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
Nguyễn Văn Mạnh	20N10	
Phan Tiến Đạt	20N10	
Nguyễn Công Cường	20N10	

ĐÀ NẴNG, 06/2023

TÓM TẮT

Trong cuộc sống ngày nay, giữa sự phát triển của khoa học và công nghệ thì laptop là một vật dụng hết sức phổ biến và không thể thiếu đối với mỗi người dân với sự tiện dụng nhỏ gọn của nó. Nhu cầu để sở hữu một chiếc laptop riêng cho bản thân ngày càng tăng lên, vì vậy trong bài tập này, nhóm chúng em thực hiện đề tài **"Dự đoán giá laptop".** Nhóm thu thập dữ liệu gồm các thuộc tính và giá của mặt hàng laptop từ trang Chợ tốt . 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

STT	Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá
01	Nguyễn Văn Mạnh	Cào dữ liệu từ trang website Làm sạch dữ liệu thô Trực quan hoá dữ liệu	Đã hoàn thành
02	Phan Tiến Đạt	Làm sạch và xử lý dữ liệu trống Xử lý ngoại lệ Chuẩn hóa Lựa chọn thuộc tính Giảm chiều dữ liệu Thể hiện hiệu quả của các quá trình tiền xử lý	Đã hoàn thành
03	Nguyễn Công Cường	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

Nội dung

1. Giới thiệu

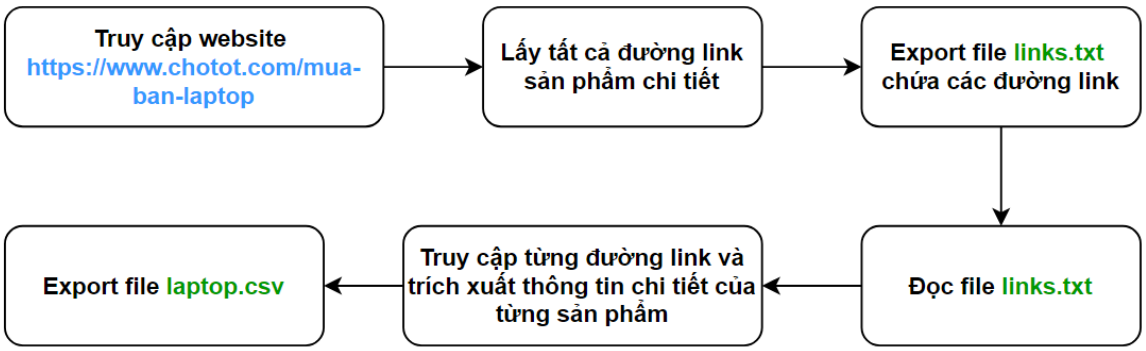
Ngày nay, cùng với sự phát triển của khoa học và công nghệ thì các thiết bị thông minh nói chung và laptop nói riêng đã trở thành một công cụ hữu ích không thể thiếu, phục vụ trong các nhu cầu học tập, làm việc cho đến việc giải trí càng khiến cho thị trường laptop ngày càng nóng lên. Việc lựa chọn một 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 tụi em 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 lấy từ trang website mua bán laptop nổi tiếng là www.chotot.com . 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.

2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

Nhóm lựa chọn sẽ lấy dữ liệu trên trang website bán laptop phổ biến là www.chotot.com
Cụ thể là đường link chi tiết sau: https://www.chotot.com/mua-ban-laptop
Quá trình cào dữ liệu có thể tổng quát thành các bước sau:



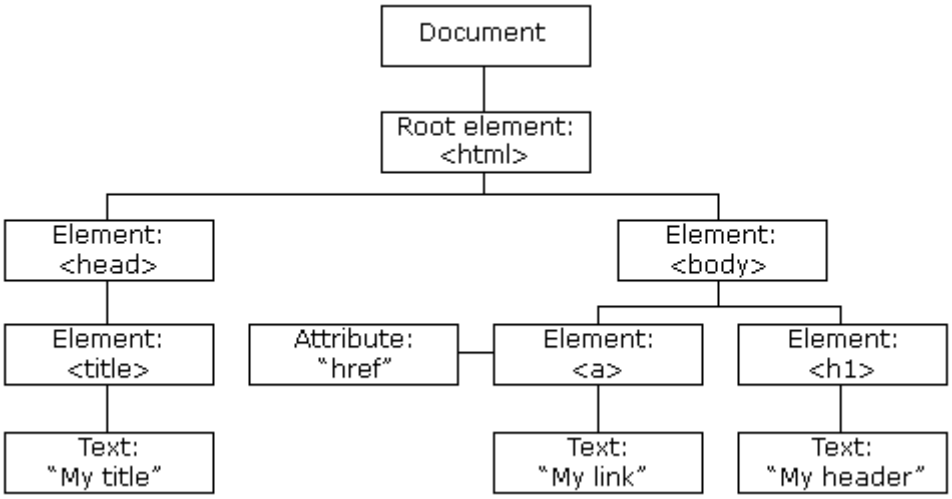
Hình 1. Các bước cào dữ liệu

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 công cụ sau để cào dữ liệu: Selenium 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 và click vào link để tiến hành cào các thông tin chi tiết .



