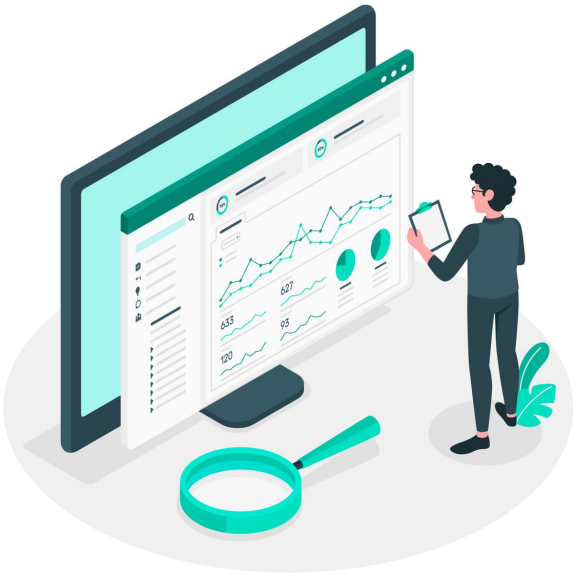
TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

TÊN ĐỀ TÀI: DỰ ĐOÁN GIÁ LAPTOP



HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN	ĐIỂM BẢO VỆ
Lê Hoàng Ngọc Hân	19N13	
Nguyễn Huy Tưởng	19N13	
Hồ Văn Vy	19N13	

ĐÀ NẴNG, 06/2022

TÓM TẮT

Hiện nay, laptop là một vật dụng hết sức phổ biến của mọi người dân với sự tiện dụng, nhỏ gọn của nó. Nhu cầu mua laptop ngày càng nhiều, vì vậy trong bài tập này, nhóm thực hiện đề tài **"Dự đoán giá laptop".** Nhóm thu thập dữ liệu gồm các thuộc tính và giá của mặt hàng laptop từ các trang thương mại điện tử. Sau đó lựa chọn các thuộc tính đắt giá, ảnh hưởng nhiều đến giá laptop để đưa ra dự đoán về giá cả. Tiền xử lý dữ liệu và khảo sát các mô hình hồi quy tuyến tính để xây dựng chương trình dự đoán giá của mặt hàng laptop.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá
Hồ Văn Vy	Cào dữ liệu từ 3 trang website Gộp dữ liệu Làm sạch dữ liệu thô Trực quan hoá dữ liệu	Đã hoàn thành
Nguyễn Huy Tưởng	Làm sạch và xử lý dữ liệu trống Xử lý ngoại lệ Phân tích mối tương quan, lựa chọn thuộc tính Chuẩn hóa và giảm chiều dữ liệu Thể hiện hiệu quả của các quá trình tiền xử lý	Đã hoàn thành
Lê Hoàng Ngọc Hân	Khảo sát mô hình. Cài đặt mô hình. Lựa chọn mô hình, điều chỉnh siêu tham số sử dụng GridSearchCV. Trực quan hóa kết quả dự đoán trên các mô hình. Tìm hiểu và tính toán các metrics để đưa ra so sánh, nhận xét.	Đã hoàn thành

MỤC LỤC

1. Giới thiệu

Ngày nay, cùng với sự phát triển của khoa học và công nghệ cũng như trong bối cảnh dịch COVID 19 kéo dài, laptop trở thành một công cụ hữu ích phục vụ nhu cầu từ việc học online, làm việc tại nhà cho đến việc giải trí càng khiến cho thị trường laptop ngày càng nóng lên. Việc lựa chọn một chiếc laptop sao cho phù hợp với nhu cầu và túi tiền không phải là điều dễ dàng đối với nhiều người.

Nắm bắt được nhu cầu cũng như tâm lý người dùng, nhóm đã ứng dụng những kiến thức đã học trong bộ môn Khoa học dữ liệu để xây dựng mô hình "Dự đoán giá laptop" trên tập dữ liệu được thu thập từ các trang website bán laptop nổi tiếng như fptshop.com.vn, philong.com.vn, cellphones.com.vn.

Giải pháp của nhóm là sẽ sử dụng các công cụ như Selenium để hỗ trợ cào dữ liệu, sau đó xây dựng các mô hình hồi quy tuyến tính nhằm dự đoán giá laptop kết hợp với các kĩ thuật xử lý dữ liệu trống, dữ liệu ngoại lệ và chuẩn hóa.

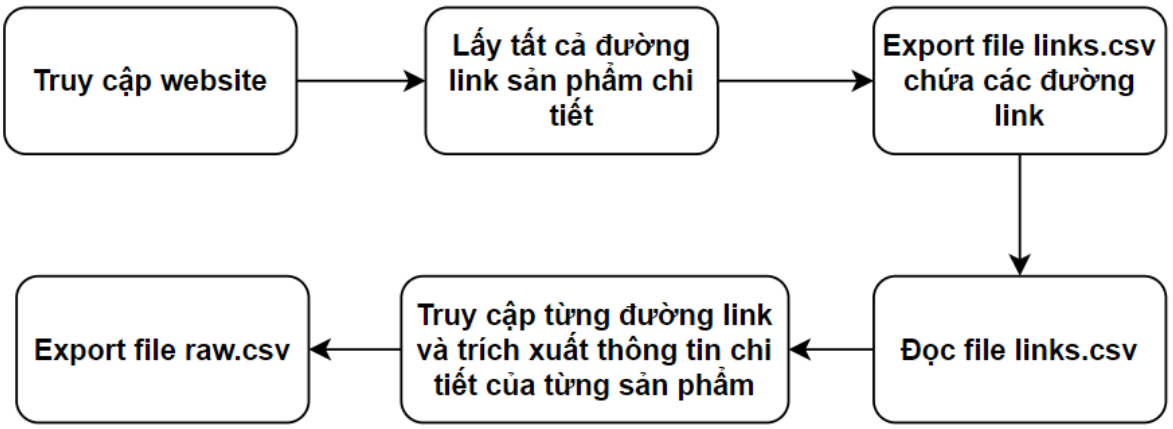
2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

Nhóm lựa chọn sẽ lấy dữ liệu trên 3 trang website bán laptop phổ biến là fptshop.com.vn, philong.com.vn, cellphones.com.vn. Cụ thể là 3 đường link chi tiết sau:

- https://fptshop.com.vn/may-tinh-xach-tay
- https://philong.com.vn/may-tinh-xach-tay.html
- https://cellphones.com.vn/laptop.html

Quá trình cào dữ liệu có thể tổng quát thành các bước sau:



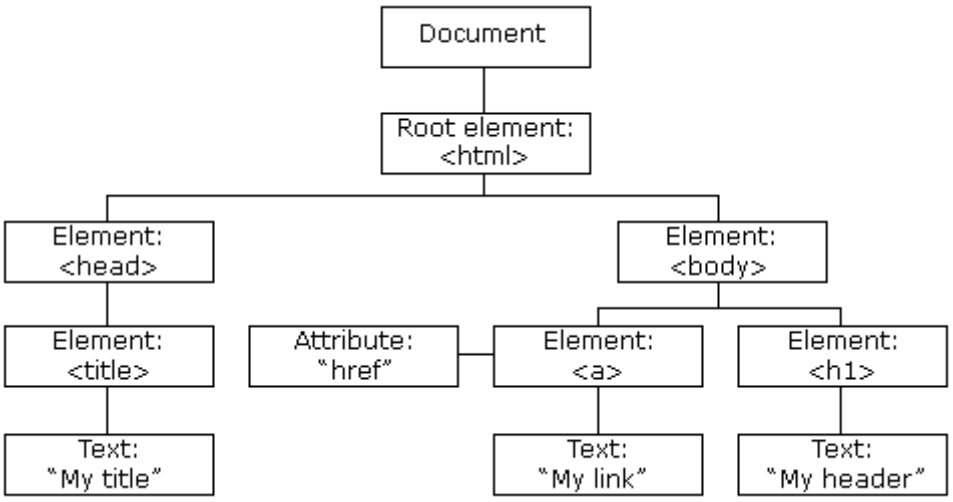
Hình 1. Các bước cào dữ liệu

Đầu vào của bước cào dữ liệu là URL website muốn cào.

Kết quả quá trình này ta sẽ thu được file dữ liệu thô chứa thông tin chi tiết của từng loại laptop.

Dựa trên sự phân tích về đặc trưng của 3 trang website trên thì nhóm lựa chọn các công cụ sau để cào dữ liệu: Selenium, Request, BeautifulSoup của Python.

- **Selenium**: Một công cụ nổi tiếng trong lĩnh vực kiểm thử, giúp giả lập các tác vụ của người dùng trên trình duyệt như click, lăn chuột,... Nhóm sử dụng Selenium để giả lập quá trình lăn chuột giúp lấy link sản phẩm và cào dữ liệu với các trang website có sử dụng kĩ thuật lazy-load (chỉ load sản phẩm khi người dùng kéo tới vị trí thích hợp trên website) như fptshop.com.vn, cellphones.com.vn
- **Requests**: Thư viện của Python, giúp lập trình gửi nhận HTTP request. Thư viện này giúp lấy source text HTML của các trang website không sử dụng kĩ thuật lazy-load như philong.com.vn
- **Beautiful Soup**: Thư viện của Python, giúp phân tích source text HTML lấy được từ 2 công cụ ở trên và phân tích thành cấu trúc cây các Object như hình bên dưới, thuận tiện cho việc truy xuất và lấy thông tin các phần tử HTML.



