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

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

TÊN ĐỀ TÀI

DỰ ĐOÁN GIÁ ĐIỆN THOẠI CŨ

Nhóm	6
Họ và tên sinh viên	Lớp học phần
Võ Hoàng Bảo	20.15
Trương Bảo Ngọc	
Nguyễn Hoàng Quân	

Đà Nẵng, 05/2023

TÓM TẮT

Bài toán được đưa ra là Dự đoán giá điện thoại cũ, với dữ liệu đầu vào là các đặc trưng độc lập biểu hiện chi tiết về điện thoại cũ, yêu cầu cần đạt là dự đoán được giá cả của điện thoại đó. Dữ liệu cho mô hình được thu thập về từ trang web Chợ tốt kết hợp với GSMArena, sau đó liệu sẽ được làm sạch, chiết xuất và tiền xử lý. Cuối cùng được mô hình hóa với 2 thuật toán là Linear Regression và Random Forest Regression. 2 mô hình này sẽ được áp dụng để đánh giá độ hiệu quả của các kĩ thuật xử lý, sau đó sẽ được hiệu chỉnh các siêu tham số để cho ra mô hình tốt nhất phù hợp với dữ liệu đã thu thập được.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá theo 3 mức (Đã hoàn thành/Chưa hoàn thành/Không triển khai)
Trương Bảo Ngọc	- Crawl data - Mô tả dữ liệu	-Đã hoàn thành
Nguyễn Hoàng Quân	- Làm sạch dữ liệu - Trích xuất đặc trưng - Tiền xử lý đặc trưng	- Đã hoàn thành
Võ Hoàng Bảo	- Mô hình hóa dữ liệu - Kiểm thử tiền xử lý đặc trưng - Hiệu chỉnh mô hình	- Đã hoàn thành

MỤC LỤC

DANH SÁCH HÌNH ẢNH

CHƯƠNG 1: GIỚI THIỆU

1.1. Mở đầu

Trong thời địa công nghệ phát triển nhanh chóng, thị trường điện thoại cũ ngày càng trở nên sôi động. Việc mua bán điện thoại cũ không chỉ là một lựa chọn tiết kiệm cho người dung, mà còn là một cơ hội kinh doanh hấp dẫn cho các sàn giao dịch. Tuy nhiên, việc xác định giá trị thực của một chiệc điện thoại cũ trở thành một thách thức lớn do sự biến đổi giá cả không đồng đều.

Vấn đề đặt ra trong nghiên cứu này là làm thế nào để dự đoán giá điện thoại di động cũ một cách chính xác nhất. Để giải quyết vấn đề này, chúng ta phải xem xét các yếu tố quan trọng có ảnh hưởng đến giá điện thoại cũ, bao gồm tuổi điện thoại, cấu hình, tình trạng, xu hướng, .. Chúng ta cần phát triển một mô hình dự đoán giá đáng tin cậy, dựa trên dữ liệu thống kê và kỹ thuật học máy

CHƯƠNG 2: THU NHẬP VÀ MÔ TẢ DỮ LIỆU

2.1. Thu nhập dữ liệu

Nguồn dữ liệu:

https://www.chotot.com/mua-ban-dien-thoai-samsung-cu-sdmd2ec3

https://www.gsmarena.com/

https://www.chotot.com/mua-ban-dien-thoai-apple-cu-sdmd1ec3

Công cụ thu nhập:

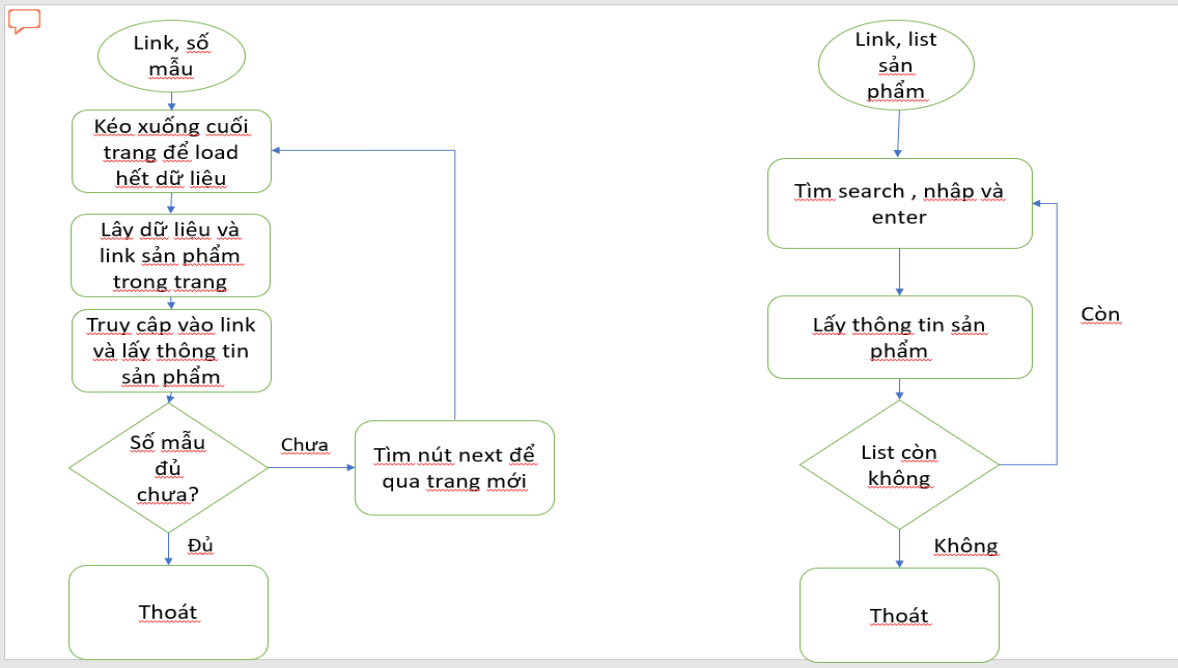
Ngôn ngữ Python và thư viện Selenium, BeautifulSoup

Cách thức sử dụng công cụ:

Selenium: Điều khiển và tương tác với trình duyệt web theo chương trình. Dùng Selenium để mở trình duyệt, điều hướng đến các trang web, điền vào biểu mẫu, nhấp chuột, kéo xuống trang và thu nhập dữ liệu

BeautifulSoup: Sử dụng để phân tích cú pháp HTML và XML. Dùng các phương thức lấy thông tin từ các phân tử HTML

Các bước thực hiện:



Hình 1: Các bước crawl

Bước 1: Lấy link từng sản phẩm , giá , nội dung và cá nhân hay cửa hàng(result=0 nếu cá nhân và ngược lại) như hình dưới

Bước 2: Truy cập vào từng link sản phẩm ta được hình dưới và lấy thông tin ram, bảo hành, tình trạng, loại sản phẩm, …

Bước 3: Tổng hợp tên dòng máy vào 1 list loại bỏ lặp và mở trang mới bắt đầu tìm kiếm

Bước 4: Nhập từng hang vào ô search nhấn enter và thực hiện lấy thông tin chi tiết như ram, camera, date release , …

SV mô tả các thống kê tổng quan về tập dữ liệu (số mẫu, số đặc trưng của 1 mẫu, kiểu dữ liệu của mỗi đặc trưng, số mẫu dữ liệu trống của mỗi đặc trưng,...) và xuất ra các thống kê mô tả trực quan về các đặc trưng (ví dụ: dùng boxplot, histogram, scatter plot về độ tương quan,…).

