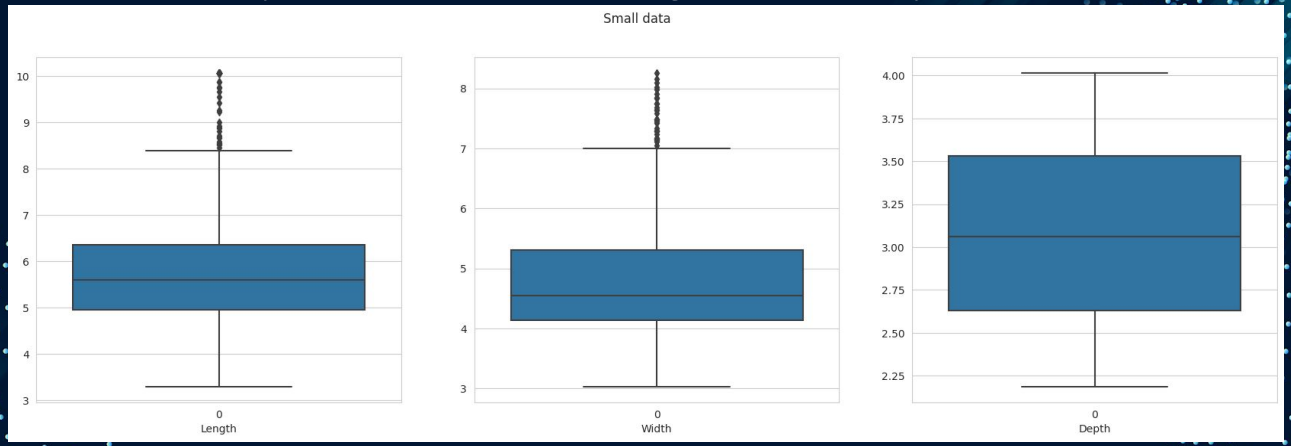
➢ Có thể thấy sau khi xử lý thì Log_Price có phân bố khá chuẩn, còn Price thì vẫn còn nhiều ngoại lệ ở bên phải