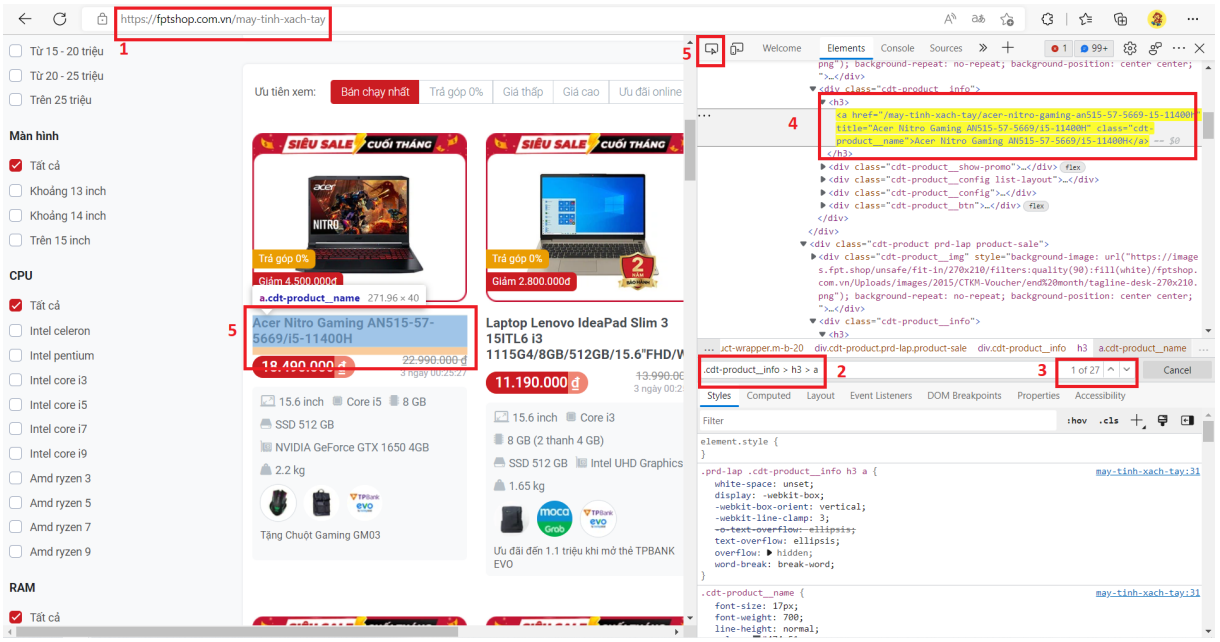
Hình 2. Cấu trúc cây HTML

Ví dụ về cào dữ liệu trang fptshop.com.vn

Bước 1: Thu thập tất cả các đường link sản phẩm

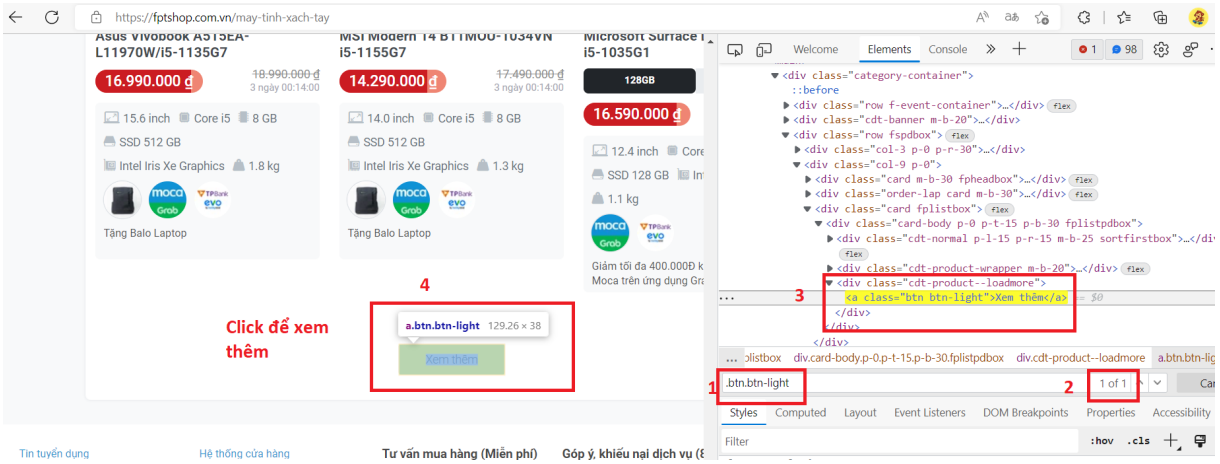
Đầu tiên ta cần phải xác định thủ công đường link sản phẩm nằm ở đâu trong source code HTML:

- Truy cập trang https://fptshop.com.vn/may-tinh-xach-tay
- Bấm F12 hoặc click chuột phải chọn View page source
- Chuyển sang tab Elements ở cửa sổ View page source
- Sau quá trình tìm thủ công với sự hỗ trợ công cụ Inspect (vị trí số 5 ở hình bên dưới), ta nhập css selector sau ở vị trí số 2 ở hình bên dưới "**.cdt-product__info > h3 > a**" thì thấy kết quả là đường link ta cần tìm và có tới 27 kết quả trong trang (ứng với vị trí số 4 và vị trí số 3 ở hình bên dưới)



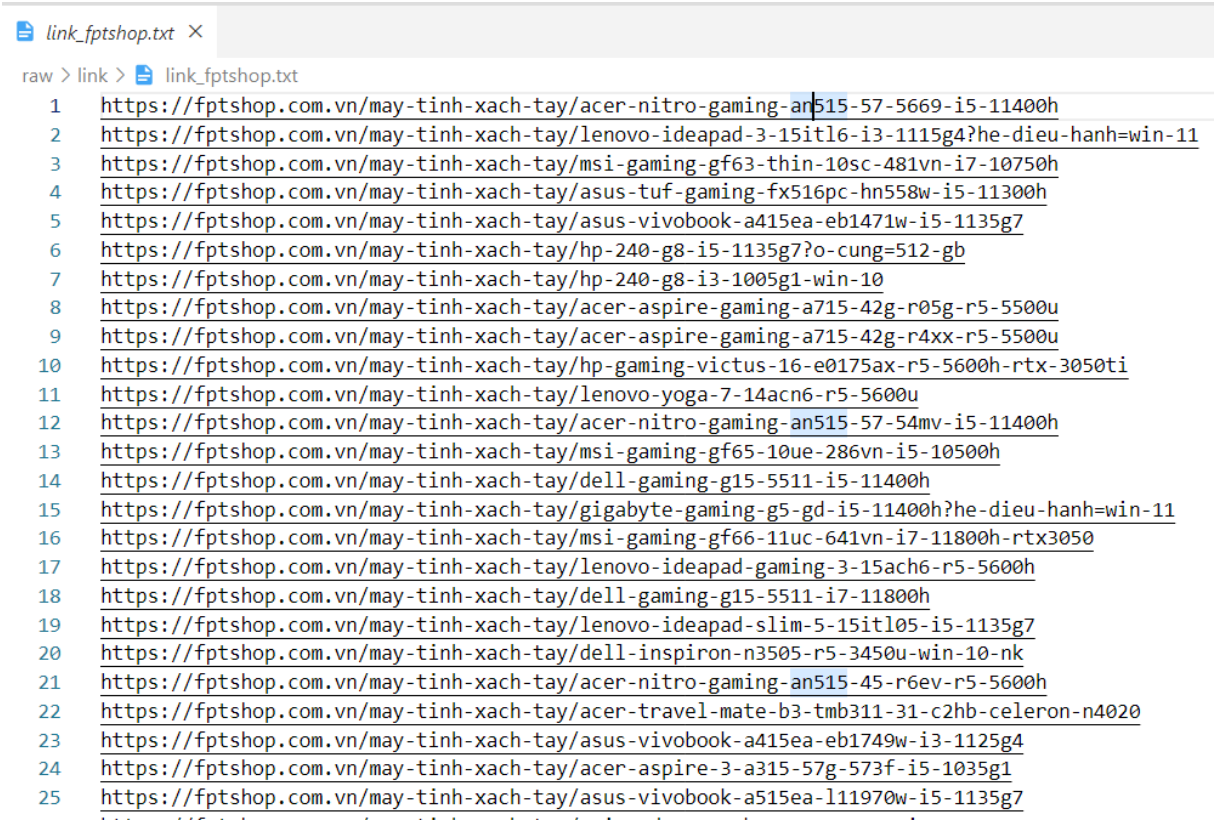
Hình 3. Trang sản phẩm laptop của fptshop

Website có rất nhiều trang, vì vậy để lấy đầy đủ thông tin ta cần xác định vị trí nút xem thêm. Bằng kỹ thuật tương tự như trên, ta dễ dàng xác định được CSS Selector cho nút xem thêm là "**.btn.btn-light**"



Hình 4. Nút xem thêm ở trang sản phẩm fptshop

Sau đó ta tiến hành sử dụng Selenium để giả lập người dùng duyệt website, và sử dụng các phương thức như **find_element(By.CSS_SELECTOR, "<css-selector-link-or-button")** để bóc tách đường link hoặc giả lập hành động click vào nút xem thêm. Lưu tất cả đường link vào file csv và đây chính là kết quả của bước đầu tiên

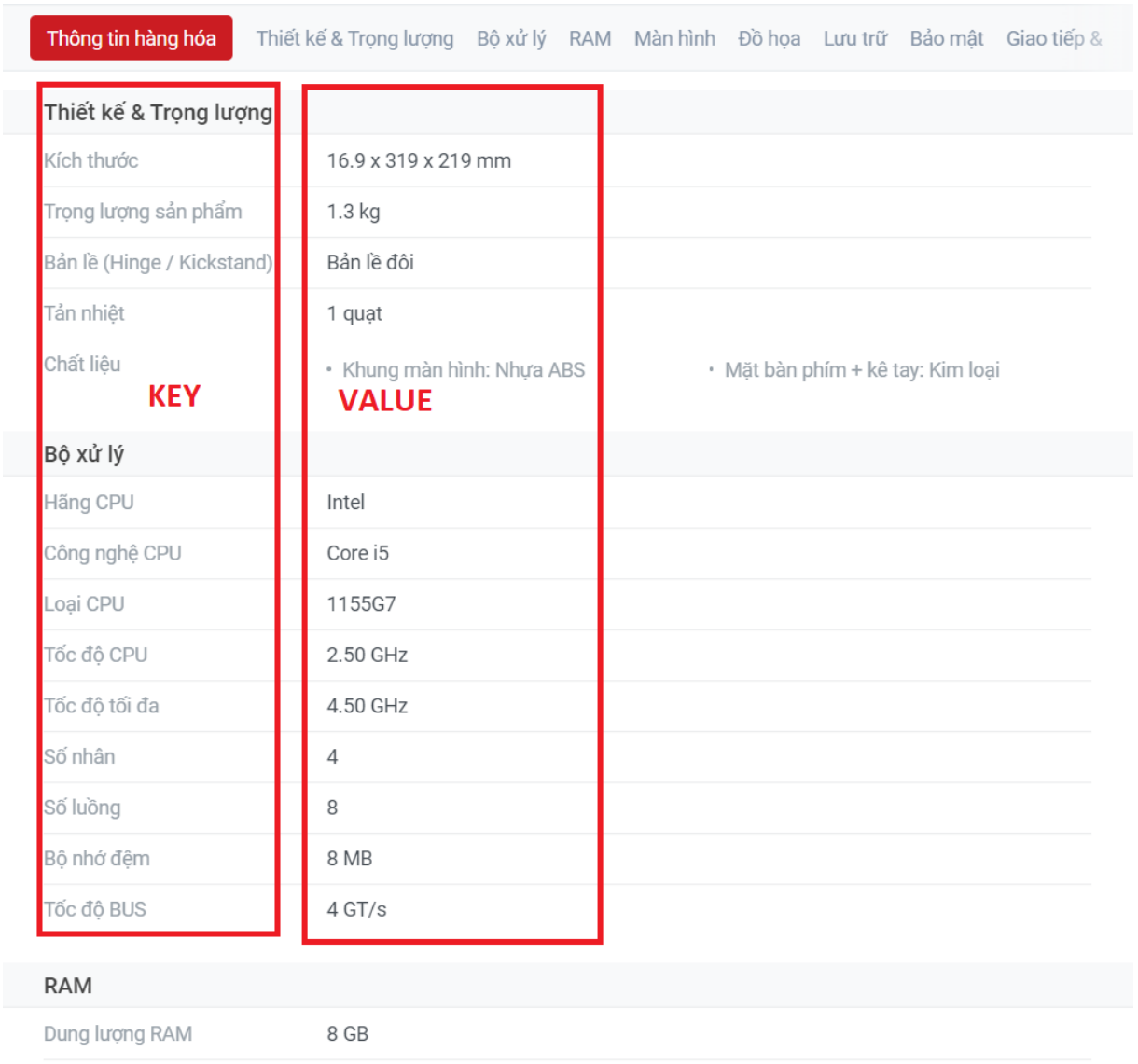


Hình 5. File csv đường link sản phẩm

Bước 2: Lấy thông tin chi tiết sản phẩm

Sau khi có được tất cả đường link sản phẩm, tiến hành duyệt qua từng link sản phẩm và sử dụng Selenium để get source text HTML và dùng Beautiful Soup để chuyển source text HTML thành cây đối tượng.