2.2. Mô tả dữ liệu

Các đặc trưng cơ bản của dữ liệu:

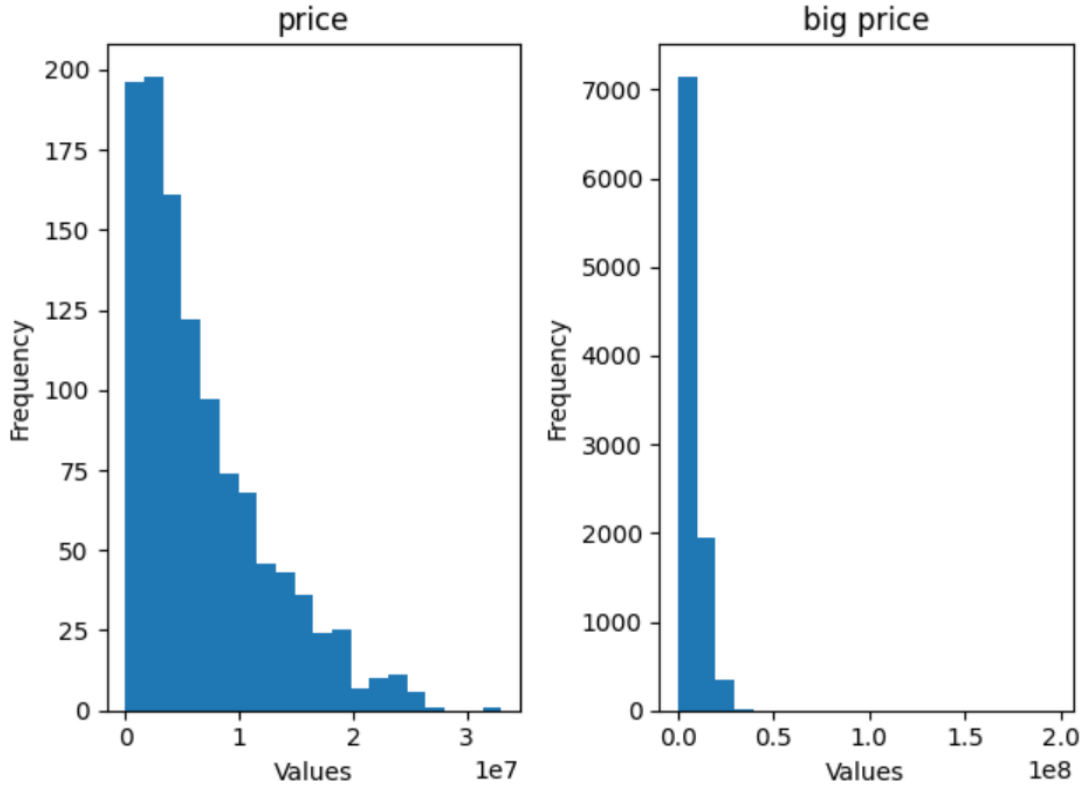
- Dữ liệu big data có 10350 mẫu và dữ liệu small data có 1150 mẫu bao gồm 13 đặc trưng

```
RangeIndex: 10350 entries, 0 to 10349
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   names        10350 non-null  object
 1   price        10350 non-null  object
 2   link         10350 non-null  object
 3   brand        10350 non-null  object
 4   type         10350 non-null  object
 5   condition    10350 non-null  object
 6   preserve     10350 non-null  object
 7   storage      10078 non-null  object
 8   result       10350 non-null  int64
 9   display      10323 non-null  object
 10  camera       10323 non-null  object
 11  pin          10323 non-null  object
 12  tag          10323 non-null  object
 13  date         10323 non-null  object
 14  ram_storage  10323 non-null  object
dtypes: int64(1), object(14)
```

```
names           0
price           0
link            0
brand           0
type            0
condition       0
preserve        0
storage       272
result          0
display        27
camera         27
pin            27
tag            27
date           27
ram_storage    27
dtype: int64
```

Hình 2: Thông tin dữ liệu

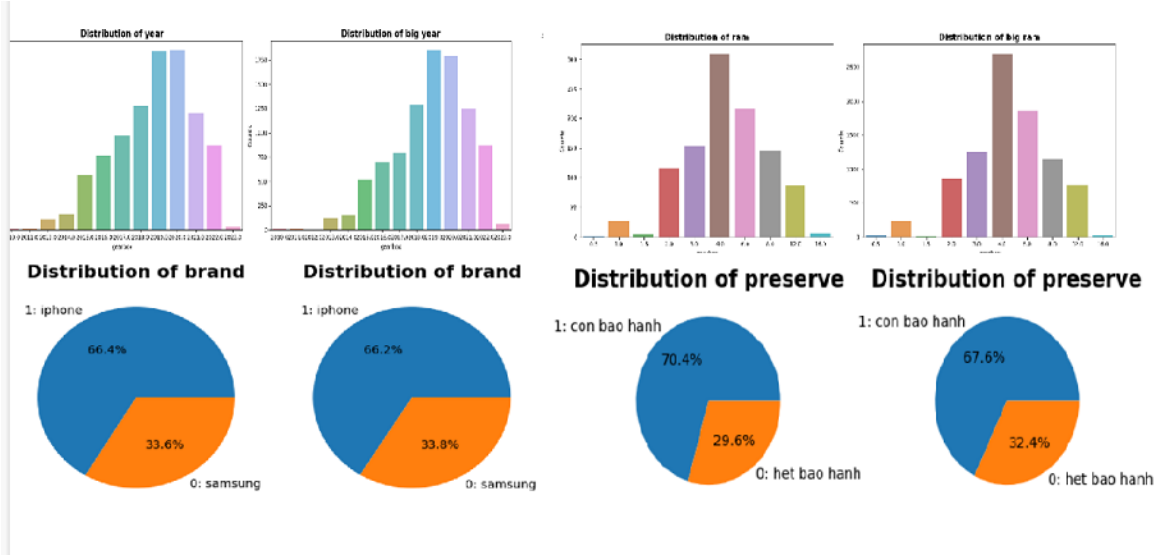
- Phân bố từng dự liệu với biến có nhiều values_count:



Hình 3: Đồ thị tần suất của biến price trong 2 dataset

Nhận xét:

- Price: Đa số dữ liệu đều nằm trong khoảng từ 1 đến 3 triệu. Ngoài ra bên dataset_10k có một số mẫu điện thoại có giá rất cao. Cao nhất trong tầm 20 triệu. Phân bố giảm dần theo chiều tăng của mức giá
- Pin: Dung lượng nằm trong khoảng từ 2000 đến 5000 mA . Nhiều nhất xung quanh khoảng 4000mA



Hình 4: Số lượng dữ liệu với từng loại trong từng đặc trưng

Nhận xét:

- Year: Dữ liệu đa số tập trung tại năm 2017 đến năm 2022, nhiều nhất là 2 năm 2019 và 2022.
- Ram: Máy có ram từ 2-8G có số lượng phổ biến nhất , và nhiều nhất là loại máy 4G
- Brand: dataset được chọn là Samsung va iphone trong đó hang iphone chiếm gấp đôi số lượng hãng Samung
- Condition: máy chưa qua sửa chữa chiếm phần lớn gấp khoảng 10 lần máy đã qua sửa chữa
- Preserve: máy hết bảo hành chiếm phầm lớn gần như gấp đôi số lượng máy còn bảo hành
- Storage: Dữ liệu gồm 6 thành phần , tập trung lớn nhất ở 64G 128G và 256G thấp nhất là 8G
- Result: Về mức độ chuyên nghiệp, sso máy được bán ra tập trung chủ yếu do cá nhân bán, gấp 4 lần so với cửa hàng hay doanh nghiệp
- Display: 24 loại màn hình , tập trung nhiều nhất ở 4.7 5.5 6.1 6.5 6.7 inch và cao nhất là 6.7 inch
- Camera: Chất lượng camera tập trung chủ yếu ở loại 12 pixel
- Ở 2 dataset khá giống nhau về phân bố số lượng dữ liệu của các đặc trưng

CHƯƠNG 3: TRÍCH XUẤT ĐẶC TRƯNG.