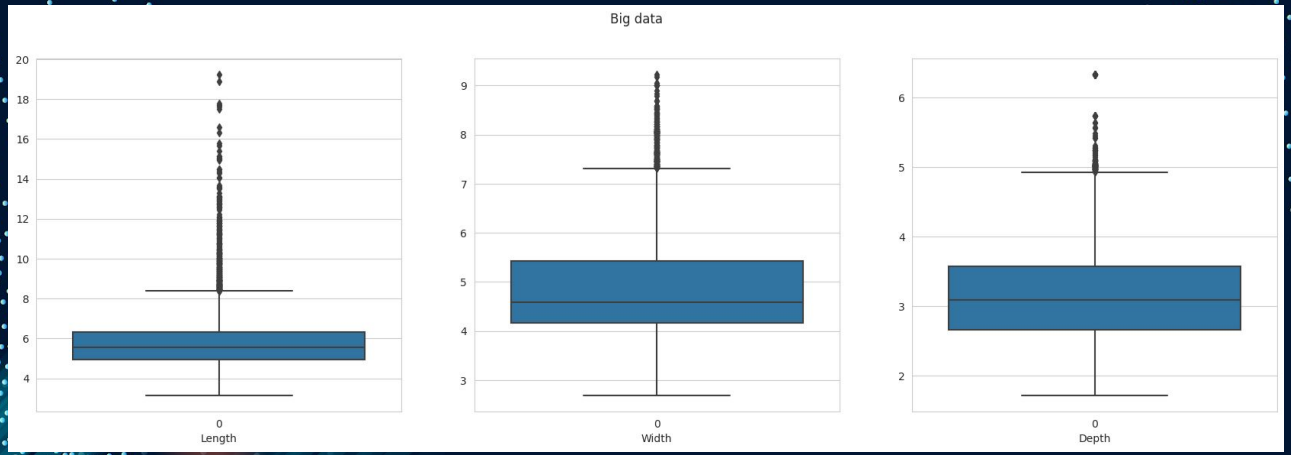
- Length, Width, Depth với Log_Price



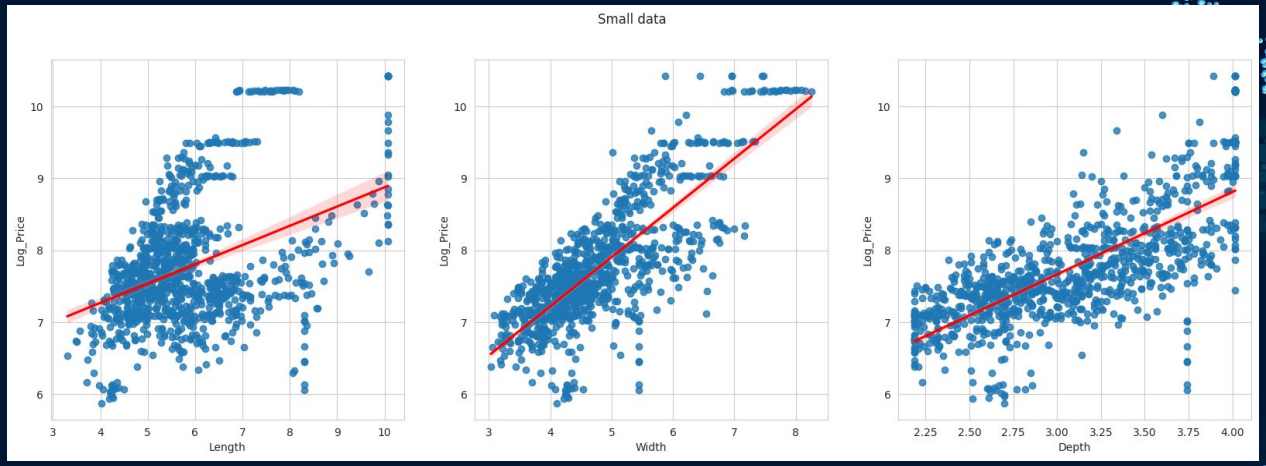


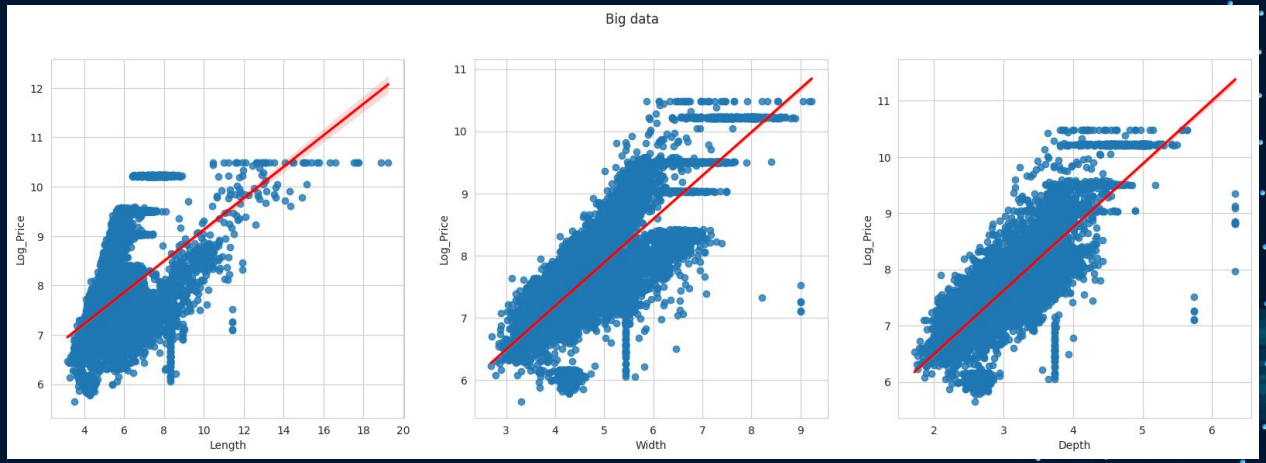
- Áp dụng các phương pháp xử lý ngoại lệ: drop các giá trị nằm xa phân bố, xử lý ngoại lệ cho phân bố lệch và phân bố chuẩn cho Length, Width, Depth