Điểm chung của tất cả trang sản phẩm chi tiết là đều có một bảng thông tin chi tiết, gồm 2 cột: cột bên trái là key, cột bên phải là giá trị. Ta có thể lợi dụng bảng thông tin này kết hợp với Beautiful Soup để truy xuất dữ liệu bằng CSS Selector như ở các bước trên.



Hình 6. Bảng thông tin chi tiết sản phẩm tại trang fptshop.com.vn

Sau khi hoàn tất bước này, nhóm thu được 1 file csv chứa đầy đủ thông tin của tất cả sản phẩm laptop tại trang fptshop.com.vn

Hình 7. File csv chứa dữ liệu thô

2.2. Mô tả dữ liệu

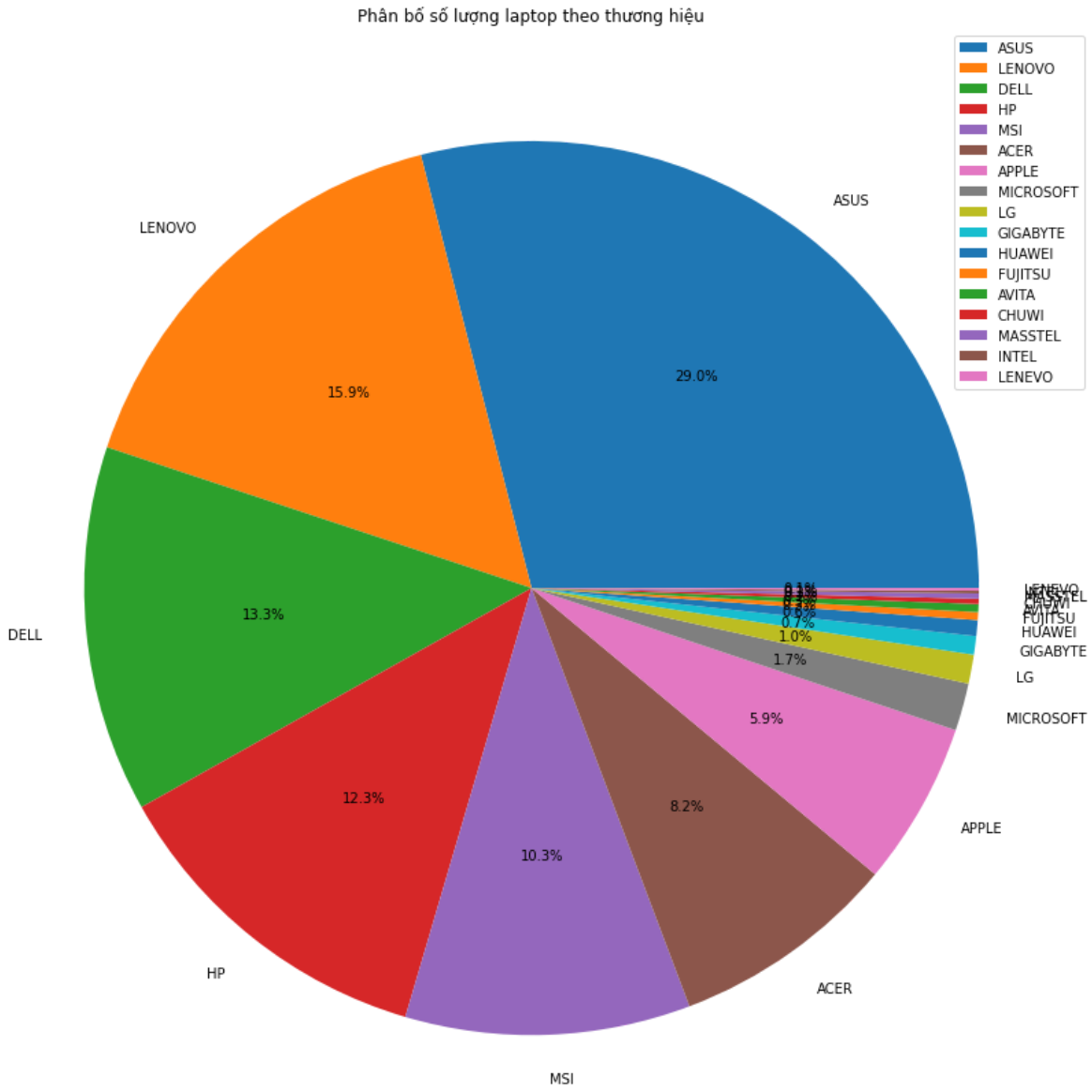
Sau khi có dữ liệu thô, tiến hành làm sạch dữ liệu bằng cách tạo các cột dữ liệu mới, trích xuất giá trị hữu ích từ các dữ liệu có sẵn và ép kiểu dữ liệu sang kiểu dữ liệu thích hợp. Kết quả thu được tập dữ liệu sau khi làm sạch có:

- Số lượng đặc trưng: **13**
- Số lượng mẫu: **1049**

Bảng 1. Tổng quan về tập dữ liệu

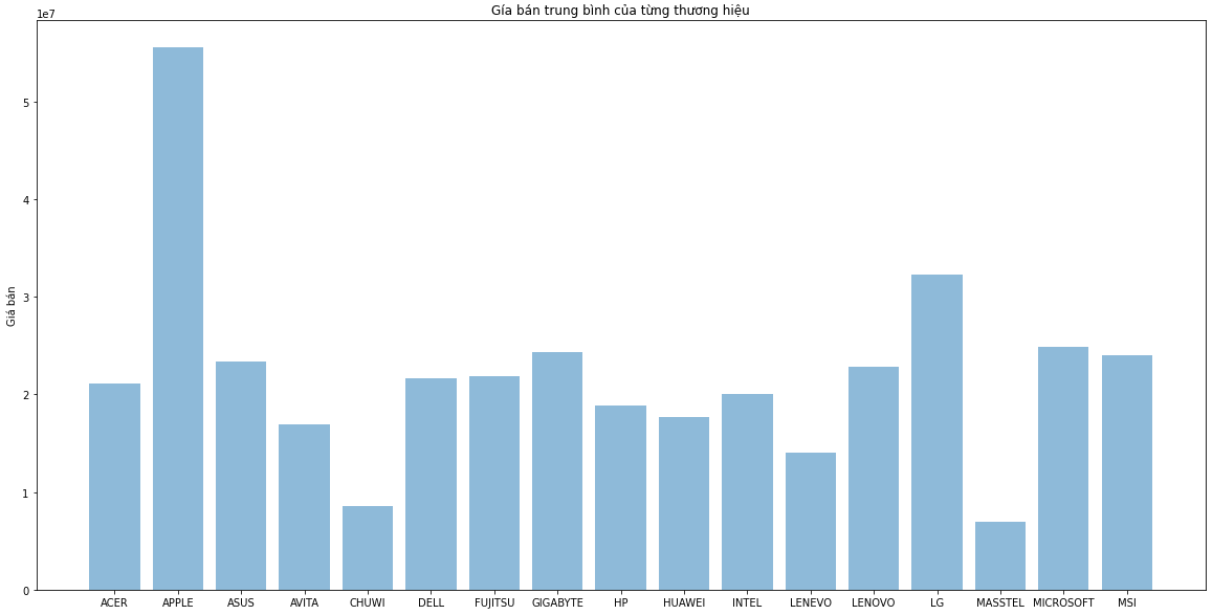
STT	Đặc trưng	Mô tả	Kiểu dữ liệu	Số mẫu dữ liệu trống
1	ProductName	Tên laptop	String	0
2	PriceSale	Giá bán (vnđ)	Integer	12
3	Brand	Thương hiệu laptop	String	0
4	RamCapacity	Dung lượng RAM (GB)	Integer	20
5	DisplayResolution	Độ phân giải màn hình	String	85
6	DisplaySize	Kích thước màn hình (inch)	Float	101

7	CPUBrand	Thương hiệu chip	String	132
8	PinCapacity	Dung lượng pin (Wh)	Float	381
9	PinCell	Số lượng cell pin (cell)	Integer	261
10	Bluetooth	Phiên bản Bluetooth	Integer	66
11	Weight	Khối lượng laptop (Kg)	Float	85
12	OS	Hệ điều hành	String	117
13	DiskSpace	Dung lượng ổ cứng	Integer	60



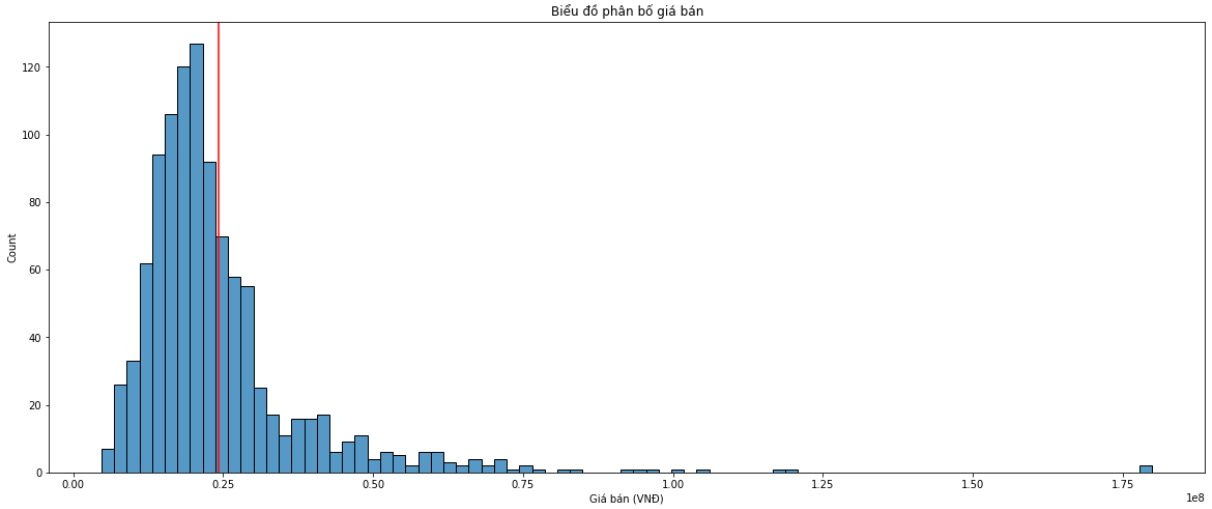
Hình 8. Biểu đồ phân bố số lượng laptop theo thương hiệu

Số lượng laptop được trưng bày ở 3 cửa hàng theo từng thương hiệu như Asus, Lenovo, Dell, Hp, MSI, Acer, Apple chiếm số lượng lớn. Ngược lại những hãng như Masstel, Chuwi, Avita thì chiếm số lượng rất ít.