3.1. Làm sạch dữ liệu

Dữ liệu trước khi được làm sạch :.

	price	brand	type	condition	preserve	storage	result	display	camera	pin	date	ram_storage
0	9.900.000 đ	apple	iphone 12 pro	đã sử dụng (chưa sửa chữa)	đang cập nhật	128 gb	1	6.7"	12MP	3687mAh	Released 2020, November 13	128GB 6GB RAM, 256GB 6GB RAM, 512GB 6GB RAM
1	1500000	samsung	galaxy a30	đã sử dụng (chưa sửa chữa)	hết bảo hành	64.0 gb	0	6.4"	16MP	4000mAh	Released 2019, March	32GB 3GB RAM, 64GB 3GB RAM, 64GB 4GB RAM
2	8.500.000 đ	apple	iphone 11 pro max	đã sử dụng (chưa sửa chữa)	hết bảo hành	64.0 gb	0	6.5"	12MP	3969mAh	Released 2019, September 20	64GB 4GB RAM, 256GB 4GB RAM, 512GB 4GB RAM
3	11.000.000 đ	apple	iphone 12 pro	đã sử dụng (chưa sửa chữa)	đang cập nhật	128 gb	1	6.7"	12MP	3687mAh	Released 2020, November 13	128GB 6GB RAM, 256GB 6GB RAM, 512GB 6GB RAM
4	16.900.000 đ	samsung	galaxy z fold3	đã sử dụng (chưa sửa chữa)	còn bảo hành	256.0 gb	1	7.6"	12MP	4400mAh	Released 2021, August 27	256GB 12GB RAM, 512GB 12GB RAM
5	1500000	apple	iphone 6s	đã sử dụng (chưa sửa chữa)	còn bảo hành	64.0 gb	0	4.7"	12MP	1715mAh	Released 2015, September 25	16GB 2GB RAM, 32GB 2GB RAM, 64GB 2GB RAM, 128G...
6	5.500.000 đ	apple	iphone xr	đã sử dụng (chưa sửa chữa)	1 tháng	64 gb	0	6.1"	12MP	2942mAh	Released 2018, October 26	64GB 3GB RAM, 128GB 3GB RAM, 256GB 3GB RAM
7	4.500.000 đ	apple	iphone xs	đã sử dụng (chưa sửa chữa)	1 tháng	64 gb	0	6.5"	12MP	3174mAh	Released 2018, September 21	64GB 4GB RAM, 256GB 4GB RAM, 512GB 4GB RAM
8	260.000 đ	apple	iphone 5	đã sử dụng (chưa sửa chữa)	đang cập nhật	16.0 gb	0	4.0"	8MP	1560mAh	Released 2013, September 20	16GB 1GB RAM, 32GB 1GB RAM, 64GB 1GB RAM
9	9200000	apple	iphone 11 pro max	đã sử dụng (chưa sửa chữa)	hết bảo hành	64.0 gb	0	6.5"	12MP	3969mAh	Released 2019, September 20	64GB 4GB RAM, 256GB 4GB RAM, 512GB 4GB RAM

Hình 5: Dữ liệu trước khi làm sạch

- Tách cột ‘ram_storage’ thành 2 cột riêng ‘ram’ và ‘storage’.
- Tách lấy giá trị year từ cột ‘date’ và cho thành một cột mới là ‘year’.
- Loại bỏ các kí tự thừa trên cột ‘camera’, ‘pin’, ‘price’, ‘display’.
- Quy đổi 2 giá trị ‘apple’ và ‘samsung’ ở cột ‘brand’ thành lần lượt là 1 và 0.
- Quy đổi các giá trị như ‘đang cập nhật’, ‘hết bảo hành’ ở cột ‘preserve’ thành 0, còn lại là 1. Tương tự với cột ‘condition’ quy đổi chưa sửa chữa thành 1 và qua sửa chữa là 0.
- Bỏ đi các cột không cần thiết nữa là ‘names’, ‘link’, ‘tag’, ‘ram_storage’, ‘date’ (‘ram_storage’ và ‘date’ đã được tách ra thành những cột mới, còn ‘names’, ‘link’, ‘tag’ chỉ cho thêm thông tin về nhãn hiệu của điện thoại trong khi ‘brand’ và ‘type’ đã cho biết đủ thông tin đó).

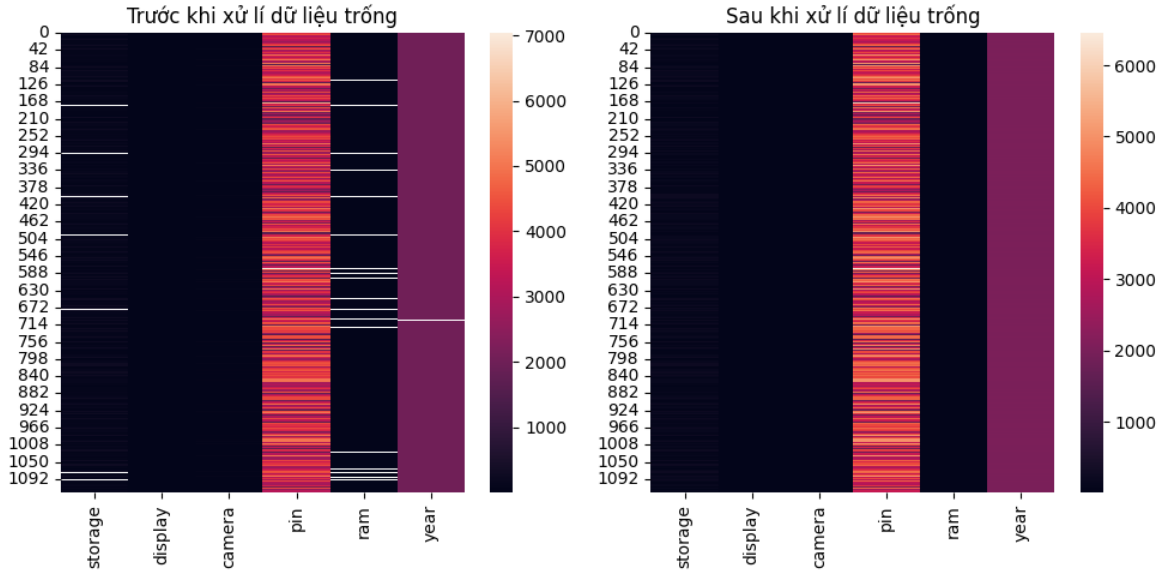
Dữ liệu sau khi đã được làm sạch :

	price	brand	type	condition	preserve	storage	result	display	camera	pin	ram	year
0	9900000.0	1	iphone 12 pro	1	0	128.0	1	6.7	12.0	3687.0	6.0	2020.0
1	1500000.0	0	galaxy a30	1	0	64.0	0	6.4	16.0	4000.0	4.0	2019.0
2	8500000.0	1	iphone 11 pro max	1	0	64.0	0	6.5	12.0	3969.0	4.0	2019.0
3	11000000.0	1	iphone 12 pro	1	0	128.0	1	6.7	12.0	3687.0	6.0	2020.0
4	16900000.0	0	galaxy z fold3	1	1	256.0	1	7.6	12.0	4400.0	12.0	2021.0
5	1500000.0	1	iphone 6s	1	1	64.0	0	4.7	12.0	1715.0	2.0	2015.0
6	5500000.0	1	iphone xr	1	1	64.0	0	6.1	12.0	2942.0	3.0	2018.0
7	4500000.0	1	iphone xs	1	1	64.0	0	6.5	12.0	3174.0	4.0	2018.0
8	260000.0	1	iphone 5	1	0	16.0	0	4.0	8.0	1560.0	1.0	2013.0
9	9200000.0	1	iphone 11 pro max	1	0	64.0	0	6.5	12.0	3969.0	4.0	2019.0

Hình 6: Dữ liệu sau khi được làm sạch và loại bỏ các chi tiết không cần thiết

3.2. Xử lý trống

- Thực hiện xử lí dữ liệu trống bằng cách điền các giá trị random từ các cột có dữ liệu trống là ‘storage’, ‘display’, ‘camera’, ‘pin’, ‘ram’, ‘year’ (các giá trị của các cột đó là những giá trị thông số kĩ thuật cố định nên ta phải dùng giá trị random từ cột dữ liệu)
- Sử dụng heatmap để thể hiện sự thay đổi sau khi xử lí dữ liệu trống :