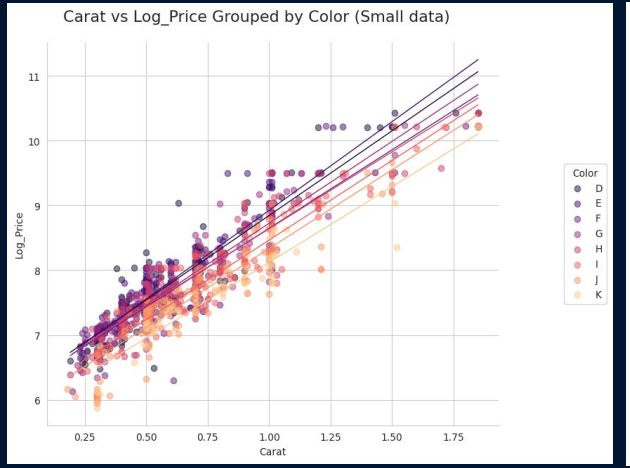


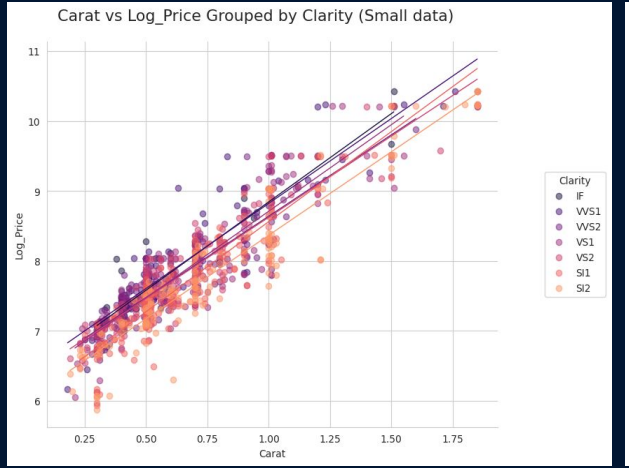
- Tương quan với Log_Price

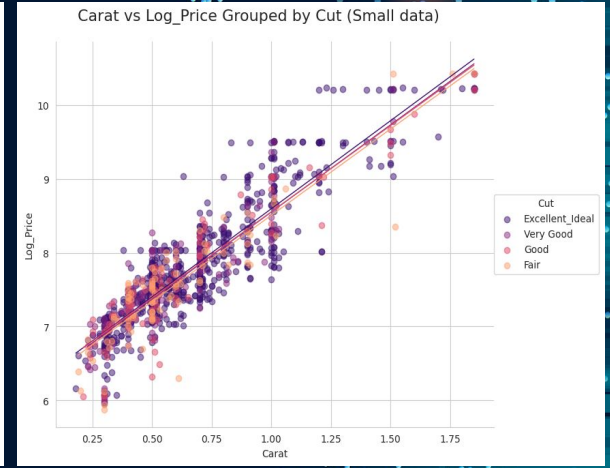


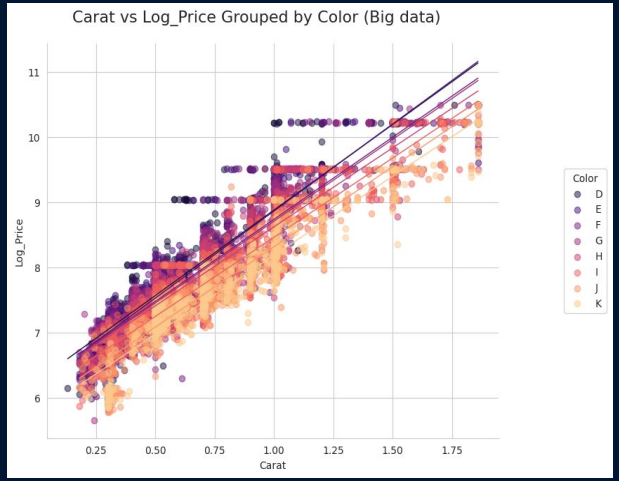


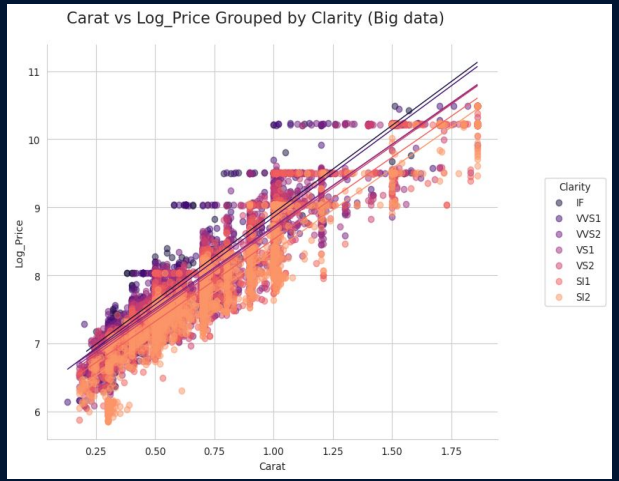
Cut, Color, Clarity, Carat và Log_Price

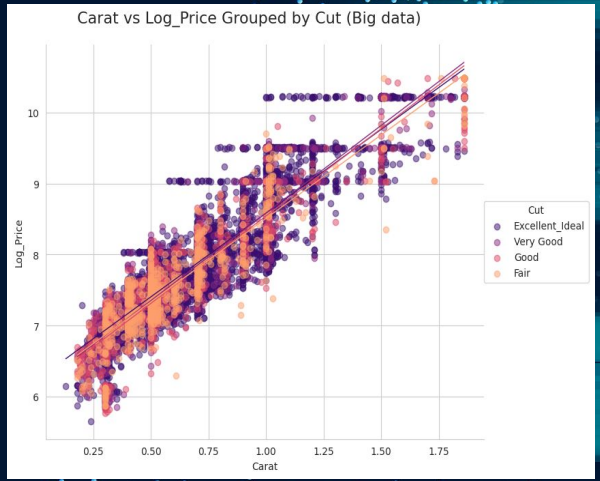




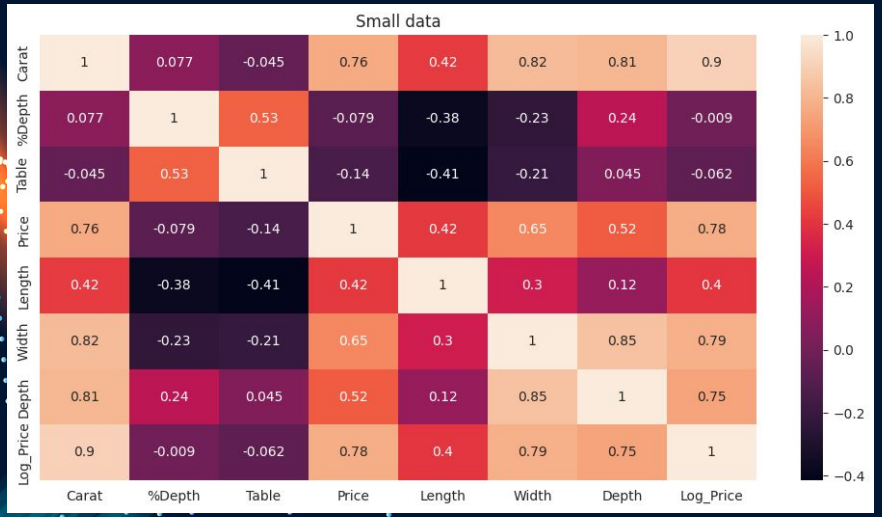








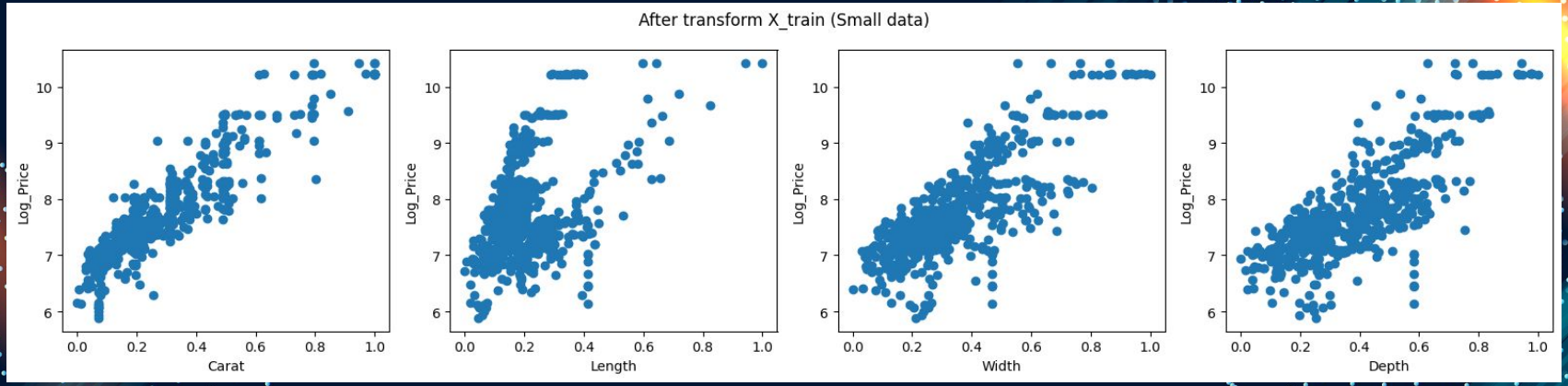
Feature selection (loại Price, %Depth và Table)

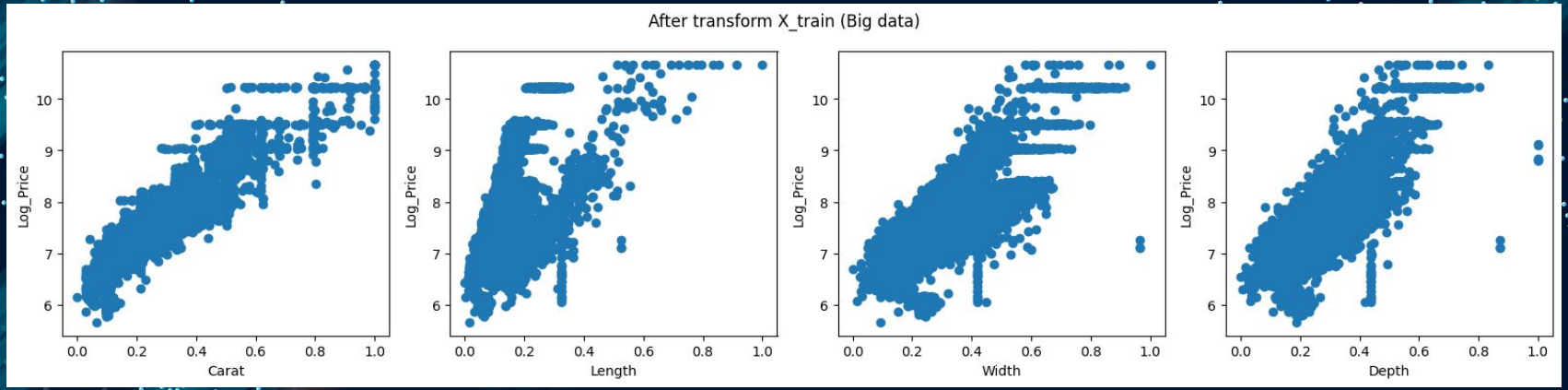


Label Encoder

	Shape	Carat	Color	Clarity	Cut	Fluorescence	Polish	Symmetry	Length	Width	Depth	Log_Price
0	1	1.01	4	6	0	0	0	0	6.43	6.47	3.950000	9.513182
1	1	1.40	0	4	0	1	0	0	7.13	7.16	4.048856	10.208248
2	1	1.85	7	5	0	0	0	0	8.20	8.26	4.048856	10.206920
3	1	0.30	5	7	0	1	0	0	4.31	4.34	2.610000	6.075346
4	1	0.56	3	4	0	0	0	0	5.23	5.27	3.260000	8.031060

Feature Transform (áp dụng Minmax Scaler)





III. Mô hình dự đoán

- Nhóm đã sử dụng hai mô hình để học và dự đoán giá kim cương là Linear Regression và Random Forest Regression.
- Chia dữ liệu train/ validation/ test bằng Stratified Sampling theo tỉ lệ 60/20/20.
- Trong mô hình, nhóm sẽ sử dụng 2 metrics là RMSE và R2 Score nhưng metrics chính để đánh giá là R2 Score vì nó khá tương đồng với Accuracy, giúp ta có cái nhìn trực quan hơn.
- Sau đó, nhóm sẽ thử nghiệm tìm kiếm bộ siêu tham số (Hyperparameter) tối ưu cho Random Forest Regression bằng RandomizedSearchCV và GridSearchCV.
- Thử nghiệm hiệu suất dự đoán với một số đặc trưng quan trọng với mô hình.