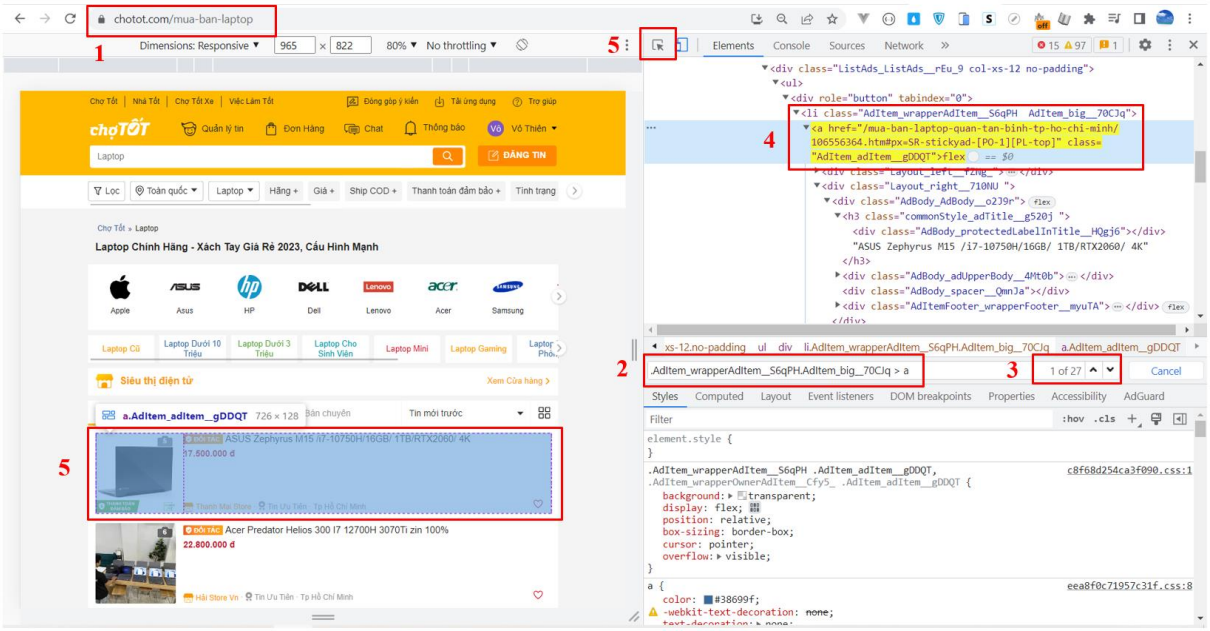
Hình 2. Cấu trúc cây HTML

Ví dụ về cào dữ liệu trang www.chotot.com

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

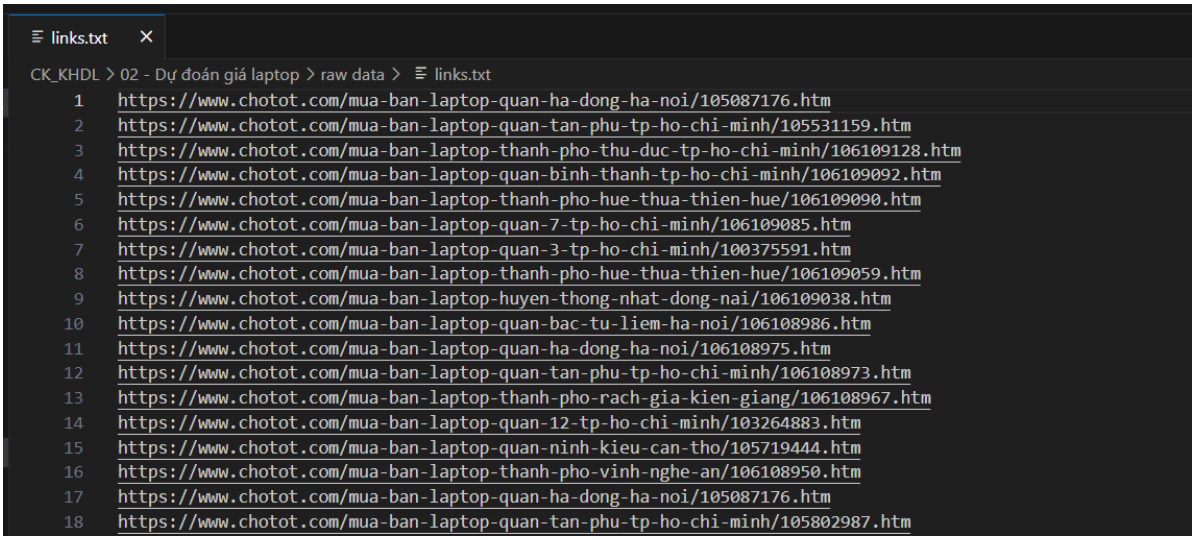
Đầu tiên ta cần phải xác định thủ công đường link sản phẩm nằm ở đâu trong source code HTML:

- Truy cập trang https://www.chotot.com/mua-ban-laptop
- 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 "**.AdItem_wrapperAdItem__S6qPH.AdItem_big__70CJq > 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 www.chotot.com

Tiếp đến tiến hành sử dụng Selenium để giả lập người dùng duyệt website, và sử dụng các phương thức như **find_elements(By.CSS_SELECTOR,"[class='AdItem_wrapperAdItem__S6qPH AdItem_big__70CJq'] > a")** để bóc tách đường link. Lưu tất cả đường link vào file txt và đây chính là kết quả của bước đầu tiên