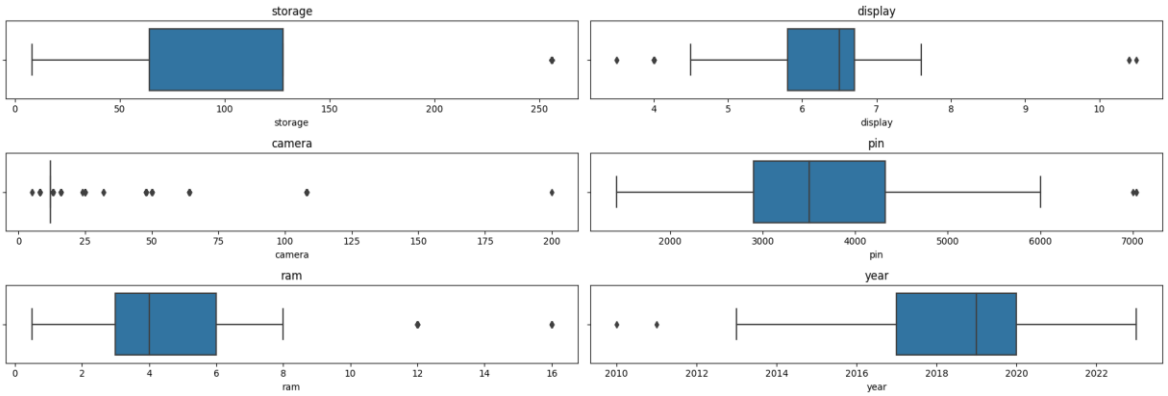


Hình 7: Heatmap của dữ liệu trước và sau khi xử lí dữ liệu trống

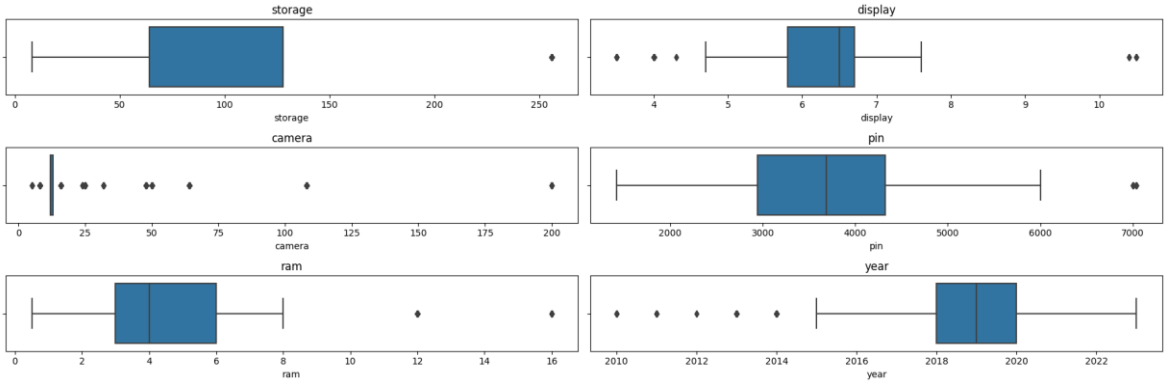
⇨ Các vạch trắng thể hiện các giá trị null trên dữ liệu và sau khi được xử lí thì các vạch trắng đã biến mất, dữ liệu đã được xử lí trống.

3.3. Xử lý ngoại lệ

Phân bố dữ liệu của các đặc trưng trước khi xử lí ngoại lệ :



Hình 8: Phân bố dữ liệu của các đặc trưng trước xử lí ngoại lệ ở tập dữ liệu nhỏ

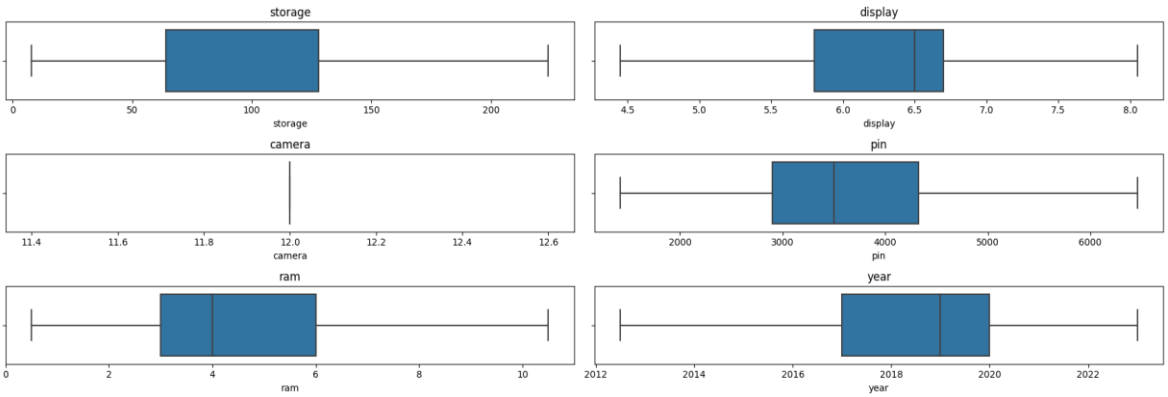


Hình 9: Phân bố dữ liệu của các đặc trưng trước xử lí ngoại lệ ở tập dữ liệu lớn

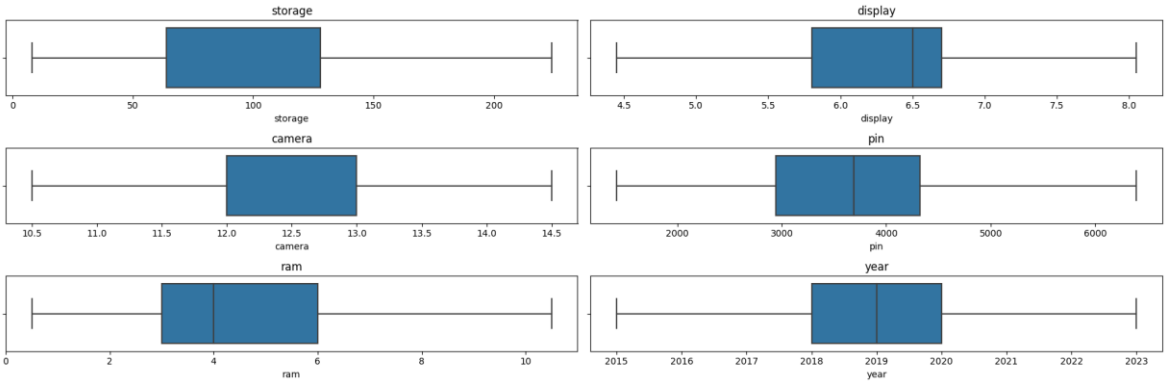
Nhận xét :

- Giá trị của các đặc trưng không liên tục mà bị rời rạc.
- Đặc trưng camera có dữ liệu không phân tán, chủ yếu các giá trị tập trung ở 12-13 MP và xuất hiện nhiều ngoại lệ.
- Đối với các đặc trưng như storage, pin, ram các ngoại lệ nằm nhiều ở bên phải trong khi ngoại lệ của year nằm ở bên trái.

Phân bố của dữ liệu sau khi áp dụng kĩ thuật xử lí ngoại lệ Skewed



Hình 10: Phân bố dữ liệu của các đặc trưng sau xử lí ngoại lệ ở tập dữ liệu nhỏ



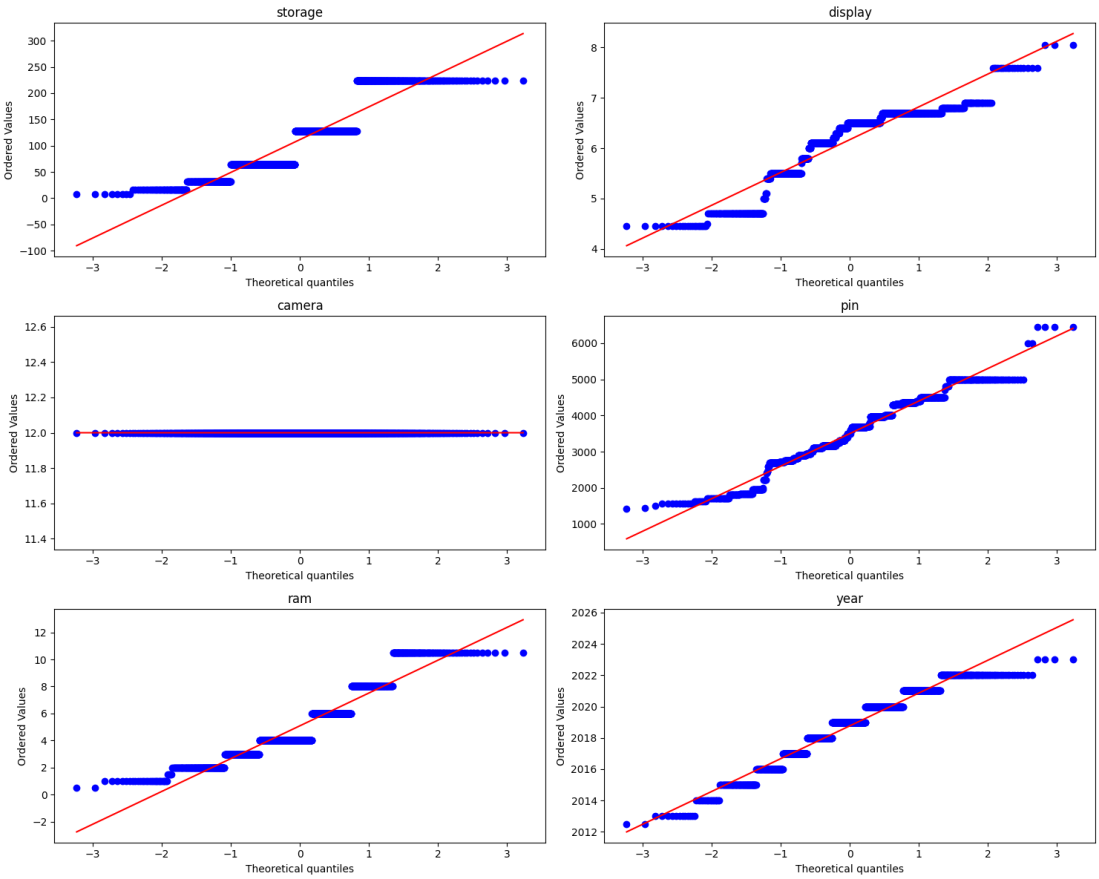
Hình 11: Phân bố dữ liệu của các đặc trưng sau xử lí ngoại lệ ở tập dữ liệu lớn