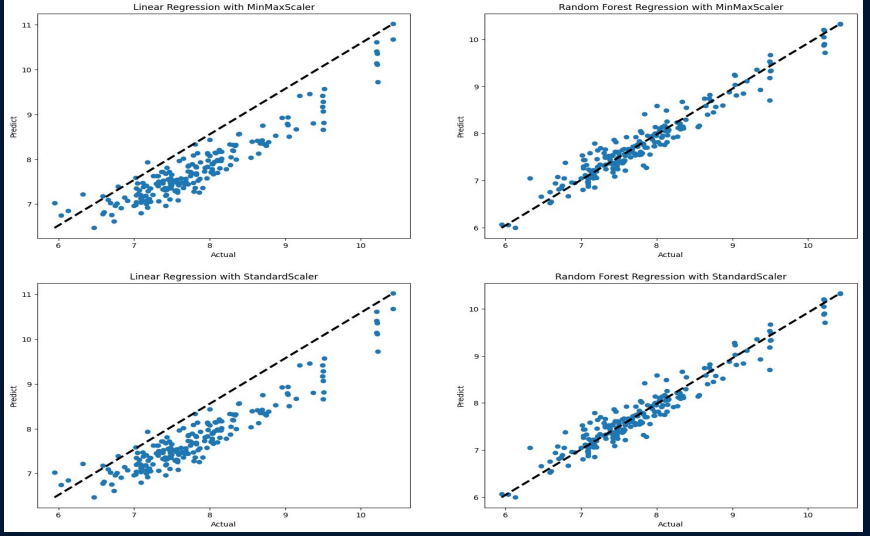
III. Mô hình dự đoán

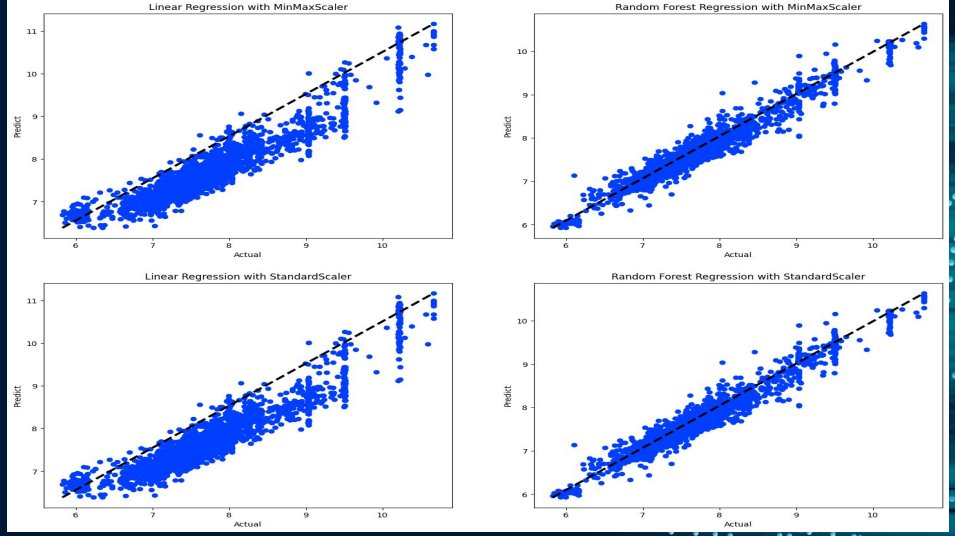
```
--- Training & Validating on Linear Regression model | MinMaxScaler---
- Time of fitting Linear Regression model: 0.04 (s)
  + RMSE on training: 0.30051670128262487
  + R2 on training: 0.8650069878283403
------------ Testing Linear Regression model | MinMaxScaler-----------
-> RMSE on testing: 0.29343077723259886
-> R2 on testing: 0.8809653361986646
======================================================================

--- Training & Validating on Random Forest Regression model | MinMaxScaler---
- Time of fitting Random Forest Regression model: 0.309 (s)
  + RMSE on training: 0.2169290113495003
  + R2 on training: 0.9296588707779418
------------ Testing Random Forest Regression model | MinMaxScaler-----------
-> RMSE on testing: 0.22646342282058876
-> R2 on testing: 0.929098037562513
======================================================================
```

Kết quả trên 1000 samples

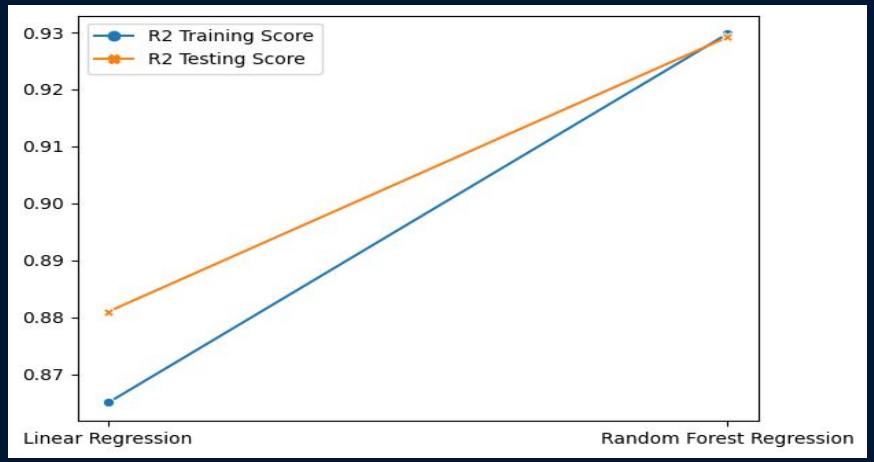
```
--- Training & Validating on Linear Regression model | MinMaxScaler---
- Time of fitting Linear Regression model: 0.013 (s)
  + RMSE on training: 0.3126950092012186
  + R2 on training: 0.8755453327676593
------------ Testing Linear Regression model | MinMaxScaler-----------
-> RMSE on testing: 0.3136070593650239
-> R2 on testing: 0.8762683510589288
======================================================================

--- Training & Validating on Random Forest Regression model | MinMaxScaler---
- Time of fitting Random Forest Regression model: 2.399 (s)
  + RMSE on training: 0.1753073020267567
  + R2 on training: 0.9608826612875282
------------ Testing Random Forest Regression model | MinMaxScaler-----------
-> RMSE on testing: 0.1782974732193012
-> R2 on testing: 0.9600055694698639
======================================================================
```