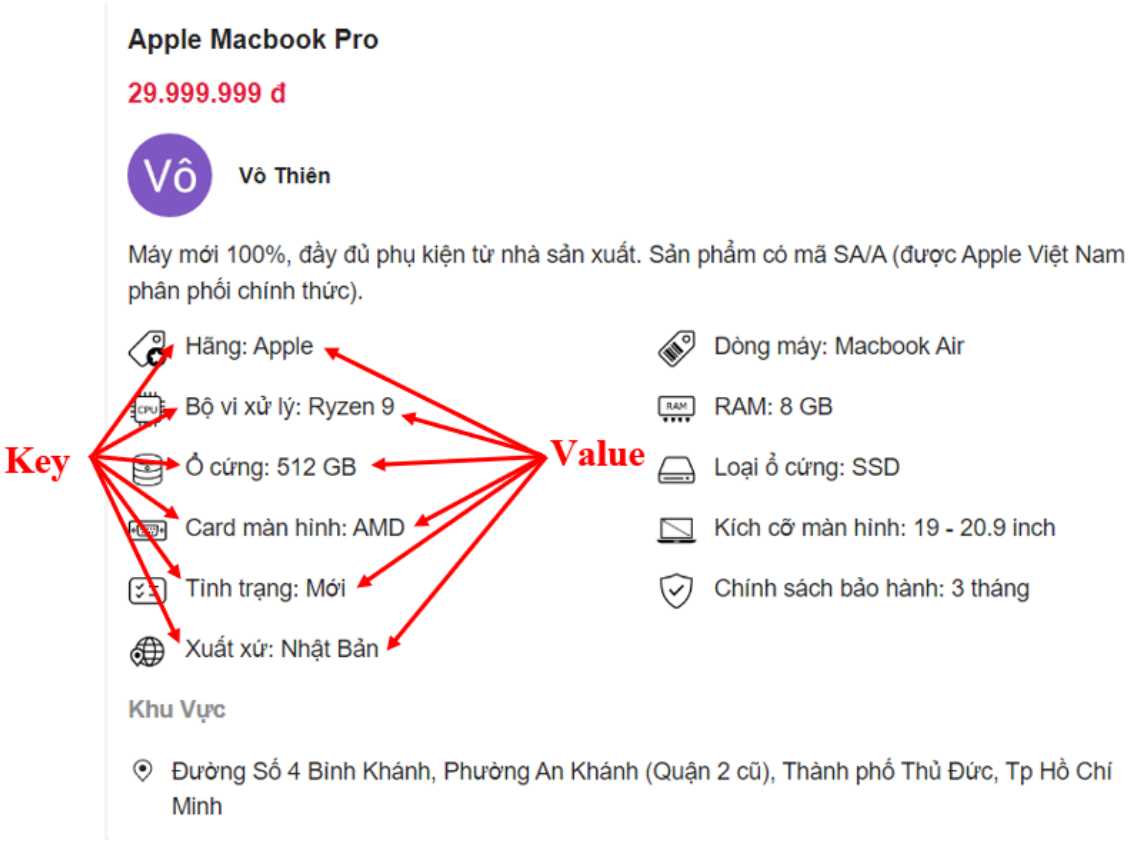


Hình 4. File txt đường link sản phẩm

Bước 2: Lấy thông tin chi tiết sản phẩm

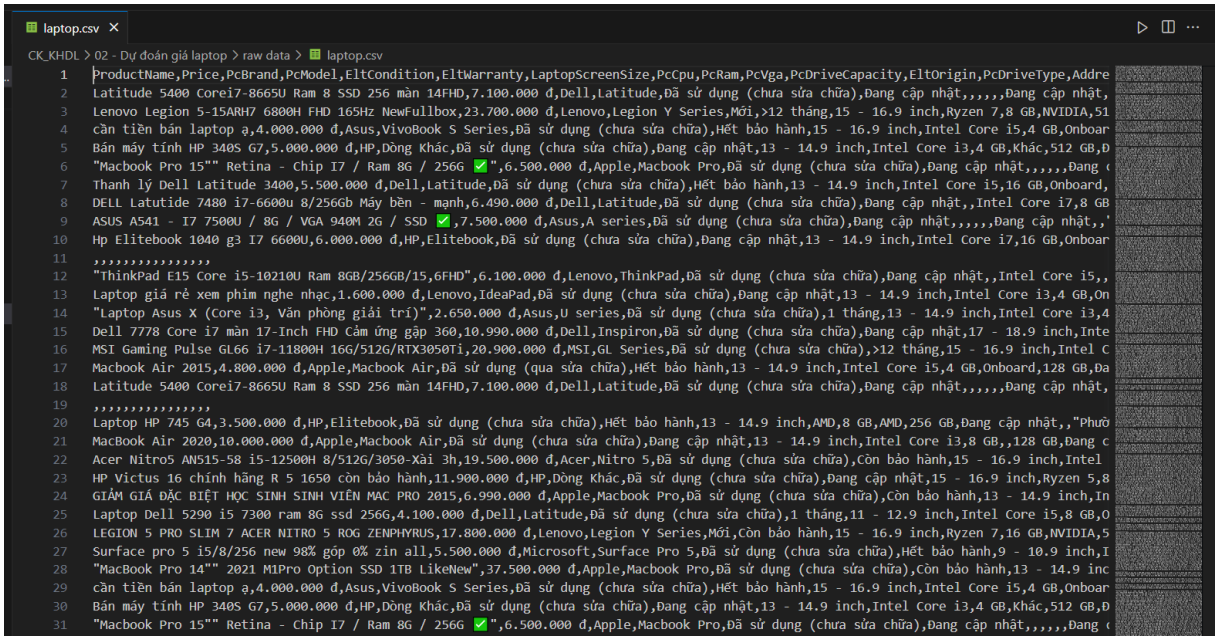
Sau khi có được tất cả đường link sản phẩm, tiến hành duyệt qua từng link sản phẩm và sử dụng Selenium để get source text HTML .

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 để truy xuất dữ liệu bằng CSS Selector như ở các bước trên.



Hình 5. Bảng thông tin chi tiết sản phẩm tại trang www.chotot.com

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 www.chotot.com



Hình 6. 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 loại bỏ các mẫu có nhiều dữ liệu trống, loại bỏ các đặc trưng không cần thiết, gây nhiễu, định dạng và 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:

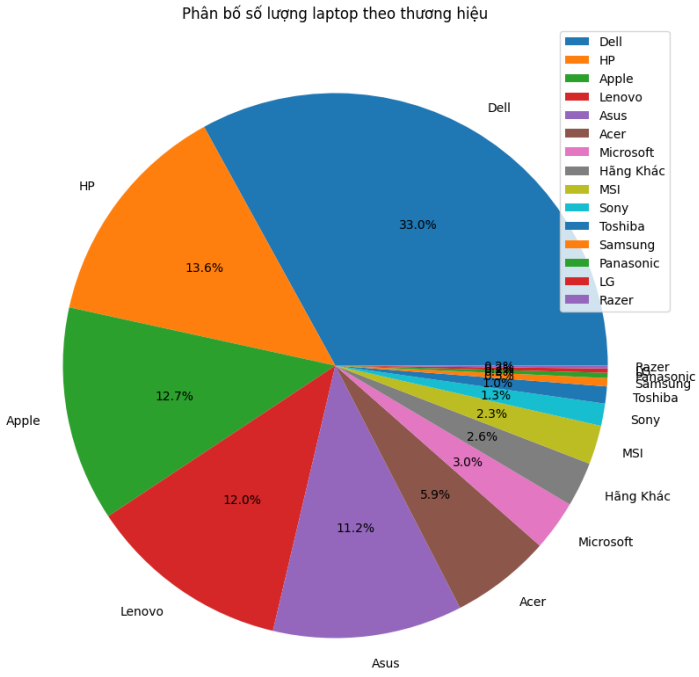
- Số lượng đặc trưng: **15**
- Số lượng mẫu: **12128**

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	Price	Giá bán (vnđ)	Float	0
3	PcBrand	Thương hiệu laptop	String	0
4	PcModel	Mẫu laptop	String	2
5	EltCondition	Trạng thái laptop	String	0
6	EltWarranty	Thời gian bảo hành	Float	3265
7	LaptopScreenSize	Kích thước màn hình	Float	268
8	PcCpu	Tên chip xử lý	String	91
9	PcRam	Dung lượng RAM (GB)	Float	38
10	PcVga	Tên Card màn hình	String	977
11	PcDriveCapacity	Dung lượng ổ cứng	Float	87
12	EltOrigin	Xuất xứ	String	11334
13	Address	Địa chỉ thành phố	String	0
14	ShopRating	Số sao đánh giá	Float	3059

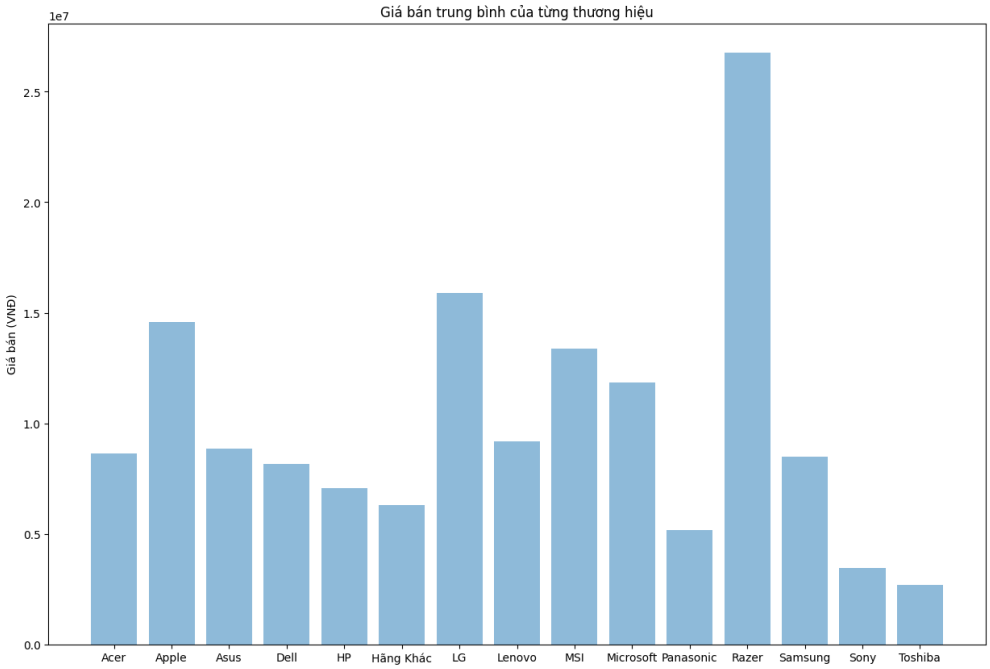
15	Comments	Số bình luận đánh giá	Float	3262

2.3. Trực quan hóa một số đặc trưng



Hình 7. Biểu đồ phân bố số lượng laptop theo thương hiệu

Số lượng laptop được trưng bán trên web theo từng thương hiệu như Dell, HP, Apple, Lenovo, Asus chiếm số lượng lớn. Ngược lại một số thương hiệu có số lượng ít.



Hình 8. Biểu đồ giá bán laptop trung bình của từng thương hiệu

Laptop mang thương hiệu Razer có giá bán trung bình cao nhất tầm khoảng 29 triệu, tiếp đến là LG tầm 15 triệu. Một số hãng laptop khác như Asus, Dell, HP thì có giá bán < 10 triệu.

3. Trích xuất đặc trưng

3.1 Xử lý dữ liệu trống

Trong quá trình thu thập dữ liệu, thường xảy ra trường hợp dữ liệu bị thiếu với tỉ lệ khác nhau tại các thuộc tính khác nhau. Vì vậy, việc làm sạch dữ liệu trống cần được thực hiện một cách cân nhắc. Trước tiên, chúng ta cần xem xét tỉ lệ dữ liệu trống ở mỗi thuộc tính để quyết định cách xử lý phù hợp.