Nhận xét :

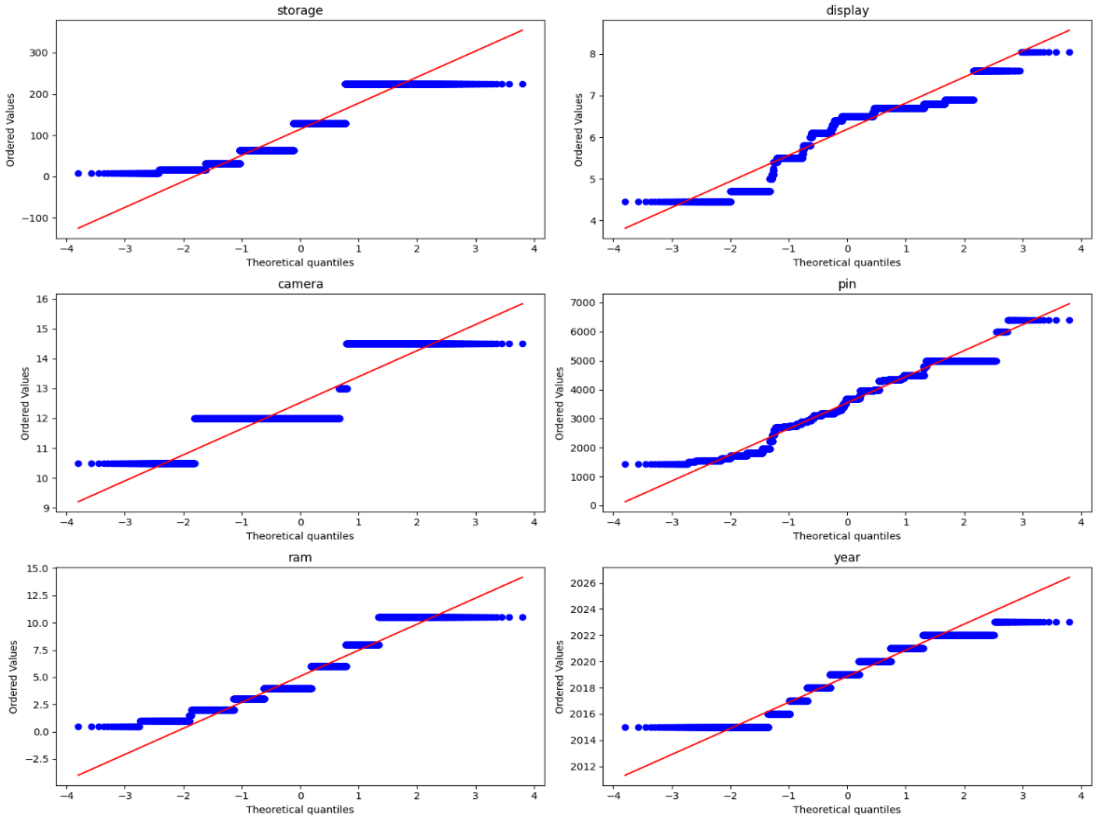
- Dữ liệu ở các đặc trưng đã ít bị lệch đi so với trước khi xử lý và ngoại lệ cũng đã được xử lý hết. Tuy nhiên ở camera các giá trị vẫn bị tập trung và không có sự phân tán.

3.4. Chuẩn hóa đặc trưng

Probability plot trước khi áp dụng Normalizer :