Kết quả trên 10000 samples

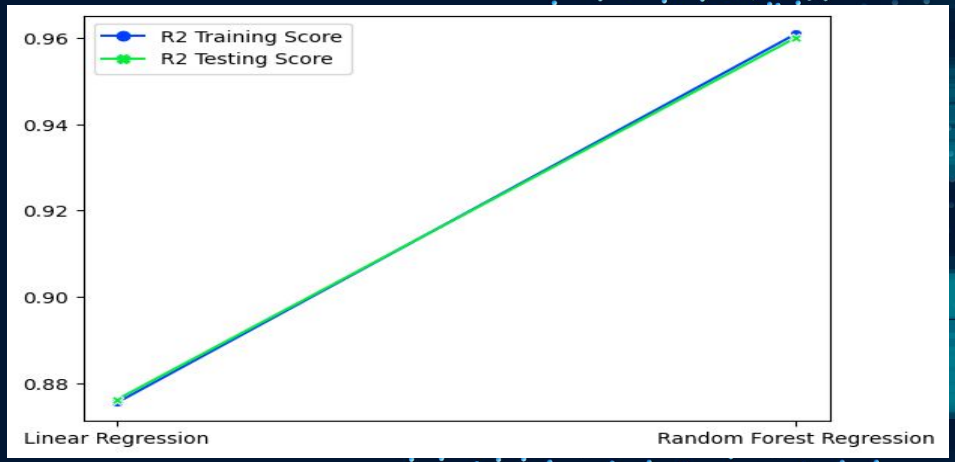




Kết quả trên 1000 samples

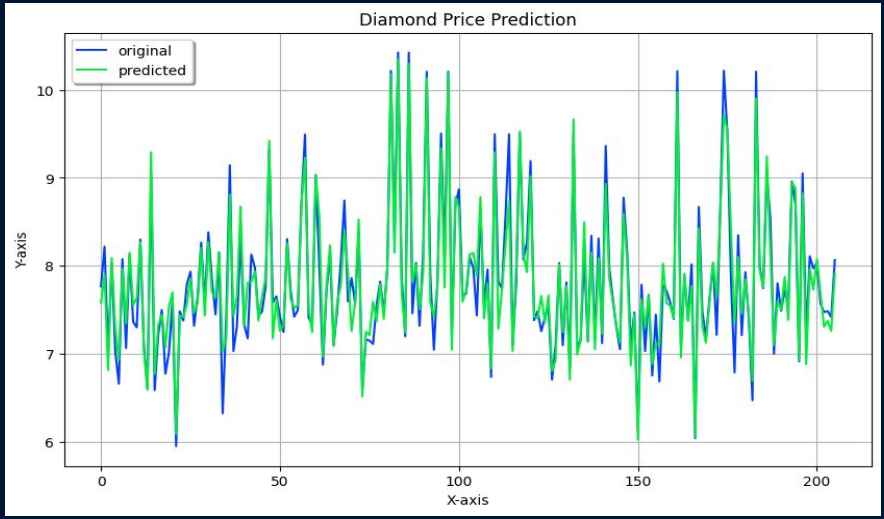


Kết quả trên 10000 samples



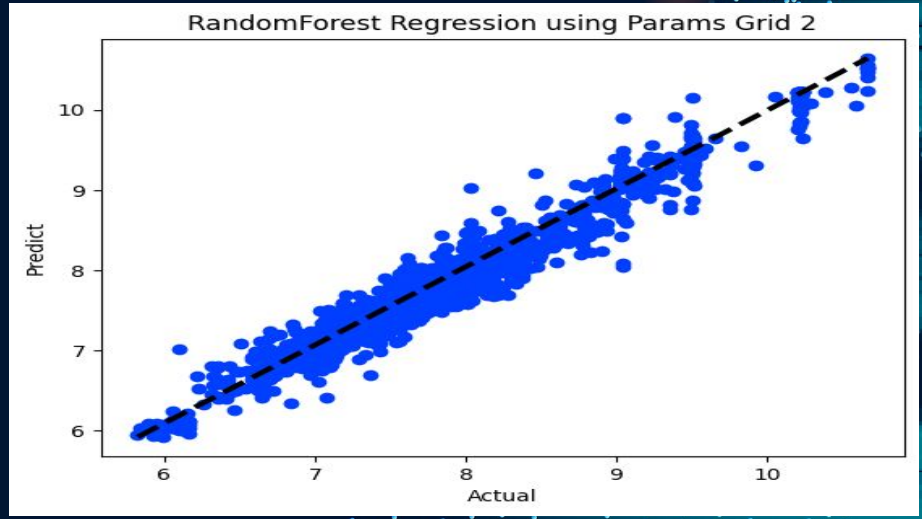
Kết quả sau khi áp dụng GridSearchCV và RandomizedSearchCV

```
Training & Validating on RandomForestRegressor(max_depth=15, max_features=10, n_estimators=200,
                  random_state=42) model
- Time of fitting RandomForestRegressor(max_depth=15, max_features=10, n_estimators=200,
                  random_state=42) model: 0.571 (s)
Testing RandomForestRegressor(max_depth=15, max_features=10, n_estimators=200,
                  random_state=42) model
-> RMSE on testing: 0.22424886778420744
-> R2 on testing: 0.9304779387892895
```