Price	0
PcBrand	0
PcModel	1
EltCondition	0
EltWarranty	2755
LaptopScreenSize	177
PcCpu	70
PcRam	32
PcVga	707
PcDriveCapacity	64
EltOrigin	9299
Address	0
ShopRating	2012
Comments	2174
dtype: int64	

Price	0
PcBrand	0
PcModel	0
EltCondition	0
EltWarranty	250
LaptopScreenSize	25
PcCpu	10
PcRam	3
PcVga	65
PcDriveCapacity	3
EltOrigin	919
Address	0
ShopRating	206
Comments	227
dtype: int64	

Hình 9. Dữ liệu trống trong BigDS và SmallDS

Với các thuộc tính có tỉ lệ dữ liệu trống quá cao, việc lấp đầy dữ liệu thiếu có thể không đem lại hiệu quả mà chỉ gây thêm sai lệch trong quá trình huấn luyện và dự đoán. Trong trường hợp này, chúng ta nên xem xét loại bỏ các trường dữ liệu đó khỏi tập dữ liệu. Điều này giúp giảm bớt nhiễu và tối ưu quá trình xử lý dữ liệu. Như hình trên thì thuộc tính "EltOrigin" có quá nhiều dữ liệu trống (919/1000) ở SmallDS và (9299/10000) ở BigDS, nên cần được loại bỏ.

Còn đối với những thuộc tính có tỉ lệ dữ liệu trống không quá cao, chúng ta có thể tiến hành xử lý và điền dữ liệu trống vào các trường đó. Có một số phương pháp thường được sử dụng để xử lý dữ liệu trống như sử dụng giá trị trung bình, giá trị trung vị, hoặc giá trị xuất hiện nhiều nhất để điền vào các vị trí thiếu dữ liệu. Cách tiếp cận

này giúp giữ lại các mẫu dữ liệu quan trọng và đồng thời giảm thiểu tác động của dữ liệu thiếu lên quá trình phân tích.

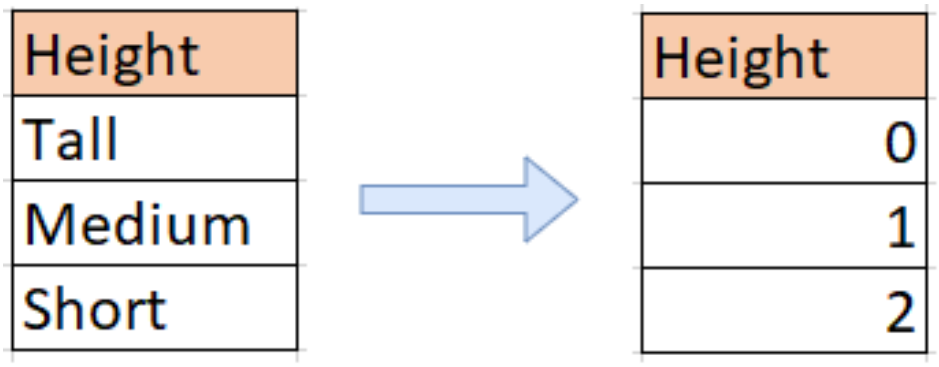
Đối với dữ liệu thuộc kiểu danh mục (category) không thể dùng các phương pháp tính toán để tính dữ liệu trống cho nên dữ liệu trống loại này sẽ được lấp đầy bằng các random từ các mẫu có dữ liệu. Phương pháp này giúp thuộc tính có thể giữ đúng phân bố của mình sau khi lấp đầy vị trí trống.

Đối với dữ liệu số (numerical) thì có thể sử dụng nhiều phương pháp điền dữ liệu trống khác nhau như mean, median, mode,… Ngoài các kĩ thuật đó, đề tài còn sử dụng đến Iterative Imputation, đây là một kỹ thuật điền giá trị thiếu trong dữ liệu bằng cách sử dụng mô hình dự đoán

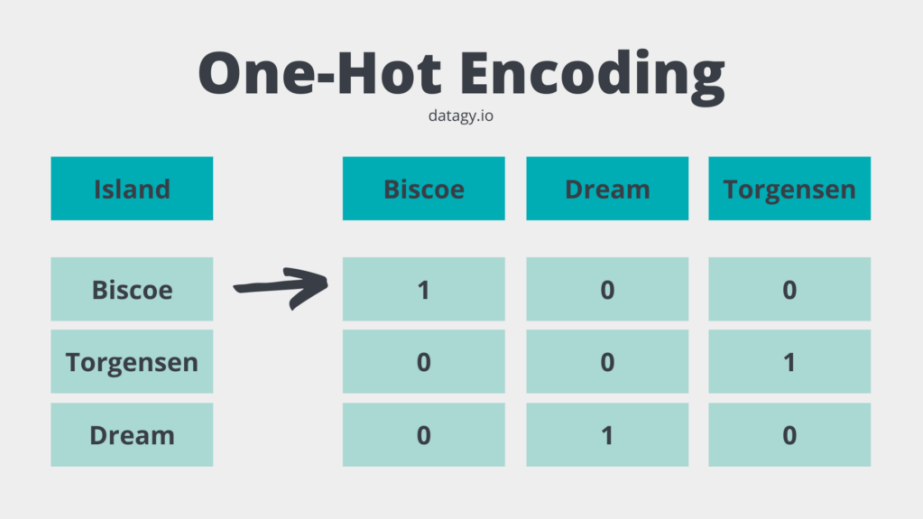
3.2 Mã hóa dữ liệu

Các dữ liệu kiểu số sẽ được giữ nguyên, trong khi đó dữ liệu kiểu category cần được mã hóa sang dạng số. Điều này là cần thiết để có thể huấn luyện và dự đoán vì các mô hình sẽ không làm việc với các dữ liệu kiểu chuỗi ký tự.

Việc mã hoá dữ liệu kiểu category có nhiều phương pháp. Đề tài sử dụng 2 phương pháp đó là LabelEncoder và One-Hot Encoder