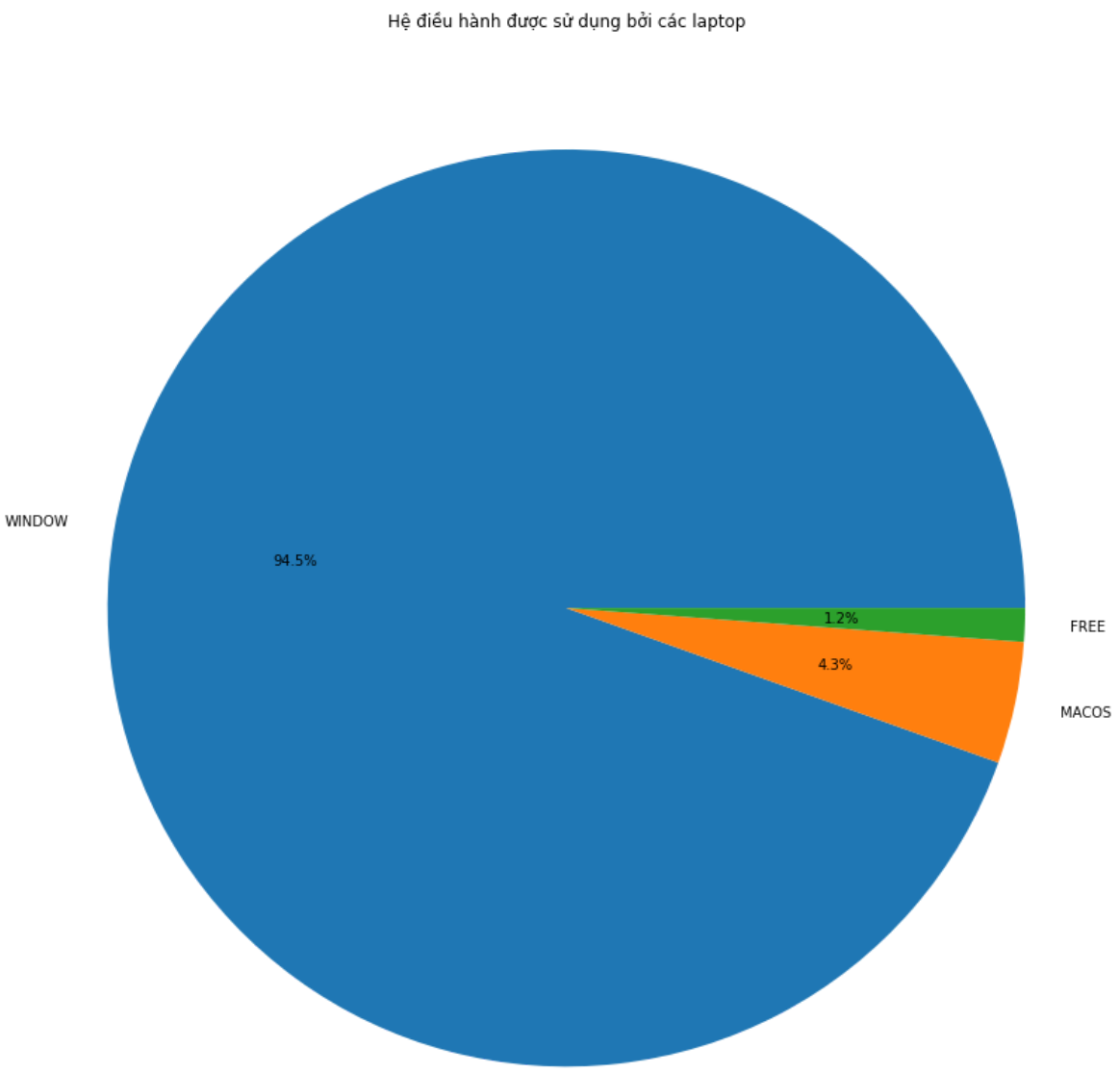
Hình 9. Biểu đồ giá bán laptop trung bình của từng thương hiệu

Laptop mang thương hiệu Apple có giá bán trung bình cao nhất tầm khoảng 55 triệu, tiếp đến là LG tầm 32 triệu. Một số hãng laptop khác như Masstel, Chuwi thì có giá bán < 10 triệu.



Hình 10. Biểu đồ phân bố giá bán

Giá bán laptop trên thị trường hiện nay tập trung chủ yếu từ không 15 đến 30 triệu, có giá bán trung bình rơi vào khoảng xấp xỉ 25 triệu đồng. Phân khúc thị trường laptop cao cấp, có giá bán cáo thì có rất ít sản phẩm.



Hình 11. Biểu đồ tỉ trọng hệ điều hành được sử dụng bởi các laptop

Đa số các laptop đều được cài đặt sẵn hệ điều hành Windows, chỉ một số ít là sử dụng MacOS, Free (Linux). Điều này cũng hoàn toàn đúng, vì hệ điều hành Windows rất phổ biến với người dùng phổ thông với giao diện dễ sử dụng, MacOS chỉ được cài đặt trên các máy tính của Apple và Linux thì chỉ thích hợp với những người dùng chuyên biệt.

3. Trích xuất đặc trưng

3.1 Làm sạch dữ liệu

Dữ liệu sau khi thu thập từ nhiều nguồn khác nhau có thể lẫn những mẫu dữ liệu không đảm bảo chất lượng, vì vậy trước khi thực hiện các bước xử lý dữ liệu cần thiết phải loại bỏ những mẩu không cần thiết.

Cụ thể trong tập dữ liệu đã thu thập, cần loại bỏ những mẩu có trường giá cả bị trống (do lỗi hoặc do nguồn không cung cấp giá). Đồng thời trong tập dữ liệu củng xuất hiện rất nhiều dữ liệu trống, bước lấp đầy dữ liệu trống là cần thiết, tuy nhiên một mẫu có quá nhiều thuộc tính trống thì cho dù dữ liệu trống được lấp đầy cũng không hiệu quả trong việc huấn luyện và dự đoán. Vì vậy tiến hành loại bỏ và chỉ giữ lại những mẫu có từ 1 trường trống trở xuống, việc lựa chọn 1 trường trống là hợp lý bởi vì qua khảo sát, số trường trống càng cao thì độ chính xác mô hình càng giảm, đồng thời nếu giữ lại những mẫu có 2 trường trống thì tập giữ liệu sẽ bị giảm mất một nửa, còn lại rất ít dữ liệu để thực hiện việc huấn luyện và kiểm thử.

Sau bước làm sạch, dữ liệu giảm từ 1049 mẫu xuống còn 796 mẫu.

3.2 Xử lý dữ liệu trống

Dữ liệu trống là những giá trị bị thiếu trong tập dữ liệu liệu do nơi thu thập không cung cấp hoặc thu thập thiếu. Có thể xử lý một cách nhanh gọn hơn bằng cách xóa tất cả các mẫu có dữ liệu trống hoặc xóa bỏ thuộc tính có nhiều mẫu trống. Tuy nhiên việc này cũng đồng thời xóa đi rất nhiều giá trị hữu ích, vì vậy cần có một phương pháp lấp đầy các dữ liệu trống này để không mất quá nhiều dữ liệu mà vẫn giữ được các giá trị cần thiết trong tập dữ liệu.

Đối với dữ liệu thuộc kiểu category (hãng CPU và hệ điều hành) không thể dùng các phương pháp tính toán để tính dữ liệu trống cho nên dữ liệu trống loại này sẽ được lấp đầy bằng các random từ các mẫu có dữ liệu. Phương pháp này giúp thuộc tính có thể giữ đúng phân bố của mình sau khi lấp đầy vị trí trống.

Đối với dữ liệu số, sử dụng phương pháp Iterative imputer để lấp đầy dữ liệu trống. Phương pháp này dùng dữ liệu các trường còn lại làm X, dữ liệu trống làm y. Tiến hành huấn luyện mô hình hồi quy trên dữ liệu có sẵn để dự đoán giá trị tại các ô

trống với mục tiêu làm cho những dữ liệu trống đó tự nhiên nhất, ít làm ảnh hưởng đến các thuộc tính khác khi huấn luyện. Phân bố dữ liệu sau khi lấp đầy các giá trị trống được thể hiện ở hình 14.

3.3 Mã hóa dữ liệu

Các dữ liệu dạng số được giữ nguyên, trong khi đó dữ liệu thuộc loại category cần được mã hóa sang dạng số để có thể huấn luyện và dự đoán vì các mô hình sẽ không làm việc với các dữ liệu kiểu chuỗi ký tự. Dữ liệu category là dữ liệu không sắp xếp, các giá trị trong thuộc tính có vai trò như nhau, vì vậy dữ liệu này được mã hóa bằng One hot vector, là một kỹ thuật mã hóa giúp giữ cho các giá trị có tầm quan trọng như nhau.

Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

One-Hot Encoded Data

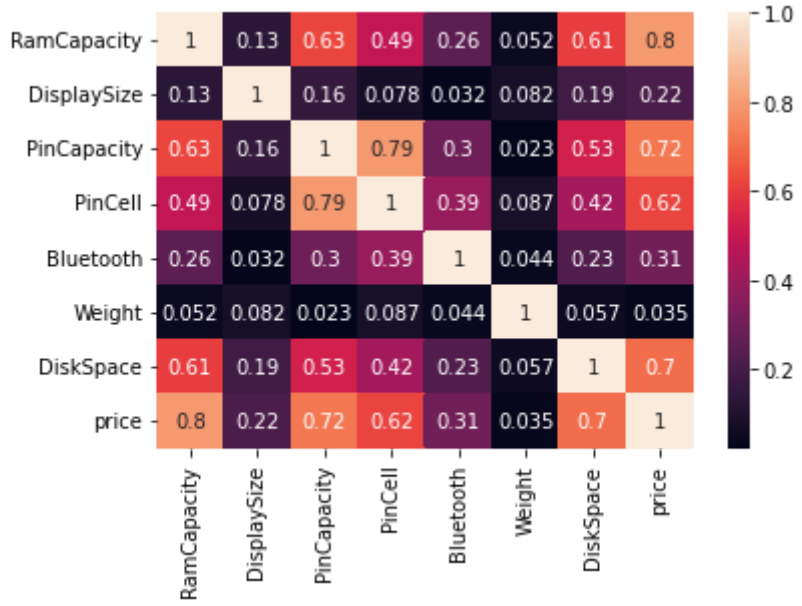
Team_A	Team_B	Team_C	Points
1	0	0	25
1	0	0	12
0	1	0	15
0	1	0	14
0	1	0	19
0	1	0	23
0	0	1	25
0	0	1	29

Hình 12. Mã hóa one hot vector

Hình trên mô tả dữ liệu được mã hóa từ dữ liệu category sang one hot vector. Thuộc tính cần mã hóa có k giá trị khác nhau, thì mỗi giá trị ban đầu sẽ được mã hóa thành vector k giá trị 0 và 1, có một vị trí duy nhất bằng 1 là vị trí mà đại diện cho giá trị ban đầu. Xem ví dụ trên có thể thấy trường Team có 3 giá trị khác nhau A, B, C, vậy nên với mỗi giá trị được mã hóa thành vector 3 chiều, vector [1, 0, 0] đại diện cho giá trị A, vector [0, 1, 0] đại diện cho giá trị B và vector [0, 0, 1] đại diện cho giá trị C.

Trong tập dữ liệu của đề tài này hãng laptop, hãng CPU và hệ điều hành được mã hóa dưới dạng one hot vector.

3.4 Phân tích sự tương quan



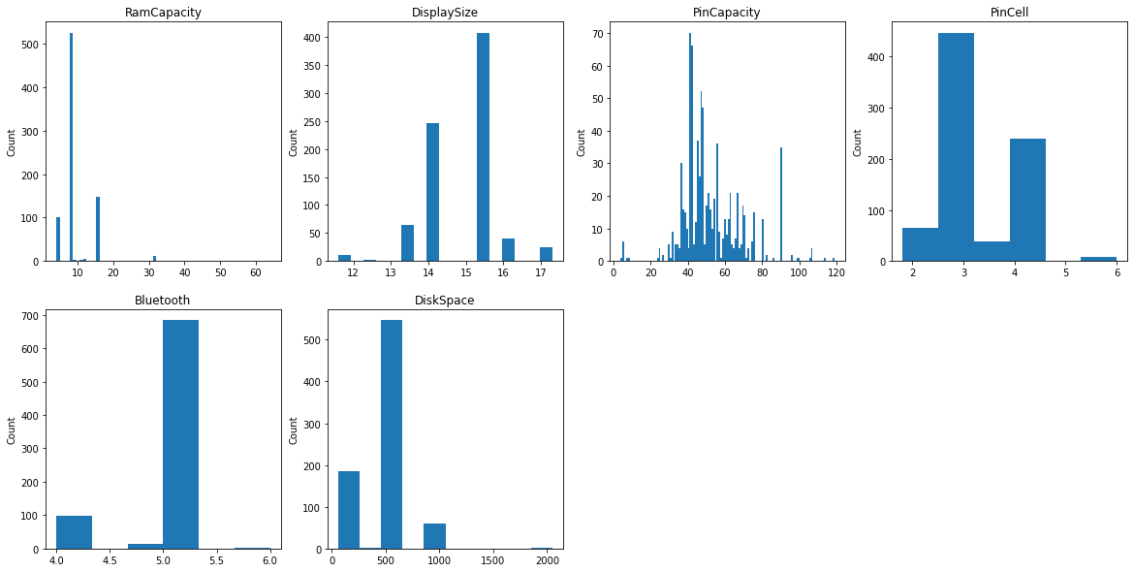
Hình 13. Ma trận tương quan giữa các thuộc tính dữ liệu số

Hình trên cho thấy mức độ tương quan giữa các thuộc tính số trong tập dữ liệu, trong đề tài này chỉ cần quan sát sự tương quan giữa giá tiền đối với các thuộc tính khác. Có thể thấy dung lượng RAM, dung lượng pin, kích thước pin và bộ nhớ có tương quan lớn (>0.6) đối với giá tiền. Điều này cũng khá thực tế vì những thuộc tính này ảnh hưởng trực tiếp đến hiệu suất và giá trị sử dụng của một chiếc laptop.

Các thuộc tính còn lại như kích thước màn hình, công nghệ bluetooth và cân nặng tương quan thấp với giá tiền, đặc biệt cân nặng có mức độ tương quan cực kỳ thấp và nên loại bỏ để tránh làm ảnh hưởng đến hiệu suất của mô hình. Sau khi loại bỏ trường cân nặng thì giá trị phần trăm trị tuyệt đối sai số (huấn luyện trên mô hình Linear Regression mặc định của thư viện scikit-learn) cải thiện từ 18.04% xuống 17.97%

3.5 Xử lý ngoại lệ

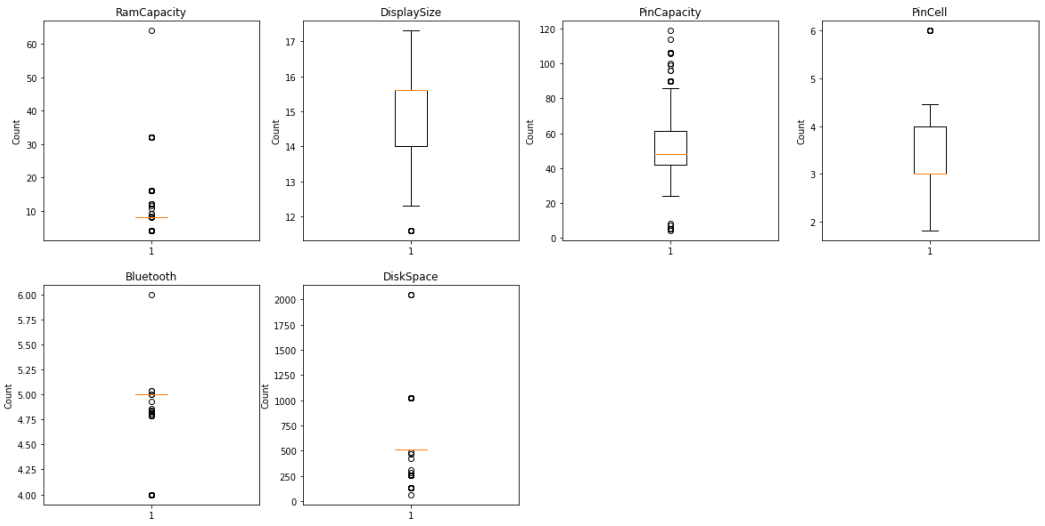
Một số mô hình rất nhạy với giá trị ngoại lệ trong đó có mô hình hồi quy tuyến tính, vì vậy cần kéo các giá trị ngoại lệ về khoảng giá trị chấp nhận được. Có hai dạng xử lý ngoại lệ là xử lý dữ liệu phân bố chuẩn và xử lý ngoại lệ bị mất cân bằng. Các trường dữ liệu số trong tập dữ liệu hầu hết xấp xỉ phân bố chuẩn.



Hình 14. Phân bố các trường dữ liệu số

Phân bố chuẩn thường được xử lý ngoại lệ theo phương pháp:

- Xác định cận trên bằng tổng giá trị trung bình và 3 lần độ lệch chuẩn
- Xác định cận dưới bằng hiệu giá trị trung bình và 3 lần độ lệch chuẩn
- Những giá trị lớn hơn cận trên gán bằng giá trị cận trên
- Những giá trị nhỏ hơn cận dưới gán bằng giá trị cận dưới
- Lưu ý: Các cận được tính trên tập train và việc gán giá trị được thực hiện trên cả tập train và tập test.



Hình 15. Boxplot của các thuộc tính giá trị số

Qua biểu đồ boxplot ở trên cho thấy tất cả các thuộc tính đều có giá trị ngoại lệ vì vậy việc xử lý ngoại lệ theo cách trên được áp dụng cho tất cả các trường. Sau khi

xử lý ngoại lệ thì giá trị phần trăm trị tuyệt đối sai số (huấn luyện trên mô hình Linear Regression mặc định của thư viện scikit-learn) cải thiện từ 17.96% xuống 17.92%

3.5 Chuẩn hóa

Dữ liệu số được chuẩn hóa theo công thức:

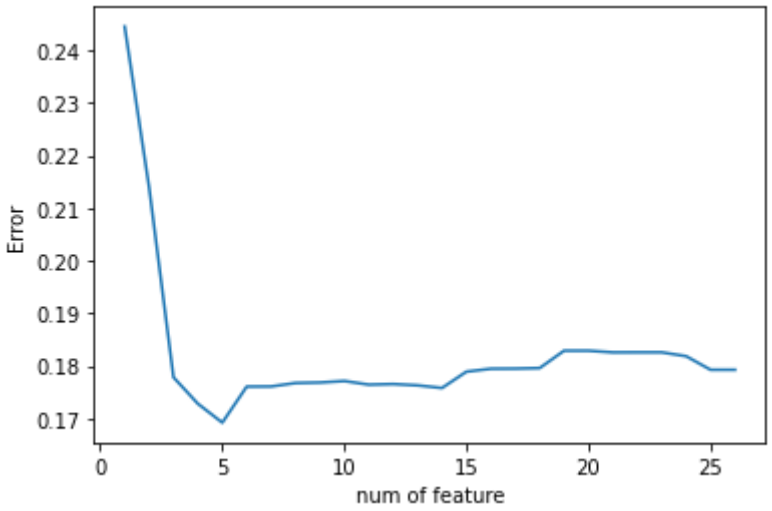
$$z = (x - mean) / std$$

Trong đó:

- z là giá trị sau khi chuẩn hóa
- x là giá trị cần chuẩn hóa
- mean là giá trị kỳ vọng của thuộc tính
- std là phương sai của thuộc tính
- Lưu ý: mean và std được tính trên tập train, sau đó chuẩn hóa cho tập train và tập test

Phương pháp chuẩn hóa này giúp các thuộc tính có đơn vị khác nhau có thể có vai trò như nhau trong tập dữ liệu vì giá trị trung bình bằng 0 và độ lệch chuẩn là 1. Sau bước chuẩn hóa này kết quả hiệu suất mô hình không có sự thay đổi nhiều, tuy nhiên đây là một bước hầu như bắt buộc trong việc tiền xử lý dữ liệu và được cho là tăng hiệu suất cho các mô hình hồi quy tuyến tính.