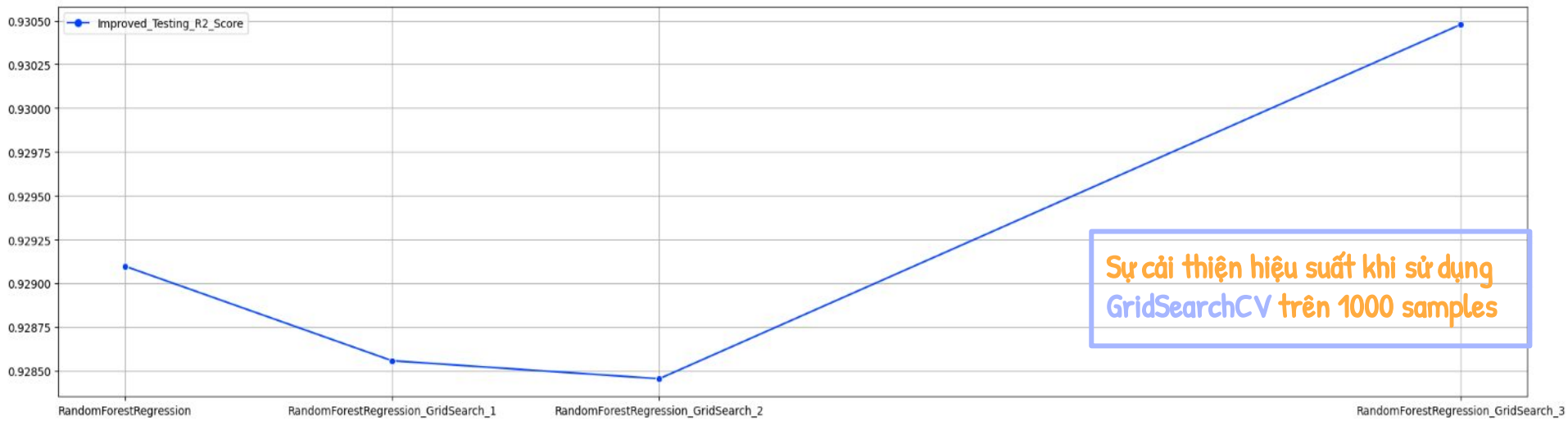


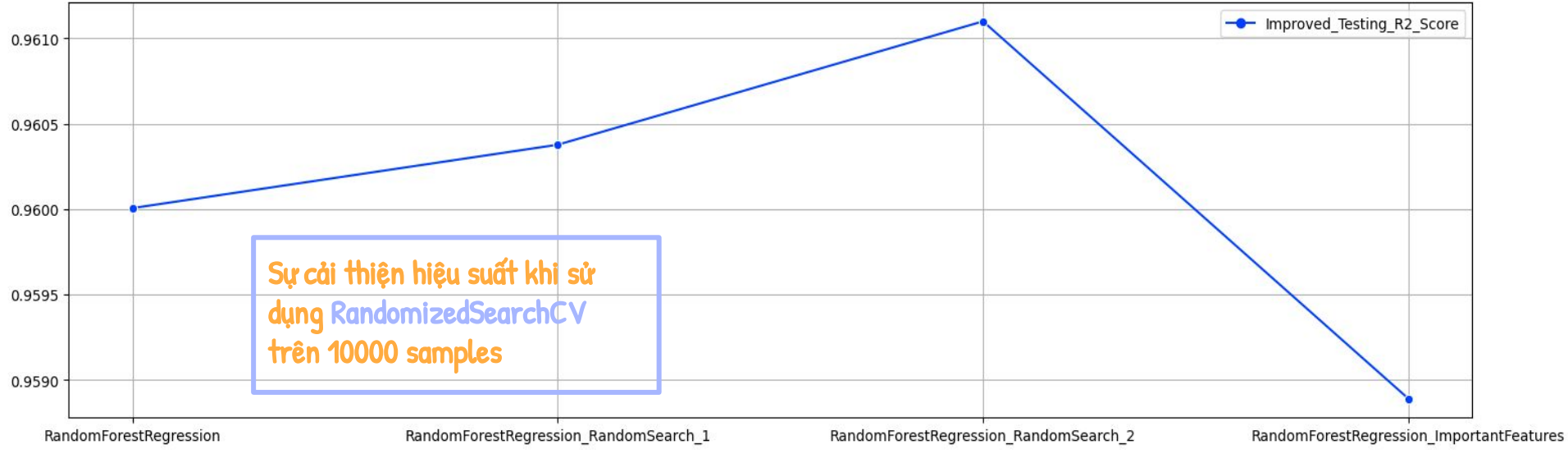
Kết quả trên 1000 samples

```
Training & Validating on RandomForestRegressor(max_depth=100, max_features=10, n_estimators=1400,
                  random_state=42) model
- Time of fitting RandomForestRegressor(max_depth=100, max_features=10, n_estimators=1400,
                  random_state=42) model: 34.698 (s)
Testing RandomForestRegressor(max_depth=100, max_features=10, n_estimators=1400,
                  random_state=42) model
-> RMSE on testing: 0.17584440954252717
-> R2 on testing: 0.961098506785092
```



Kết quả trên 10000 samples


Improved_Testing_R2_Score
0.93050
0.93025
0.93000
0.92975
0.92950
0.92925
0.92900
0.92875
0.92850
RandomForestRegression
RandomForestRegression_GridSearch_1
RandomForestRegression_GridSearch_2
RandomForestRegression_GridSearch_3
Sự cải thiện hiệu suất khi sử dụng GridSearchCV trên 1000 samples


Improved_Testing_R2_Score
0.9610
0.9605
0.9600
0.9595
0.9590
RandomForestRegression
RandomForestRegression_RandomSearch_1
RandomForestRegression_RandomSearch_2
RandomForestRegression_ImportantFeatures
Sự cải thiện hiệu suất khi sử dụng RandomizedSearchCV trên 10000 samples