Hình 10. Minh hoạ LabelEncoder

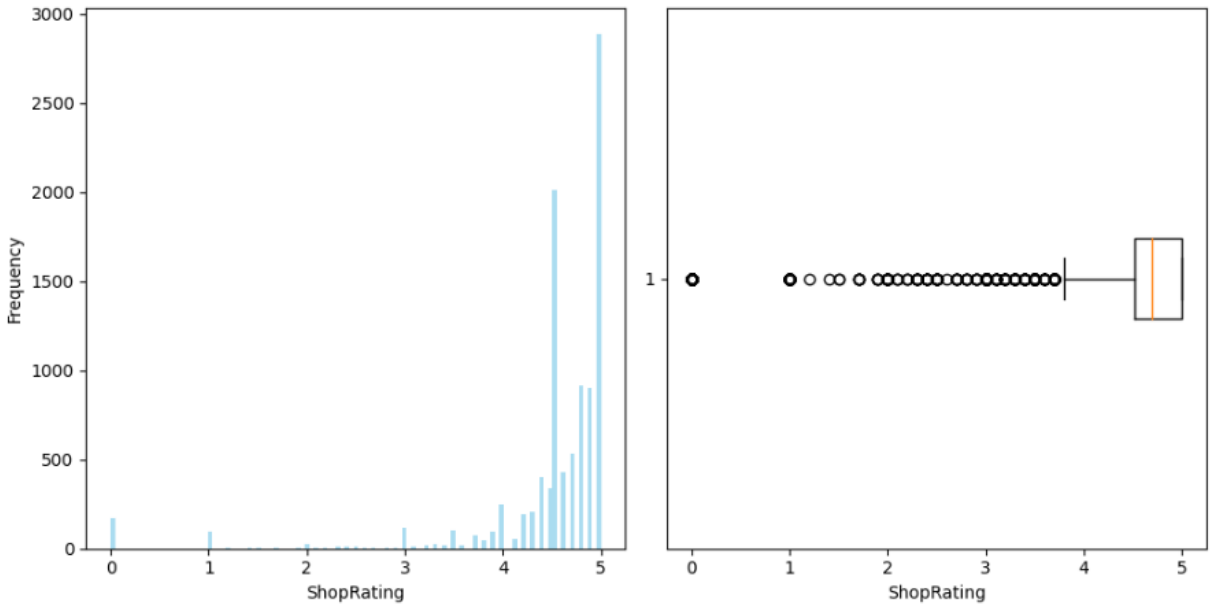


Hình 11. Minh hoạ one-hot Encoder

Đối với dữ liệu hiện tại, các thuộc tính kiểu category sử dụng mã hoá này: EltWarranty, LaptopScreenSize, PcRam, PcDriveCapacity, ShopRating, Comments, PcBrand, PcModel, EltCondition, PcCpu, PcVga

3.3 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 phân bố lệch.



Hình 12. Histogram và boxplot của thuộc tính ShopRating

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

Histogram để xác định loại phân bố của thuộc tính, đối với ví dụ trên thì đây là dạng phân bố lệch

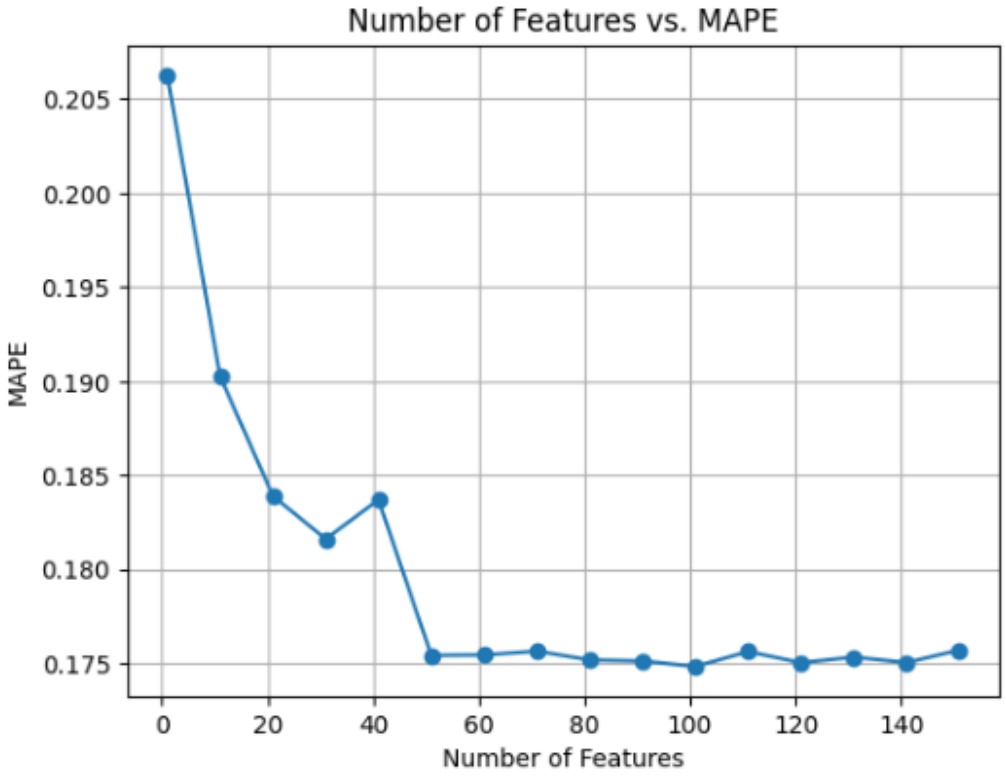
Tiếp tục sử dụng biểu đồ boxplot để xem xét việc xử lý ngoại lệ, Như ví dụ trên thì có thể đưa ra kết luận rằng cần xử lý ngoại lệ phân bố lệch cho thuộc tính ShopRating

3.4 Chuẩn hóa

Chuẩn hóa dữ liệu (Normalization) là một quá trình điều chỉnh dữ liệu để đưa chúng về một phạm vi hoặc định dạng chung nhất để đem đến ổn định và cải thiện trong hiệu suất mô hình

Bảy loại chuẩn hoá phổ biến được áp dụng: StandardScaler, MinMaxScaler, RobustScaler, MaxAbsScaler, Normalizer, QuantileTransformer, PowerTransformer

3.5 Lựa chọn đặc trưng



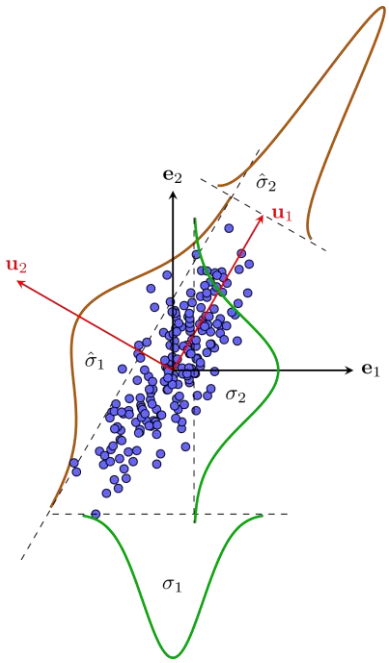
Hình 13. Mức độ lỗi theo số lượng thuộc tính

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 tốt nhất của mô hình.

Đề tài áp dụng cả RFE (Recursive Feature Elimination) và SelectKBest là hai phương pháp phổ biến được sử dụng trong việc chọn đặc trưng (feature selection). RFE sẽ hoạt động dựa trên mô hình dự đoán, còn SelectKBest không phụ thuộc vào mô hình mà sẽ tính toán dựa theo độ tương quan giữa các đặc trưng và biến mục tiêu

3.6 Giảm chiều dữ liệu

PCA (Principal Component Analysis) là một phương pháp quan trọng trong xử lý dữ liệu và machine learning với nhiều lợi ích. Một trong những lợi ích chính của PCA là khả năng giảm chiều dữ liệu. Thay vì làm việc với nhiều biến ban đầu, PCA trích xuất các thành phần chính, đại diện cho các hướng tương quan chính trong dữ liệu. Việc giảm số chiều này giúp làm giảm phức tạp tính toán, tăng tốc độ xử lý, và giảm yêu cầu bộ nhớ. Tuy nhiên việc áp dụng PCA không lúc nào cũng đem lại hiệu quả cho mô hình.