3.6 Lựa chọn đặc trưng

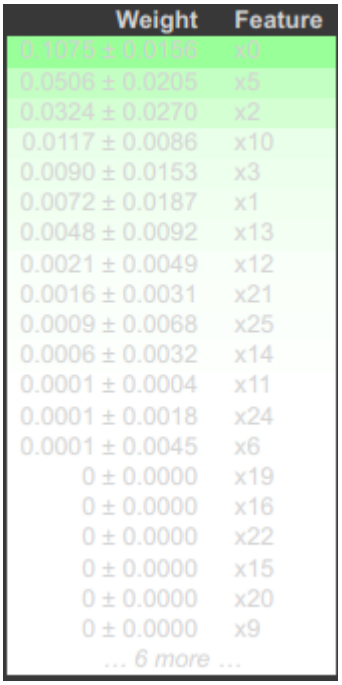


Hình 16. Mức độ lỗi theo số lượng thuộc tính

Tập dữ liệu có nhiều thuộc tính không đồng nghĩa với việc hiệu suất sẽ tốt. Đôi khi các thuộc tính lỗi lại gây giảm hiệu suất cho mô hình dự đoán. Vì vậy, cần lựa chọn và giữ lại những đặc trưng đắt giá của mô hình.

Trong đề tài này, phương pháp Permutation importance được sử dụng để lựa chọn các thuộc tính đắt giá trong tập dữ liệu. Phương pháp này đánh giá mức độ quan trọng của thuộc tính là độ giảm của điểm số khi giá trị của thuộc tính đó bị hoán vị, làm mất đi giá trị dự đoán của thuộc tính đó. Dựa vào phương pháp này, chúng ta sẽ đánh giá mức độ quan trọng của tất cả thuộc tính, sau đó lựa chọn k thuộc tính tốt nhất cho hiệu suất cao nhất.

Hình 16 cho ta thấy mức độ lỗi của mô hình đạt giá trị nhỏ nhất khi lựa chọn 5 thuộc tính quan trọng nhất. Vì vậy ta chỉ chọn 5 thuộc tính này cho việc huấn luyện và dự đoán.

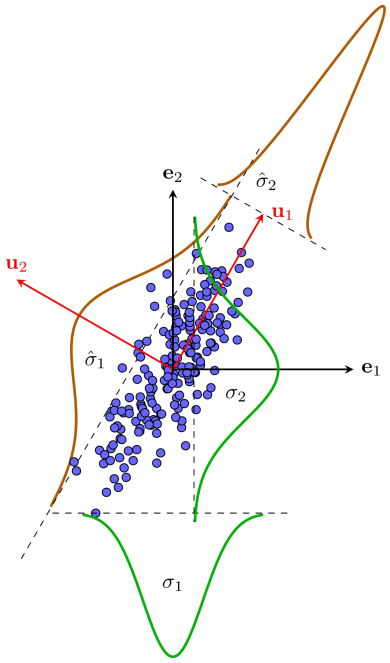


Hình 17. Mức độ quan trọng của các thuộc tính

3.7 Giảm chiều dữ liệu

Phương pháp giảm chiều dữ liệu thường được dùng để giảm kích thước của dữ liệu, đồng thời tìm ra quy luật ẩn, một không gian mới mà thể hiện tốt hơn thuộc tính đó. Phân tích thành phần chính (PCA) là phương pháp giảm chiều dữ liệu khá phổ biến

với mục tiêu tìm ra không gian mới mà mức độ phân tán dữ liệu càng cao càng tốt (tuy nhiên không phải bao giờ giờ dữ liệu càng phân tán củng là có lợi).



Hình 18. Mô tả phương pháp PCA

Hình trên cho ta thấy cái nhìn trực quan về phương pháp PCA. Với 2 vector kỳ vọng ban đầu e_1, e_2 có phương sai o_1, o_2 tương đối đều nhau. Phương pháp PCA tìm các trị riêng ứng với các vector riêng của ma trận hiệp phương sai sau khi đã dịch chuyển dữ liệu về trung tâm. Sắp xếp trị riêng giảm dần đề có được các không gian mới có phương sai giảm dần. Từ đó ta có được các chiều dữ liệu mới và có dữ liệu ở chiều mới bằng cách chiếu dữ liệu cũ vào các vector tìm được. Lưu ý rằng PCA không làm thay đổi phương sai dữ liệu vì theo quy tắc dù cho chiếu trên những không gian khác nhau cùng số chiều thì tổng phương sai không thay đổi, PCA chỉ tìm ra không gian mới mà tối ưu hóa phương sai lên mức tối đa.

Trong đề tài này việc áp dụng phương pháp PCA hầu như không tối ưu được hiệu suất của mô hình trong bất kỳ số lượng chiều nào.

Bảng 2. Phần trăm trị tuyệt đối sai số theo số chiều PCA

Số chiều PCA	**Phần trăm trị tuyệt đối sai số**
1	18.47%

2	17.41%
3	17.35%
4	17.35%
5	16.92%

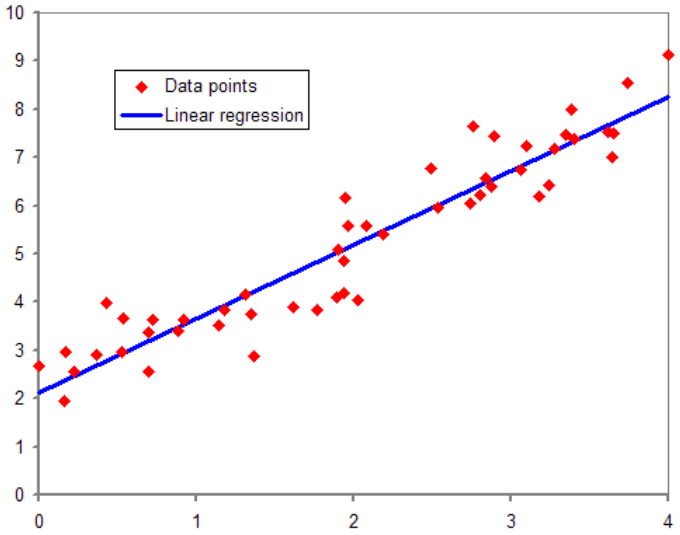
4. Mô hình hóa dữ liệu

4.1. Mô hình sử dụng

4.1.1. Hồi quy tuyến tính (Linear Regression)

Hồi quy tuyến tính là một thuật toán học máy cơ bản thuộc loại học có giám sát. Đây là phương pháp thống kê để hồi quy dữ liệu với những biến phụ thuộc có giá trị liên tục dựa vào những biến độc lập (có thể liên tục hoặc không liên tục).

Một bài toán được hồi quy tuyến tính giải quyết một cách hiệu quả khi mà các biến độc lập có mối quan hệ tuyến tính với các biến phụ thuộc. Hay nói cách khác, ảnh hưởng của sự thay đổi trong giá trị của các biến độc lập nên ảnh hưởng thêm vào tới các biến phụ thuộc.



Hình 19. Minh họa mô hình hóa Hồi quy tuyến tính.

- Phương trình tuyến tính ở dạng ma trận: y = X.w + bias.

Trong đó:

y: vector giá trị dự đoán.

X: ma trận các đặc trưng (giá trị không phụ thuộc), với số hàng là số lượng mẫu và số cột là số lượng đặc trưng.

w: ma trận các trọng số ứng với từng phần tử trong X.

bias: vector hệ số tự do, có số phần tử bằng số lượng mẫu.

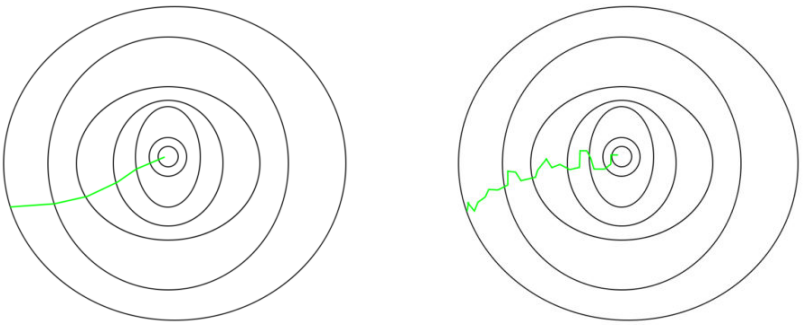
- **Thuật toán:**
  + **Bước 1:** Thu thập dữ liệu (dạng X và y), chia dữ liệu thành train và test.
  + **Bước 2:** Khởi tạo 1 ma trận w (trọng số) ứng với các phần tử của ma trận đặc trưng X và các hệ số bias. Dự đoán y' theo công thức y = X.w + bias. Tính loss giữa y (giá trị thật) và y' (giá trị dự đoán) sau đó tính đạo hàm và cập nhật w một lượng bằng $-\alpha.dt$ trong đó α là một tham số điều chỉnh (thường là 0.001 ~ 0.01) và dt là đạo hàm của loss.
  + **Bước 3:** Lặp lại bước 2 với n lần cho đến khi loss không thể tối ưu thêm được nữa thì dừng, ta thu được bộ trọng số w và bias đã được huấn luyện xong.
  + **Bước 4:** Tiến hành dự đoán trên tập test sử dụng bộ trọng số w, bias vừa train xong và đánh giá kết quả.

→ Ở bài tập này, nhóm sử dụng Linear Regression do thư viện sklearn cung cấp nên tham số α đã được lựa chọn tối ưu nhất. Đối với Linear Regression thì không có bất cứ một siêu tham số nào nên sẽ không áp dụng được các kĩ thuật lựa chọn, điều chỉnh mô hình như fine tuning, model selection.