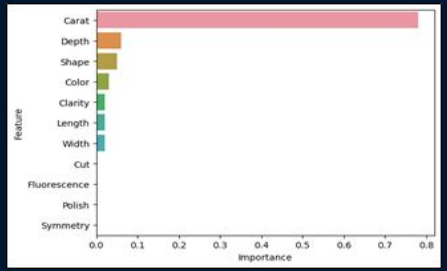
III. Mô hình dự đoán

Data size (samples)	Models	Train time (s)	R2 Score (%)
1.000	Linear Regression	0.040	88.096534
1.000	Random Forest Regression Base	0.040	92.909804
1.000	Random Forest Regression (RandomizedSearchCV)	4.524	92.803461
1.000	Random Forest Regression (GridSearchCV)	0.571	93.047794
10.000	Linear Regression	0.013	87.626835
10.000	Random Forest Regression Base	2.399	96.000557
10.000	Random Forest Regression (RandomizedSearchCV)	34.698	96.109851
10.000	Random Forest Regression (Important Features)	30.380	95.889092

Kết quả các mô hình trên 2 tập dữ liệu 1000 và 10000 samples

Nhận xét

- Linear Regression khi sử dụng ở 10.000 mẫu cho độ chính xác (R2 Score) thấp hơn khoảng 0.47% chính nó khi sử dụng ở 1.000 mẫu cụ thể là *87.63% so với 88.10%.*
- Random Forest Regression khi sử dụng ở 10.000 mẫu cho độ chính xác (R2 Score) cao hơn khoảng 3.09% chính nó khi sử dụng ở 1.000 mẫu cụ thể là *96.00% so với 92.91%.*
- Khi sử dụng RandomizedSearchCV và GridSearchCV cho Random Forest Regression:
  - *Tốn nhiều thời gian hơn so với mô hình Base* nhưng không phải lúc nào cũng cho kết quả tốt hơn.
  - Kích thước dữ liệu càng lớn thì thời gian train càng lâu và khi sử dụng RandomizedSearchCV để tìm bộ siêu tham số thì sẽ tốn thời gian hơn rất nhiều lần.
  - Ở tập dữ liệu 1.000 mẫu thì sử dụng GridSearchCV sẽ tìm bộ siêu tham số nhanh hơn và hiệu quả mô hình ứng với bộ siêu tham số đó cũng tốt hơn so với sử dụng RandomizedSearchCV. Thời gian giảm từ 3.55(s) -> 0.552(s) và độ chính xác tăng từ 92.80% -> 93.05%.
  - Ở tập dữ liệu 10.000 mẫu thì sử dụng RandomizedSearchCV sẽ có lợi hơn bởi nó tìm kiếm tốt trên dữ liệu lớn so với GridSearchCV Sau khi hiệu chỉnh bằng siêu tham số thì độ chính xác tăng lên nhưng không đáng kể khoảng 0.11% (96.00% -> 96.11%). Thời gian tăng lên rất nhiều từ 2.291(s) -> 38.469(s).
- Khi *sử dụng bộ 7 đặc trưng quan trọng nhất* để dự đoán thì độ chính xác giảm xuống còn 95.89% nhưng thời gian train cũng giảm theo từ 34.698(s) -> 30.380(s).



➢ Tóm lại: Mô hình cho dự đoán về giá kim cương khá chính xác với độ chính xác lên đến khoảng 96.1% với thời gian dự đoán cũng khá tốt khoảng 34.7 (s).

IV. Kết luận và Hướng phát triển

Thông qua đề tài lần này, các thành viên trong nhóm hiểu rõ hơn về môn Khoa học dữ liệu, đặc biệt là quá trình thu thập dữ liệu, trực quan hóa dữ liệu để lựa chọn các đặc trưng phù hợp nhất và sử dụng các thông số để đánh giá mô hình, từ đó đưa ra được kết quả tốt nhất qua các lần kiểm thử.

Qua quá trình thực hiện ở trên, có thể thấy được tất cả các bước trên đều quan trọng trong việc đưa ra mô hình dự đoán tối ưu nhất có thể. Có nhiều yếu tố ảnh hưởng đến tốc độ huấn luyện và độ chính xác làm mô hình còn một số hạn chế:

- Việc thu thập dữ liệu trên các trang web có thể gặp khó khăn do việc bảo mật của trang web hoặc do cùng một dữ liệu về giá kim cương thì thông tin chi tiết giữa các trang web lại khác nhau nên có thể ảnh hưởng đến việc áp dụng cho nhiều trang web.
- Đối với tập dữ liệu nhỏ hơn thì độ chính xác của mô hình thấp hơn so với khi sử dụng với tập dữ liệu lớn hơn. Tuy nhiên, trong một số trường hợp thì độ chính xác tăng không đáng kể mà còn làm cho mô hình trở nên phức tạp hơn và ảnh hưởng đến thời gian huấn luyện.

Từ đó, nhóm đưa ra một số giải pháp và hướng phát triển:

- Thu thập dữ liệu từ nhiều trang web khác nhau với số lượng mẫu có thể lớn hơn rất nhiều để có được tập dữ liệu tối ưu nhất có thể.
- Sử dụng các phương pháp tiền xử lý dữ liệu khác để có thể đưa ra được một mô hình dự đoán tốt hơn.
- Thử nghiệm với nhiều mô hình hơn như Decision Tree... để có kết quả và so sánh giữa các mô hình với nhau để lựa chọn mô hình tốt nhất.

THANK YOU