Hình 14. Mô tả phương pháp PCA

Việc áp dụng phương pháp PCA trong đề tài đã không tối ưu được hiệu suất của mô hình trong bất kỳ số lượng chiều nào.

```
PCA 1 : độ lỗi là 0.6024928712468125 cải thiện -0.4539189431425624
PCA 11 : độ lỗi là 0.5995589153239442 cải thiện -0.45098498721969416
PCA 21 : độ lỗi là 0.5470124643171436 cải thiện -0.3984385362128935
PCA 31 : độ lỗi là 0.5460914329008166 cải thiện -0.39751750479656656
PCA 41 : độ lỗi là 0.5353811626090156 cải thiện -0.3868072345047655
PCA 51 : độ lỗi là 0.540342612601335 cải thiện -0.3917686844970849
PCA 61 : độ lỗi là 0.5497112671059537 cải thiện -0.4011373390017037
PCA 71 : độ lỗi là 0.524417805059933 cải thiện -0.3758438769556829
PCA 81 : độ lỗi là 0.5268129344577466 cải thiện -0.3782390063534965
PCA 91 : độ lỗi là 0.47011207736899374 cải thiện -0.32153814926474367
PCA 101 : độ lỗi là 0.4131018287371292 cải thiện -0.2645279006328791
```

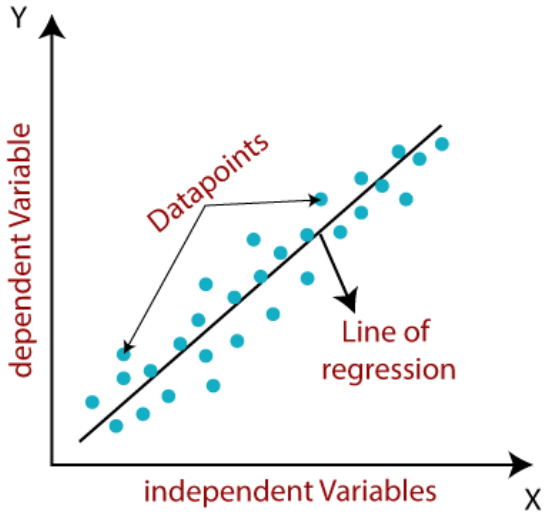
4. Mô hình hóa dữ liệu

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.

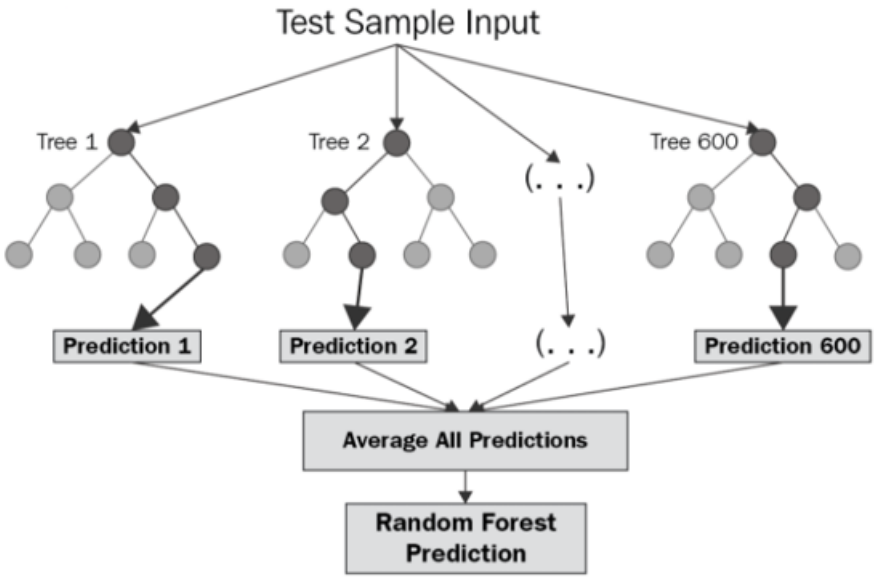


Hình 15. Minh họa mô hình hóa Hồi quy tuyến tính.

4.1.2. Random Forest:

Random Forest Regressor là một thuật toán học máy được sử dụng trong bài toán dự đoán giá trị liên tục, hay còn gọi là bài toán hồi quy. Được xây dựng dựa trên khái niệm "rừng ngẫu nhiên", thuật toán này kết hợp nhiều cây quyết định (decision trees) để tạo ra một mô hình dự đoán mạnh mẽ và ổn định.

Khi thực hiện dự đoán, Random Forest Regressor tính trung bình giá trị dự đoán của tất cả các cây quyết định trong rừng để cho ra giá trị dự đoán cuối cùng. Điều này giúp cân nhắc và hạn chế ảnh hưởng của các cây quyết định "nhiễu" và giúp tạo ra một mô hình mạnh mẽ và ổn định hơn.



Hình 16. Minh họa mô hình hóa Hồi quy random forest.

4.2. Chia dữ liệu

- Tổng toàn bộ dữ liệu sau bước Feature engineering:
  + Big Data ~10000 mẫu, Small Data ~1000 mẫu

- Training set: 70%
  + Big Data ~7000 mẫu, Small Data ~700 mẫu
- Testing set: 30%
  + Big Data ~3000 mẫu, Small Data ~300 mẫu

- Training Validation set:

  + Big Data ~4900 mẫu, Small Data ~490 mẫu

- Test Validation set:

  + Big Data ~2100 mẫu, Small Data ~210 mẫu

4.3. Tham số huấn luyện

4.3.1. Linear Regression:

Không có siêu tham số để điều chỉnh.

4.3.2. Random Forest Regressor:

- n_estimators: Số lượng cây trong rừng ngẫu nhiên.

Giá trị này thường được chọn lớn để đảm bảo tính ổn định của dự đoán.

Ví dụ: [100, 200].

- max_depth: Độ sâu tối đa của cây.

Giới hạn độ sâu này giúp tránh việc mô hình quá phức tạp và overfitting.

Ví dụ: [None, 5, 10].

- min_samples_split:

Số lượng mẫu tối thiểu cần có trong mỗi nút để tiếp tục quá trình chia.

Giá trị này có thể giúp kiểm soát độ phức tạp của cây.Ví dụ: [2, 5].

- min_samples_leaf: Số lượng mẫu tối thiểu cần có trong mỗi lá để xem xét một phân vùng là một lá.

Giá trị này có thể giúp tránh việc quá khớp (overfitting). Ví dụ: [1, 2, 4].

Riêng với BigData ta sử dụng 'n_estimators': [10, 20, 50], 'max_depth': [None, 1, 2], để giảm độ phức tạp của mô hình

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

- **Small Data:**

  Best Hyperparameters:

  {'max_depth': None,'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}

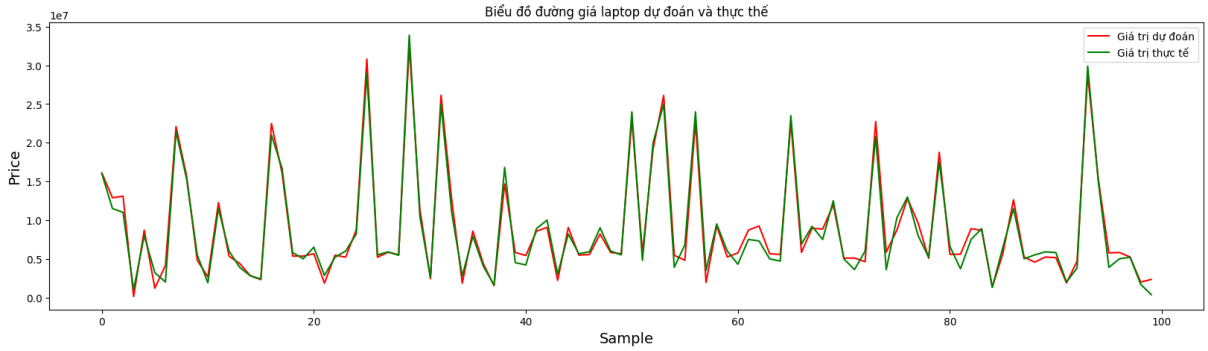
- **Big Data:**

  Best Hyperparameters:

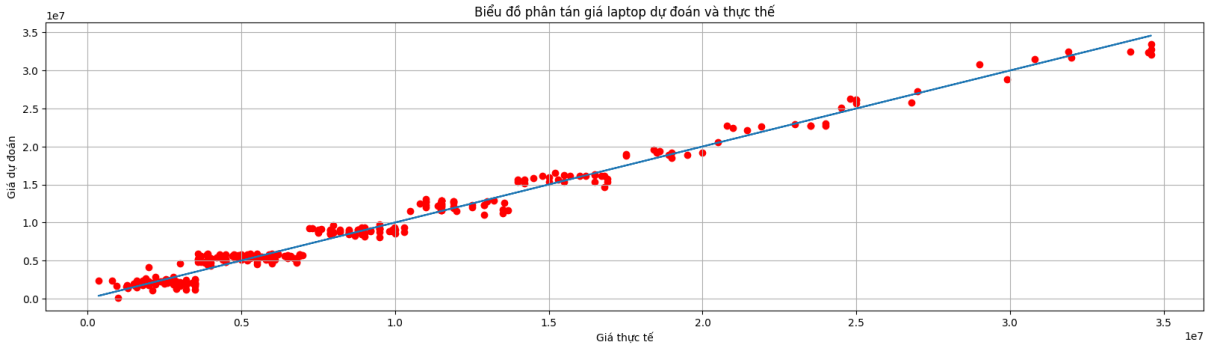
  {'max_depth': None, 'n_estimators': 50}

4.4. Đồ thị kết quả

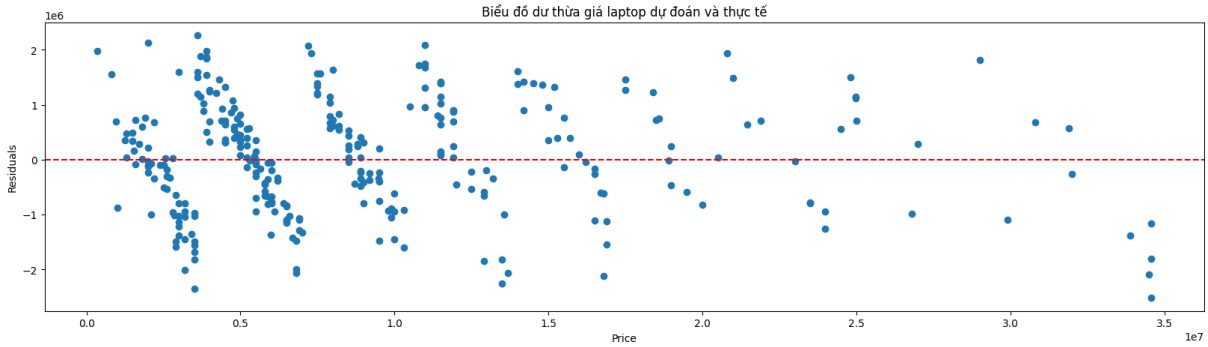
4.4.1. Hồi quy tuyến tính (Linear Regression)



Hình 17. Biểu đồ đường giá dự đoán và giá thực tế trên tập test với 100 mẫu sử dụng Linear Regression.

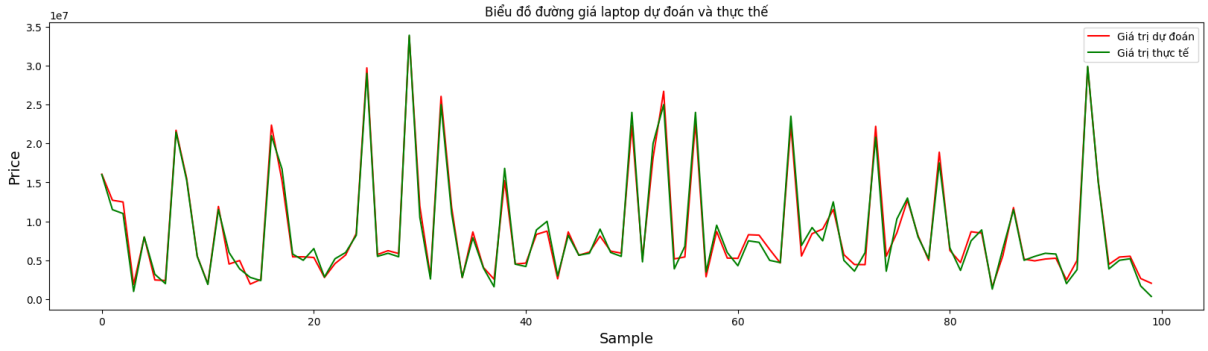


Hình 18. Biểu đồ scatter giá dự đoán và giá thực trên đường tuyến tính liên hệ giữa 2 giá trị (Linear Regression)

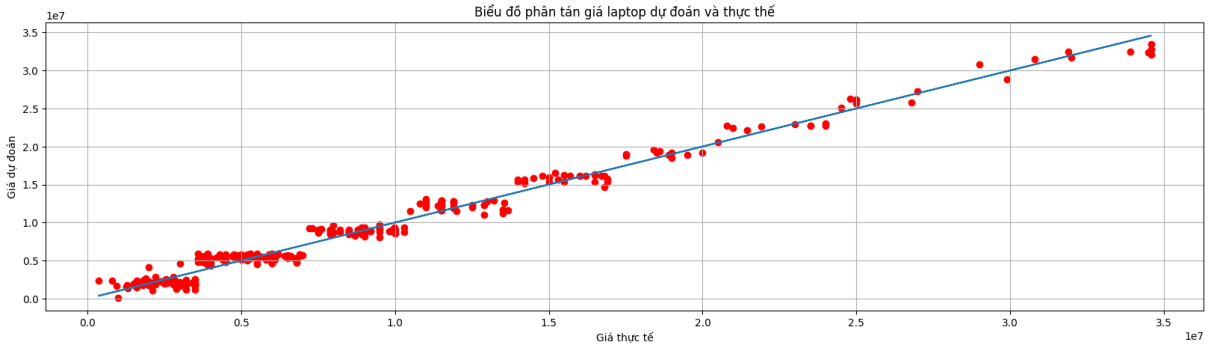


Hình 19. Biểu đồ dư thừa giá dự đoán và giá thực (Linear Regression)

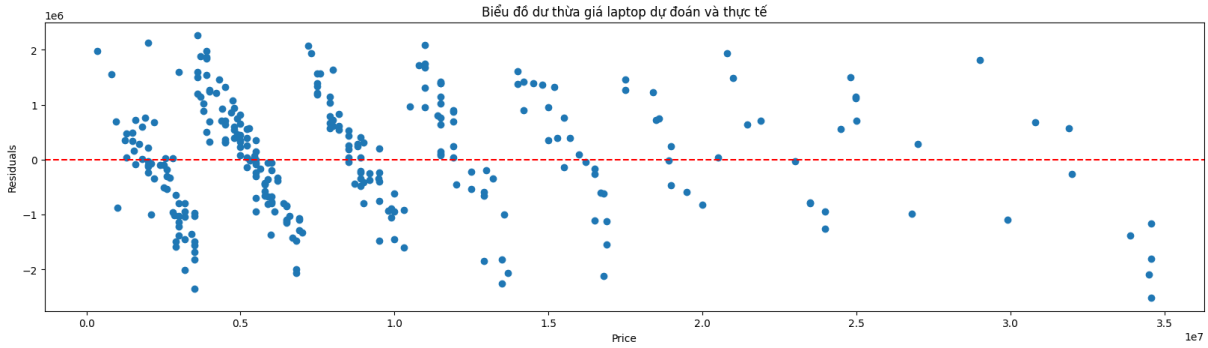
4.4.2. Hồi quy Random Forest (Random Forest Regressor)



Hình 20. Biểu đồ đường giá dự đoán và giá thực tế trên tập test sử dụng Random Forest Regressor.