4.1.2. Stochastic Gradient Descent Regressor:

Hồi quy SGD cũng là một mô hình hồi quy tuyến tính nhưng tối ưu loss bằng thuật toán Stochastic Gradient Descent (SGD). Nói về SGD thì đây là thuật toán chọn ngẫu nhiên 1 vài điểm dữ liệu trên toàn bộ dữ liệu (thường gọi là 1 batch) và tiến hành tính toán loss, cập nhật w cho một lần lặp. SGD mặc dù

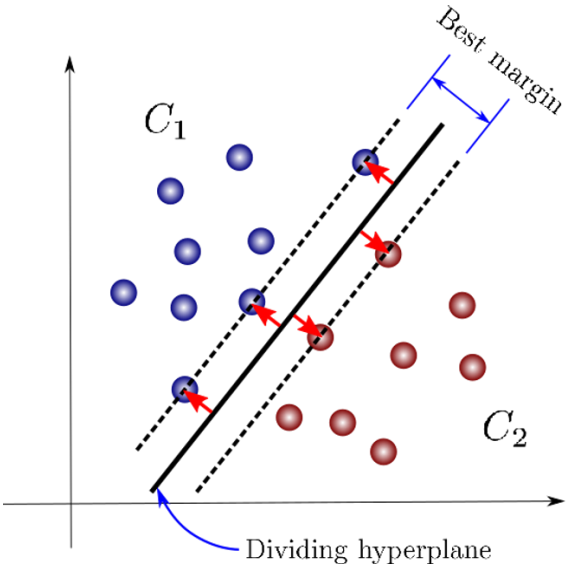
đường hồi quy đến cực tiểu biến thiên gồ ghề hơn và số lần lặp lớn hơn nhưng lại giúp tiết kiệm chi phí về mặt tính toán và thời gian đáng kể so với GD (gradient descent nguyên thủy). Trong thực tế, người ta ưu tiên dùng SGD hơn GD.



GD nguyên thủy *Stochastic GD*

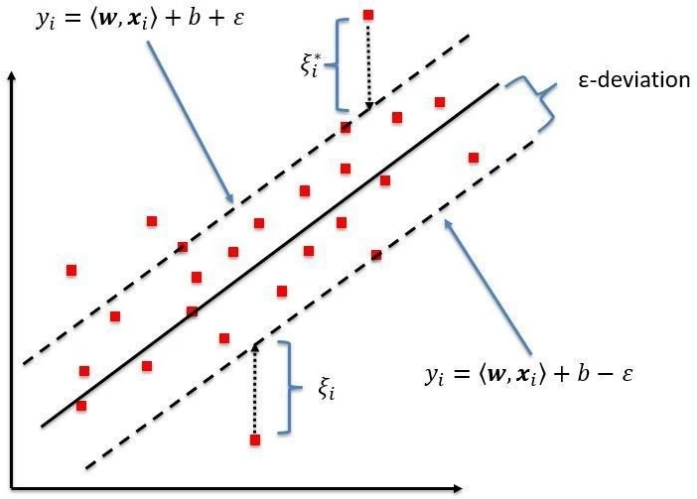
4.1.3. Support Vector Regression:

Support Vector Machine thường được biết đến là được sử dụng cho bài toán phân loại (classify) và hiếm khi được sử dụng trong bài toán hồi quy. Bởi vì mục tiêu của Support Vector là đi tìm một siêu phẳng trong không gian n chiều sao cho margin giữa các điểm dữ liệu trong từng lớp là lớn nhất.



Hình 20. Minh họa SVM phân biệt 2 class dựa trên margin lớn nhất.

Tuy nhiên, khi hầu hết các điểm dữ liệu chủ yếu nằm trong biên độ (margin) tốt nhất về mỗi phía của siêu phẳng thì SVR hay còn gọi là hồi quy vector hỗ trợ có thể được sử dụng để xác định và dự đoán dữ liệu phụ thuộc (y).



Hình 21. Minh họa SVM xác định đường tuyến tính dựa trên margin tốt nhất

Thuật toán:

- **Bước 1:** Thu thập dữ liệu (dạng X và y), chia dữ liệu thành train và test.
- **Bước 2:** Khảo sát dữ liệu và chọn kernel phù hợp, SVR của thư viện sklearn cung cấp một vài kernel như linear, poly, sigmoid, rbf,...Trong bài tập này, nhóm sử dụng linear, poly và rbf để khảo sát và chọn ra kernel phù hợp nhất.
  + ***Linear kernel:*** thuật toán hồi quy tuyến tính đơn giản sử dụng đa thức bậc 1 dạng: y = X.w + bias.
  + ***Poly kernel:*** thuật toán hồi quy sử dụng đa thức bậc cao (bậc n) có dạng: y = X**n.wn + X**m.wm + … + bias với n, m,... là các bậc của đa thức. Poly giúp cho bài toán đáp ứng được việc dự đoán những giá trị output có sự phụ thuộc phức tạp hơn vào các đặc trưng (phi tuyến tính, phụ thuộc theo 1 phương trình đường cong).

+ ***RBF (radial basic function) kernel:*** sử dụng khi dữ liệu là tuyến tính không thể tách rời, hay nói cách khác là phi tuyến tính. RBF kernel hay còn được xem như phân phối Gaussian vì tên nó là radial nghĩa là hướng tâm. Đây được xem là phương pháp rất mạnh và có thể hồi quy được phần lớn các loại dữ liệu.

- **Bước 3:** Xác định các bộ tham số, vì SVR sử dụng các kernel khác nhau để hồi quy nên ứng với mỗi kernel (mỗi mô hình) ta sẽ có một bộ siêu tham số khác nhau.

4.2. Chia dữ liệu

- Tổng toàn bộ dữ liệu sau bước Feature engineering: 796 mẫu.
- Training set: 80% ~ 636 mẫu.
- Testing set: 20% ~ 160 mẫu.

4.3. Tham số huấn luyện

4.3.1. Linear Regression: không có siêu tham số để điều chỉnh.

4.3.2. SGD Regressor:

- *alpha:* 10^{-7} đến 1.
- *loss:* [‘huber’, ‘epsilon_intensitive’, ‘squared_error’].
- *penalty:* [‘l2’, ‘l1’, ‘elasticnet’]
- *learning rate:* [‘constant’, ‘optimal’, ‘invscaling’, ‘adaptive’].
- early_stopping: [True, False].
- *eta0:* 10^{-7} đến 1.

→ Sử dụng GridSearchCV thu được bộ tham số tốt nhất:

- *alpha:* 0.001. Hằng số để nhân trong regularization term.
- *loss:* squared_error. Sai số căn bậc 2 tổng bình phương.
- *penalty:* l2. l2 regularization được sử dụng để tránh overfitting bằng cách cộng thêm vào loss 1 lượng bằng tích của alpha với norm2 của weight.

- *learning rate:* adaptive. eta = eta0 khi mà loss tiếp tục giảm trong quá trình training. Khi loss không giảm trong n (mặc định là 5) lần liên tiếp thì eta giảm đi 5 lần.
- early_stopping: True. Nếu set là True thì khi loss không thể tối ưu được nữa việc training sẽ dừng trước khi số lần lặp chạm đến tối đa.
- *eta0:* 0.001. Giá trị khởi tạo cho learning_rate.
- *max iterations:* mặc định 1000. Số lần lặp tối đa.
- *tol:* mặc định là 10^{-3} (độ lệch giữa giá trị dự đoán và giá trị thực).

4.3.3. Support Vector Regression:

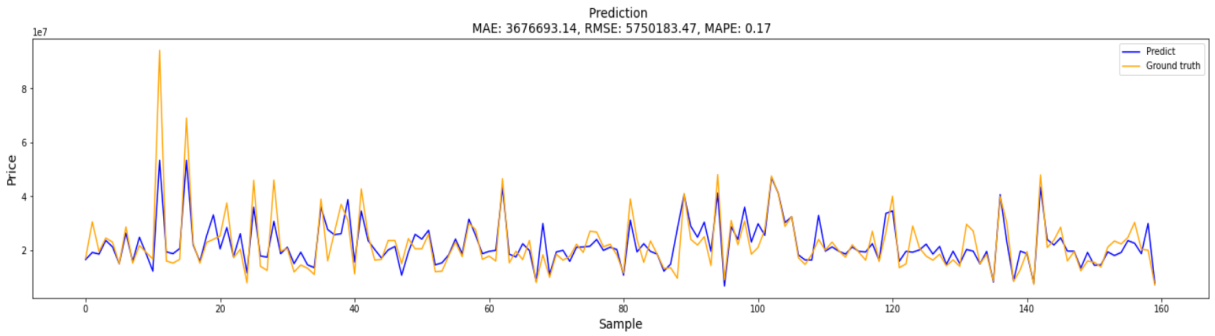
- *kernel:* ['linear', 'poly', 'rbf']
- *C:* [150000, 1000000, 1500000, 5000000, 10000000, 50000000]
- *gamma:* [10^{-10}, 10^{-9}, 10^{-8}, 10^{-7}, 0.000001, 0.00001, 0.0001]

→ ***Sử dụng GridSearchCV thu được bộ tham số tốt nhất:***

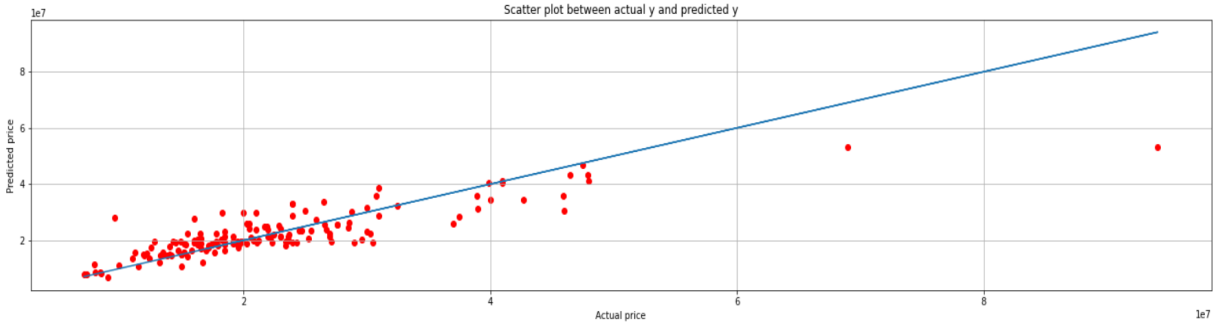
- *kernel:* linear.
- *C:* 50000000. Tham số điều chỉnh cho kĩ thuật regularization. Phải là số dương (bắt buộc), biểu thức regularization tỉ lệ nghịch với C và công thức là l2 penalty.
- *gamma:* 10^{-9}. Tham số quyết định độ cong (trọng số cong) của đường quyết định. Đối với kernel Gaussian thì gamma quyết định

4.4. Đồ thị kết quả

4.4.1. Hồi quy tuyến tính (Linear Regression)

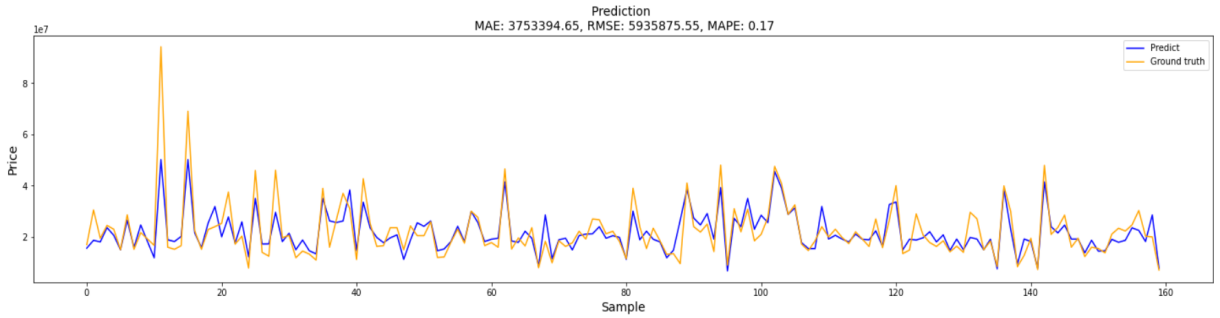


Hình 22. Biểu đồ đường giá dự đoán và giá thực tế trên tập test với 160 mẫu sử dụng Linear Regression.