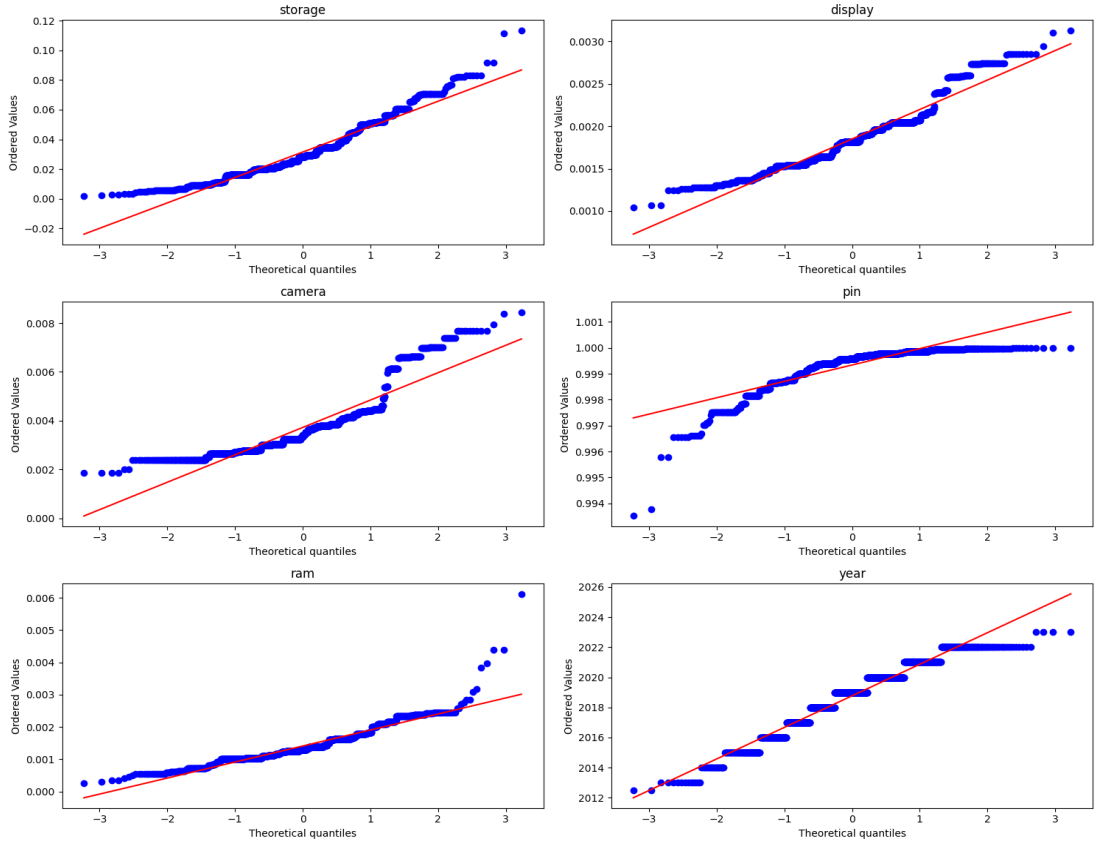


Hình 12: Prob plot của các đặc trưng trước khi chuẩn hóa ở tập dữ liệu nhỏ

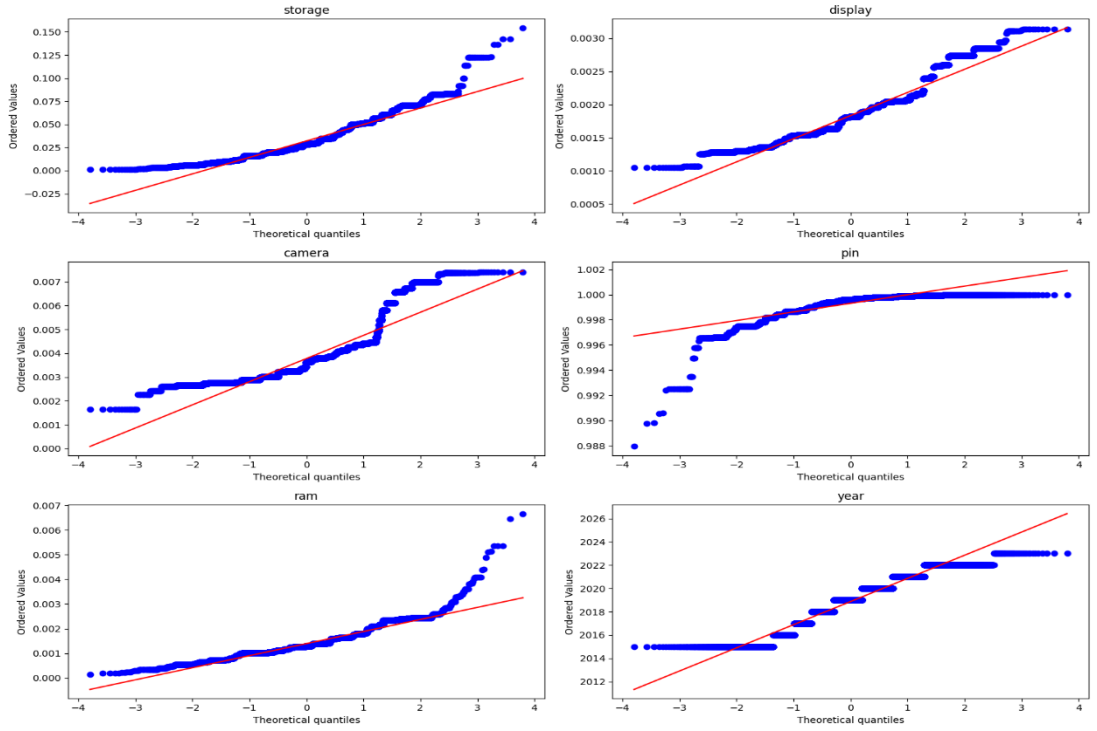


Hình 13: Prob plot của các đặc trưng trước khi chuẩn hóa ở tập dữ liệu lớn

- Áp dụng Normalizer scaler cho 6 đặc trưng không có giá trị binary : storage, display, camera, pin, ram, year.
- Sau khi áp dụng Normalizer :



Hình 14:Prob plot của các đặc trưng sau khi chuẩn hóa ở tập dữ liệu nhỏ



Hình 15: Prob plot của các đặc trưng sau khi chuẩn hóa ở tập dữ liệu lớn

Nhận xét :

- Đối với đặc trưng camera, trước khi dùng Normalizer dữ liệu phân phối cho thấy camera có nhiều giá trị giống nhau lặp lại nhưng sau khi chuẩn hóa bằng Normalizer dữ liệu của camera đã được trải đều trong khoảng giá trị nhất định. Tương tự các đặc trưng khác cũng đã tuân theo phân phối chuẩn sau khi dùng Normalizer.

3.5. Lựa chọn đặc trưng

Sử dụng ma trận tương quan để thể hiện độ tương quan giữa các đặc trưng đối với price và sau đó dùng heatmap để trực quan sự tương quan đó :

Hình 16: Ma trận tương quan giữa các đặc trưng

Nhận xét:

- Qua đó ta thấy các đặc trưng có tương quan lớn với price là year.
- Các đặc trưng có tương quan thấp là condition và result.
- Các đặc trưng còn lại phần lớn đều nằm trong khoảng 0.2 – 0.3, vì vậy chọn ta chọn ngưỡng 0.2 nhằm lấy vừa đủ các đặc trưng quan trọng.

CHƯƠNG 4: MÔ HÌNH HÓA DỮ LIỆU.

4.1. Lựa chọn mô hình

Lựa chọn mô hình là một quá trình quan trọng trong việc dự đoán, và có nhiều yếu tố cần xem xét để chọn được mô hình phù hợp. Với bài toán Dự đoán giá điện thoại cũ, là một bài toán dự đoán giá trị liên tục. Vì vậy ở đây liệt kê ra 2 mô hình phù hợp với việc dự đoán các giá trị liên tục từ các đặc trưng đầu vào

- Linear Regression: Linear Regression là một mô hình hồi quy đơn giản trong việc học máy. Nó là một phương pháp thống kê để xác định mối quan hệ tuyến tính giữa các biến đầu vào (độc lập) và biến đầu ra (phụ thuộc)

- Random Forest Regression: Random Forest Regression là một mô hình dựa trên nguyên tắc "rừng ngẫu nhiên". Nó kết hợp nhiều cây quyết định (decision tree) trong quá trình huấn luyện và tạo ra sự đa dạng trong dự đoán. Điều này giúp giảm thiểu hiện tượng overfitting và cải thiện khả năng dự đoán tổng quát của mô hình

4.2. Huấn luyện mô hình

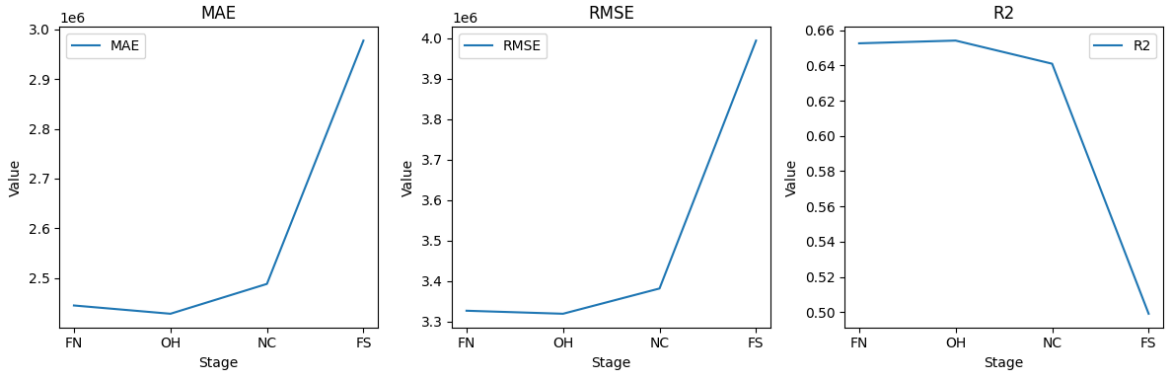
Đầu tiên các bộ dataset sau khi được xử lý sẽ chia được chia thành dữ liệu thành tập Train và tập Test theo tỉ lệ 7: 3. Sau đó từng bộ dataset theo mỗi stage tiền xử lý là lọc giá trị trống, lọc ngoại lệ, chuẩn hóa và lựa chọn đặc trưng sẽ được fit vào và kiểm thử ở từng mô hình để đánh giá mức ảnh hưởng của các kĩ thuật trên đối với việc mô hình hóa dữ liệu, các độ đo được áp dụng trong mô hình là MAE, RMSE và R2:

Với MAE chỉ trị tuyệt đối sai lệch giữa dự đoán và thực tế

Với RMSE chỉ căn bậc hiệu bình phương sai lệch giữa dự đoán và thực tế

Với R2 chỉ mức độ tương quan giữa mô hình và biến giá trị đích(giá tiền)

Ở BigDS:



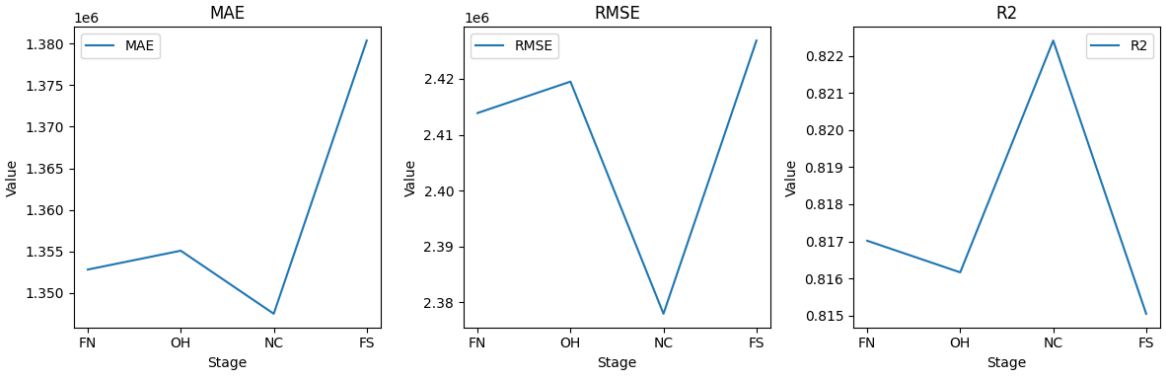
```
Linear Regression
Best metrics value: MAE: 2428231.004813155 RMSE: 3318906.701552667 R2: 0.65408174167379
```

Hình 17: Bộ đo mô hình kiểm thử tiền xử lý Linear Regression BigDS

Nhận xét độ quan trọng của các quá trình xử lý dữ liệu trên mô hình hóa Linear Regression ở BigDS:

- Mô hình được cải thiện qua các bước: Outlier Handling
- Mô hình trở nên kém đi qua các bước: Normalizer Scaling, Feature Selection

-> Các kĩ thuật cải thiện hiệu quả của thuật toán là: **Outlier Handling**




```
Random Forest Regression
Best metrics value: MAE: 1347485.8191810192 RMSE: 2377989.52640211 R2: 0.8224162296105847
```

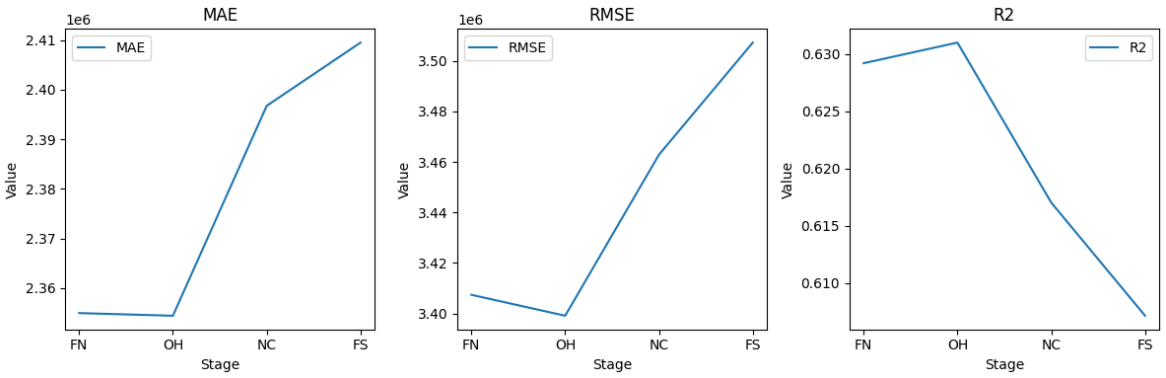
Hình 18: Bộ đo mô hình kiểm thử tiền xử lý Random Forest Regression BigDS

Nhận xét độ quan trọng của các quá trình xử lý dữ liệu trên mô hình hóa Random Forest Regression ở BigDS:

- Mô hình được cải thiện qua các bước: Normalizer Scaling
- Mô hình trở nên kém đi qua các bước: Outlier Handling, Feature Selection

-> Các kĩ thuật cải thiện hiệu quả của thuật toán là: **Normalizer Scaling**

Ở SmallDS:



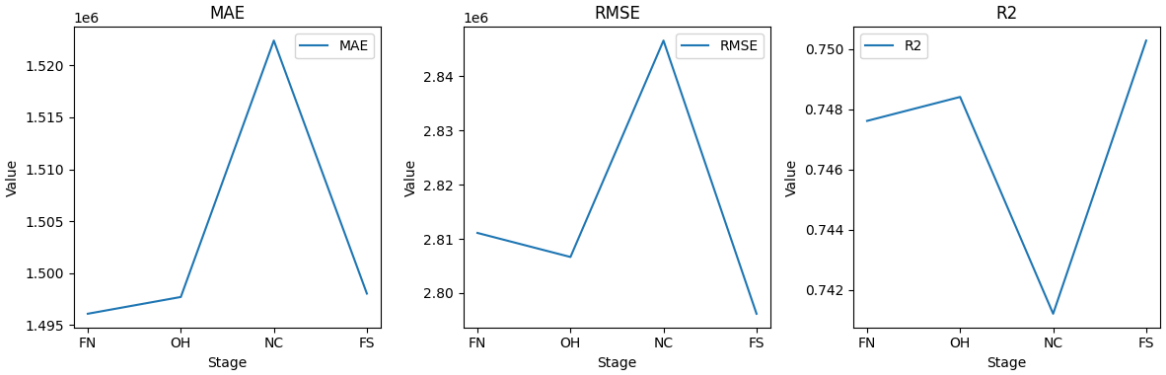
```
Linear Regression
Best metrics value: MAE: 2354408.8512621955 RMSE: 3399076.0944626066 R2: 0.630994536619345
```

Hình 19: Bộ đo mô hình kiểm thử tiền xử lý Linear Regression SmallDS

Nhận xét độ quan trọng của các quá trình xử lý dữ liệu trên mô hình hóa Linear Regression ở SmallDS:

- Mô hình được cải thiện qua các bước: Outlier Handling
- Mô hình trở nên kém đi qua các bước: Normalizer Scaling, Feature Selection

-> Các kĩ thuật cải thiện hiệu quả của thuật toán là: **Outlier Handling**




```
Random Forest Regression
Best metrics value: MAE: 1496081.5533366832 RMSE: 2796190.0746765975 R2: 0.7502851316728109
```

Hình 20: Bộ đo mô hình kiểm thử tiền xử lý Random Forest Regression SmallDS

Nhận xét độ quan trọng của các quá trình xử lý dữ liệu trên mô hình hóa Random Forest Regression ở SmallDS:

- Mô hình được cải thiện qua các bước: Outlier Handling, Feature Selection
- Mô hình trở nên kém đi qua các bước: Normalizer Scaling

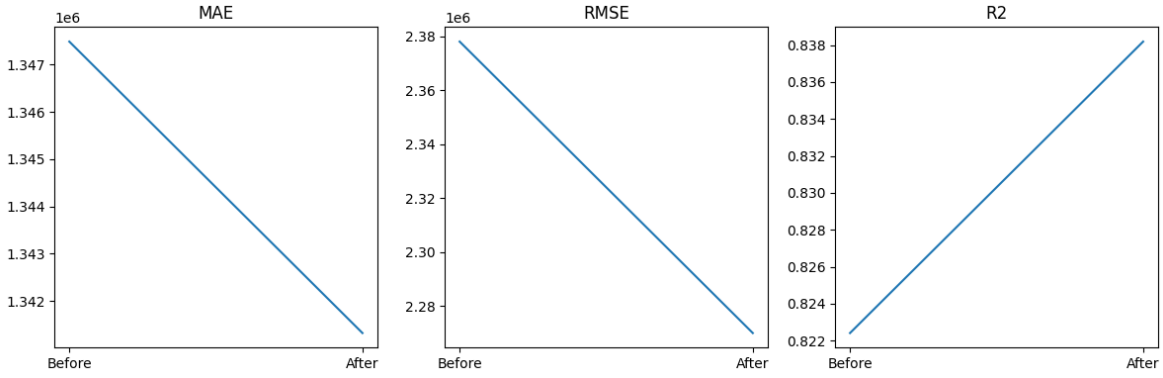
-> Các kĩ thuật cải thiện hiệu quả của thuật toán là: **Outlier Handling, Feature Selection**