Hình 21. Biểu đồ scatter giá dự đoán và giá thực (Random Forest Regressor)



Hình 22. Biểu đồ dư thừa giá dự đoán và giá thực (Random Forest Regressor)

→ Biểu đồ đường:

Đường giá trị dự đoán giá laptop của cả hai mô hình tương đối trùng khớp với giá trị thực tế.

→ Biểu đồ phân tán

- Giá dự đoán phân tán xung quanh đường thẳng đại diện cho giá dự đoán khớp với giá thực tế
- Giá laptop càng lên cao số lượng phân bố càng thưa thớt (có lẽ là do số lượng laptop có giá cao không nhiều)

→ Biểu đồ dư thừa:

- Ta có thể nhìn thấy được giá laptop dự đoán sai dao động trong khoảng 2 triệu.
- Phân bố các giá trị dự đoán sai chủ yếu tập trung trong đoạn từ âm 1 đến dương 1 triệu

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

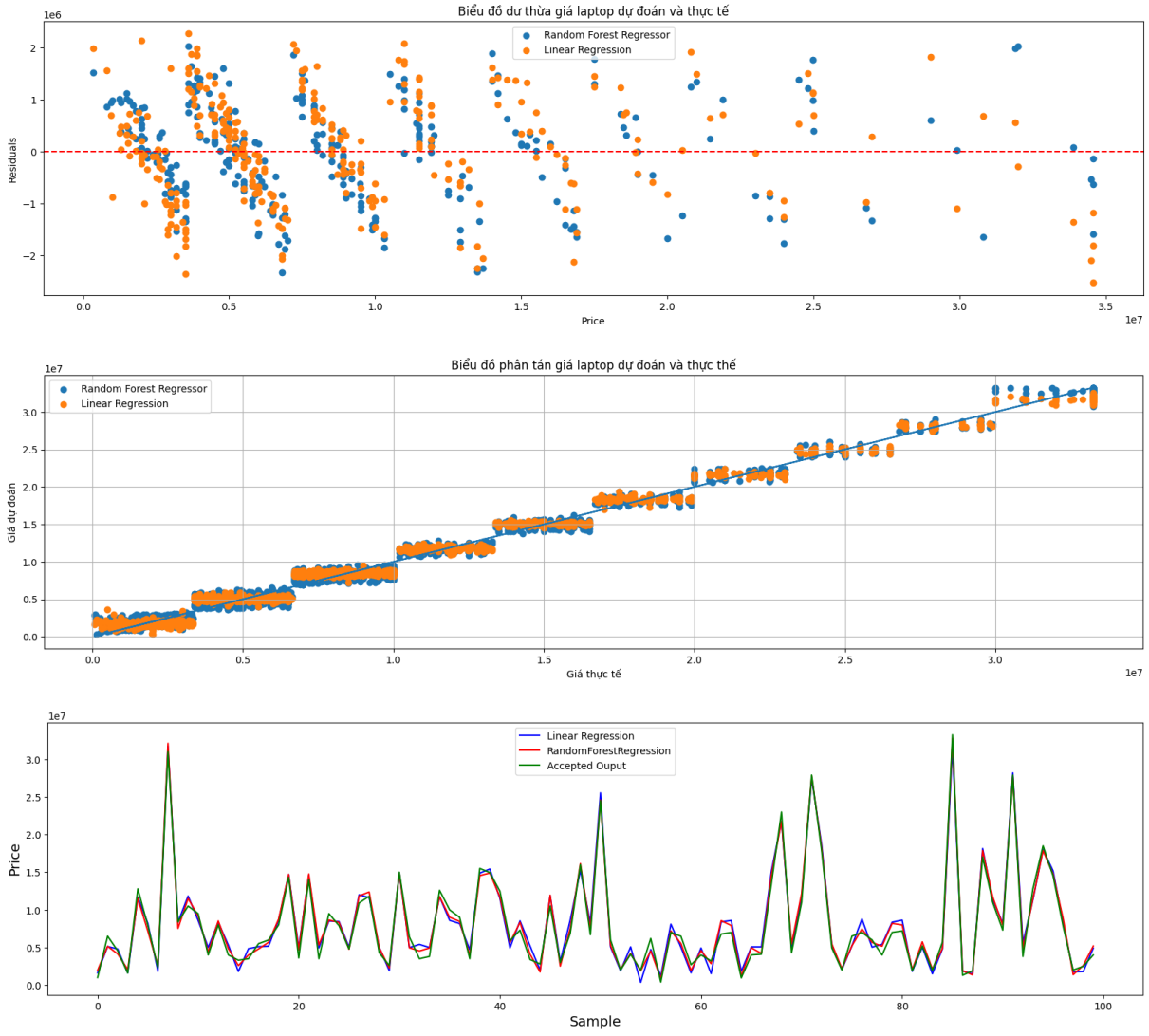
Small Data

Mô hình	MAE (VND)	RMSE (VND)	MAPE (%)
Random Forest Regressor	753,327.2	929,044.8	14.9
Linear Regression	833,749.8	1015,213.2	16.7

Big Data

Mô hình	MAE (VND)	RMSE (VND)	MAPE (%)
Random Forest Regressor	707,041.0	878,669.2	17.5
Linear Regression	827,058.1	965,991.4	19.6

4.5.3 Đồ thị thể hiện hiệu suất



Hình 23. Biểu đồ dư thừa(Small data), phân tán, đường giá dự đoán và giá thực (BigData)

Qua ba biểu đồ trên ta có thể thấy sự chênh lệch kết quả của 2 mô hình nhờ biểu đồ dư thừa và biểu đồ đường:

- **Biểu đồ dư thừa:** giá dự đoán dư thừa so với trục x thì Linear Regresessor bị lệch ra ngoài nhiều hơn với Random Forest Regression
- **Biểu đồ phân tán:** không nhìn ra được sự khác biệt lớn
- **Biểu đồ đường:** đường dự đoán của mô hình Random Forest Regression gần với đường giá thực tế hơn so với đường mô hình Linear Regression

5. Kết luận

5.1. Tổng quát

Trong bài toán dự đoán giá laptop, với 2 loại dataset gồm 1000 và 10000 mẫu được crawl từ web chottot.com:

- Phần trích xuất đặc trưng việc áp dụng các kĩ thuật như xử lí dự liệu trống, xử lí ngoại lệ, chuẩn hóa dữ liệu và lựa chọn đặc trưng không tác động lớn vào kết quả dự đoán.

- Ngược lại trong mô hình hóa dữ liệu việc chọn các mô hình và thuật toán phù hợp có ảnh hưởng đến kết quả dự đoán.

- Cụ thể mô hình Random Forest Regression cho kết quả dự đoán tốt hơn ~2% so với dự đoán bằng mô hình Linear Regression

5.2. Hướng phát triển

- 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: SVR, SGD Regressor, XGBoost,...

6. Tài liệu tham khảo

[1] Selenium with Python: Selenium with Python

[2] Random Forest Regressor: Random Forest Regressor

[3] Iterative Imputer: Iterative Imputer

[4] Linear Regression: Linear Regression