Hình 23. Biểu đồ scatter giá dự đoán và giá thực trên đường tuyến tính liên hệ giữa 2 giá trị (Linear Regression)

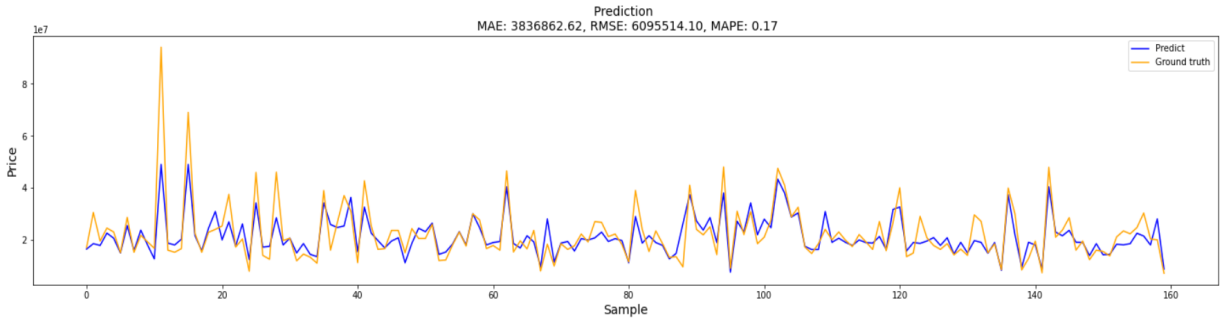
4.4.2. Hồi quy tuyến tính SGD (SGD Regressor)



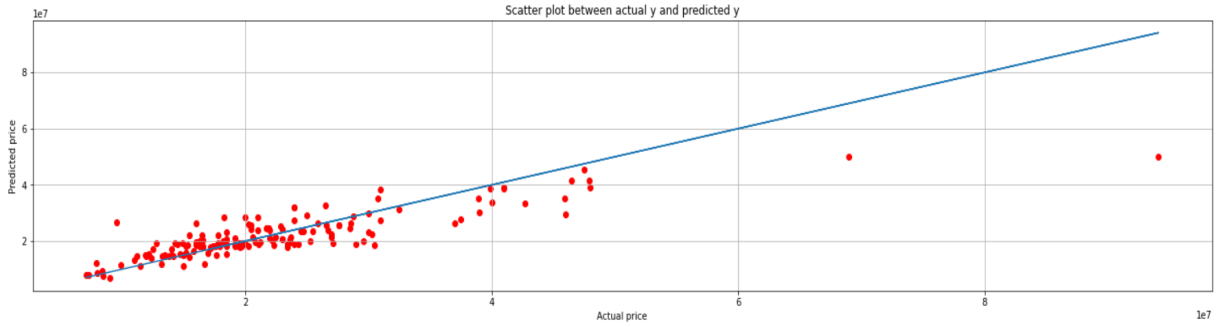
Hình 24. Biểu đồ đường giá dự đoán và giá thực tế trên tập test với 160 mẫu sử dụng SGD Regressor.

Hình 25. Biểu đồ scatter giá dự đoán và giá thực trên đường tuyến tính liên hệ giữa 2 giá trị (SGD Regressor)

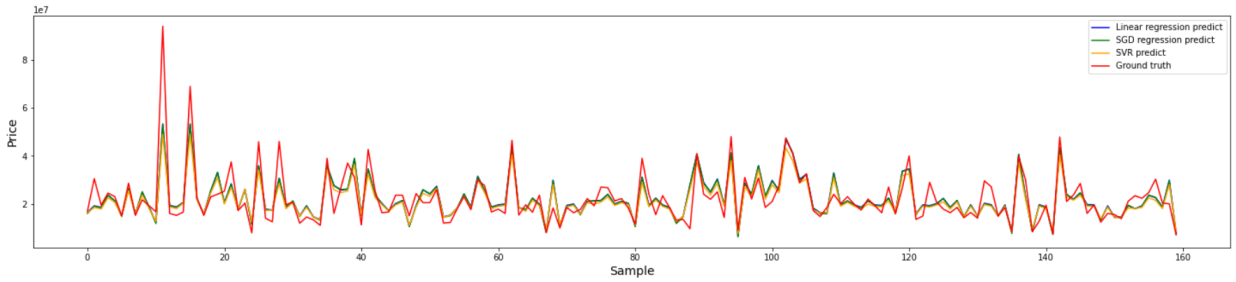
4.4.3. Support Vector Regression (SVR)



Hình 26. Biểu đồ đường giá dự đoán và giá thực tế trên tập test với 160 mẫu sử dụng SVR.



Hình 27. Biểu đồ scatter giá dự đoán và giá thực trên đường tuyến tính liên hệ giữa 2 giá trị (SVR).



Hình 28. Biểu đồ đường giá trị dự đoán (3 mô hình) và giá trị thực tế (đường màu đỏ).

→ Biểu đồ đường:

Đường giá trị dự đoán tương đối trùng khớp với giá trị thực tế, tuy nhiên tại một số điểm giá trị cực đại thì giá trị dự đoán chưa đạt đến được. Phần lớn các giá trị nằm ở tầm từ dưới 20,000,000 đến xấp xỉ 40,000,000.

→ Biểu đồ scatter:

Các điểm nằm tập trung quanh đường tuyến tính với margin tương đối thấp. Giá trị càng tăng thì việc dự đoán càng sai lệch về phía dưới, nghĩa là giá trị dự đoán sẽ càng thiếu gia tốc khi dự đoán với những mức giá cao hơn (từ 40,000,000 trở lên).

4.5. Metrics đánh giá

4.5.1. Khái niệm và mô tả

- ***MAE (Mean absolute error)***: Sai số tuyệt đối trung bình.

$$\frac{\sum_{i=1}^{n} |y_i - y'_i|}{n}$$

- ***RMSE (Root mean square error)***: Sai số trung bình bình phương. Tương tự với MAE nhưng thay vì tính trung bình trị tuyệt đối thì RMSE tính căn bậc hai của trung bình bình phương độ lệch giữa giá trị dự đoán và giá trị thực tế.

$$\sqrt{\frac{\sum_{i=1}^{n} (y_i - y'_i)^2}{n}}$$

- ***MAPE (Mean absolute percentage error)***: Sai số tuyệt đối trung bình phần trăm. MAPE cho biết các giá trị dự đoán trung bình sai lệch bao nhiêu phần trăm so với giá trị thực tế

$$\frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - y'_i}{y_i} \right|$$

4.5.2. Metrics các mô hình

Mô hình	MAE (VND)	RMSE (VND)	MAPE (%)
Linear Regression	3,676,693	5,750,183	16.9
SGD Regressor	3,664,400	5,760,659	16.8
SVR	3,836,862	6,095,514	16.8

5. Kết luận

5.1. Hiệu suất mô hình

Cả 3 mô hình đều cho hiệu suất tương đối đồng đều với nhau.

Trong đó, sai số MAE xấp xỉ 3,000,000 vnđ, RMSE xấp xỉ 5,500,000 vnđ và MAPE là 17%. Tuy nhiên, các metrics này phản ánh sai số trung bình trên toàn bộ dữ liệu test nên cần kết hợp với kết quả trực quan hóa.

Nhóm đánh giá sự sai lệch không phân bố đồng đều trên mọi điểm dữ liệu, trong đó, các mẫu laptop có giá tầm trung (từ 10,000,000 đến 30,000,000 vnđ) được dự đoán khá khớp với giá thực tế trong khi các mẫu laptop có giá cao (từ 40,000,000 vnđ trở lên) thì có sai số khá lớn (lên đến 5,000,000 trên 1 mẫu dữ liệu).

5.2. Giải thích, dự đoán nguyên nhân

- Kích thước dữ liệu nhỏ (xấp xỉ 800 mẫu).
- Mất cân bằng dữ liệu, phần lớn dữ liệu tập trung ở tầm giá thấp và vừa nên mô hình sẽ không học được những mẫu có giá ở tầm cao hơn.

5.3. Hướng phát triển

- Mặc dù dữ liệu thu thập có nhiều thuộc tính nhưng chưa khai thác hết các thuộc tính đó, sau bước lựa chọn thuộc tính thì loại bỏ rất nhiều thuộc tính, đặc biệt là thuộc tính thuộc loại category. Vì vậy cần lựa chọn phương thức mã hóa thích hợp để khai thác các thuộc tính này.
- Thu thập thêm dữ liệu, làm đa dạng dữ liệu
- Thử nghiệm down sample (đối với những mẫu chiếm tỉ lệ lớn trong tập dữ liệu) hoặc up sample (đối với những mẫu dữ liệu chiếm tỉ lệ thấp trong tập dữ liệu) để giúp cân bằng dữ liệu.
- Sử dụng một số mô hình khác như: Random Forest Regressor, XGBoost,...Hiện tại nhóm đã thử nghiệm trên RFR thì đạt được MAPE ~ 15%.

6. Tài liệu tham khảo

[1] Nasima Tamboli, *All You Need To Know About Different Types Of Missing Data Values And How To Handle It*, 01/06/2022

[2] Tiep Vu Huu, *Bài 27: Principal Component Analysis (phần 1/2), https://machinelearningcoban.com/2017/06/15/pca/* ,10/06/2022

[3] Stochastic Gradient Descent, Machine learning cơ bản, (Phần 2/2),

https://machinelearningcoban.com/2017/01/16/gradientdescent2/

[4] L2 and L1 regularization in machine learning, Neelam Tyagi, https://www.analyticssteps.com/blogs/l2-and-l1-regularization-machine-learning