4.3. HyperParameter Tunning

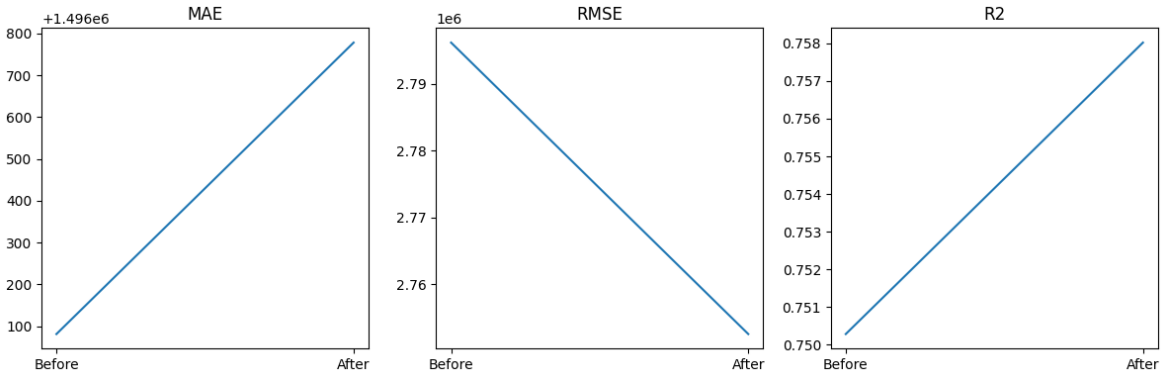
Sau khi khảo sát các kĩ thuật xử lý, ta sẽ tiếp tùng cải thiện bằng việc điều chỉnh các siêu tham số. Với 2 mô hình đã chọn, Linear Regression là một mô hình đơn giản nên không có bất kì siêu tham số nào, ngược lại với Random Forest Regression lại chứa rất nhiều siêu tham số có thể khảo sát, áp dụng các siêu tham số với Grid Search ta có:

- bootstrap: [True,False], quyết định liệu có lấy mẫu ngẫu nhiên tái chọn để tạo ra mỗi cây trong Random Forest hay sử dụng toàn bộ mẫu để tạo ra mỗi cây trong Random Forest
- 'max_depth': [None, 5, 10], None: Các cây trong Random Forest không có giới hạn độ sâu. 5, 10: Giới hạn độ sâu của các cây trong Random Forest. Điều này giới hạn số lượng nút và giúp kiểm soát độ phức tạp của mô hình.
- 'max_features': ['auto', 'sqrt'], 'auto': Số lượng đặc trưng được sử dụng để xây dựng mỗi cây trong Random Forest được tự động điều chỉnh. 'sqrt': Số lượng đặc trưng được sử dụng để xây dựng mỗi cây là căn bậc hai của số lượng đặc trưng ban đầu.
- 'min_samples_leaf': [1, 2, 4], Số lượng mẫu tối thiểu cần có trong mỗi lá của cây quyết định. Giá trị nhỏ hơn sẽ tạo ra các cây phức tạp hơn và dễ gây overfitting.
- 'min_samples_split': [2, 5, 10], Số lượng mẫu tối thiểu yêu cầu để chia một nút trong cây. Giá trị nhỏ hơn sẽ tạo ra cây phức tạp hơn và dễ gây overfitting.
- 'n_estimators': [100, 200, 300], Số lượng cây trong Random Forest

Ở BigDS:



và SmallDS:



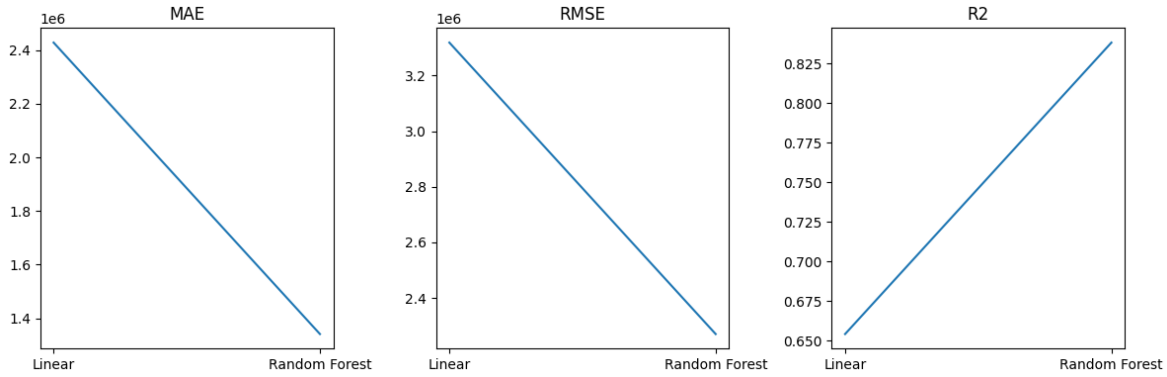
Hình 21: Kết quả so sánh trước và sau khi áp dụng điều chỉnh tham số

➔ Đều cho thấy việc điều chỉnh tham số gia tăng độ hiệu quả của mô hình

4.4. So sánh 2 mô hình

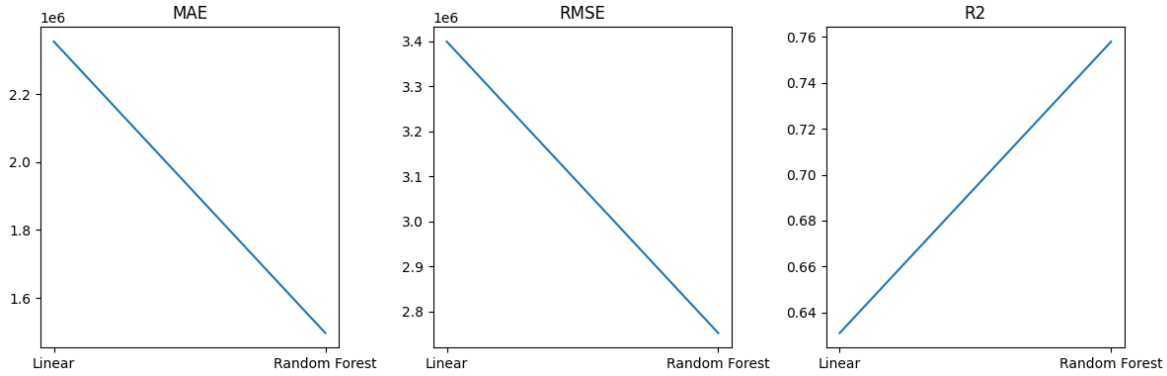
Hãy khảo sát 2 mô hình với các bộ đo hiệu quả nhất của nó:

Ở BigDS:




```
Linear Regression: MAE: 2428231.004813155 RMSE: 3318906.701552667 R2: 0.65408174167379
Random Forest Regression: MAE: 1341324.1543589572 RMSE: 2269913.6731243203 R2: 0.8381912215193819
```

Ở SmallDS:




```
Linear Regression: MAE: 2354408.8512621955 RMSE: 3399076.0944626066 R2: 0.630994536619345
Random Forest Regression: MAE: 1496778.4213080446 RMSE: 2752547.709871014 R2: 0.7580192978493049
```

Hình 22: So sánh 2 mô hình Linear Regression và Random Forest Regression

Bộ đo cho thấy, Random Forest Regression tỏ ra hiệu quả hơn nhiều so với Linear Regression. Ngoài ra bộ dữ liệu BigDS cho ra mô hình có độ hiệu quả cao hơn so với mô hình của bộ dữ liệu SmallDS

4.5. Kết quả

Điểm chung:

- Linear Regression ở cả 2 bộ dữ liệu đều có thể được cải thiện với kĩ thuật lọc ngoại lệ Outlier Handling

- Random Forest Regression hiệu quả hơn so với Linear Regression ở cả 2 bộ dữ liệu

Điểm khác:

- Random Forest ở BigDS có thể cải thiện bởi kĩ thuật chuẩn hóa Normalizer Scaling, ngược lại khi ở SmallDS có thể cải thiện bởi Outlier Handling và Feature Selection

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

- Đã thỏa mãn các yêu cầu tiểu luận đưa ra bao gồm: Thu thập dữ liệu, trích xuất đặc trưng, huấn luyện mô hình
- Mô hình đầu ra có độ hiệu quả cao, có thể áp dụng vào việc dự đoán giá điện thoại cũ

2. HƯỚNG PHÁT TRIỂN

- Thử nghiệm bộ dữ liệu với các mô hình dự đoán giá trị liên tục khác như Support Vector Regression hoặc Gradient Boosting Regression

TÀI LIỆU THAM KHẢO

[1] *https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html*

[2] *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html*

[3] Google Crawling and Indexing | Google Search Central | Documentation | Google for Developers

[4] The Selenium Browser Automation Project | Selenium