

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



PHÂN TÍCH CÁC YẾU TỐ ẢNH HƯỞNG ĐẾN
GIÁ LAPTOP VÀ DỰ ĐOÁN

Sinh viên thực hiện:

STT	Họ tên	MSSV
1	Trịnh Thế Hiển	20521310
2	Nguyễn Hùng Trung Hiếu	20521323
3	Lưu Quang Tiến Hoàng	20521342
4	Nguyễn Thành Nhân	20521701

TP. HỒ CHÍ MINH – 12/2022

1. GIỚI THIỆU

Hiện nay, thời đại công nghệ ngày càng phát triển mạnh mẽ, dẫn đến mức sống của con người ngày càng được nâng cao, các sản phẩm công nghệ tiên tiến phục vụ cho đời sống con người đang nhận được sự ủng hộ và thúc đẩy rất lớn. Đối với học sinh sinh viên ngày nay, thậm chí là người đã đi làm, chiếc laptop là một thiết bị rất cần thiết để phục vụ cho học tập và công việc. Vì vậy ở thời điểm hiện tại và trong tương lai không xa, Laptop sẽ là một sản phẩm cực kì thân thuộc và không thể thiếu trong đời sống của mỗi người. Do đó việc tìm hiểu về cấu hình, linh kiện và giá cả của sản phẩm Laptop là điều rất cần thiết. Vì vậy, trong khuôn khổ đồ án này, chúng tôi thực hiện đề tài này để thực hiện phân tích trực quan các yếu tố ảnh hưởng đến giá cả cũng như xây dựng các mô hình dự đoán cho bộ dữ liệu chúng tôi tự thu thập từ các trang web thương mại điện tử và linh kiện điện tử. Cụ thể, chúng tôi đã thực hiện nhiều phân tích đánh giá bằng công cụ ngôn ngữ lập trình chính là Python cùng các thư viện hỗ trợ phân tích như, xử lý dữ liệu như: Pandas, Numpy,...; các thư viện hỗ trợ trực quan trực quan như: Matplotlib, Seaborn, Plotly, ...; kết hợp suy luận từ vốn hiểu biết và các khía cạnh dữ liệu liên quan trong bộ dữ liệu. Chúng tôi xây dựng bộ dữ liệu bao gồm về thông số cấu hình, giá cả của những chiếc laptop và xây dựng mô hình máy học dự đoán. Bước đầu cho kết quả tốt nhất ở mô hình Random Forest với kết quả theo độ đo R^2 là 0.75

2. DỮ LIỆU

2.1. Quy trình thu thập bộ dữ liệu

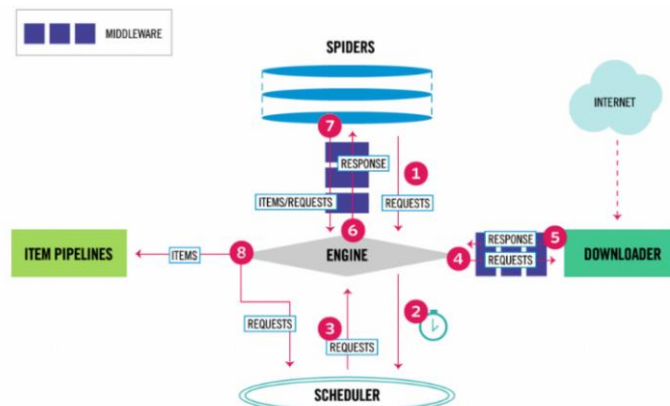
Chúng tôi quyết định tự xây dựng bộ dữ liệu và thực hiện thu thập dữ liệu từ 3 nguồn khác nhau, bao gồm:

- [amazon.com](https://www.amazon.com) (là nguồn chính để lấy danh sách và thông tin laptop).
- cpubenchmark.net và notebookcheck.net (thu thập thông tin về xếp hạng cũng như hiệu năng của từng CPU và Card đồ họa của Laptop)

2.1.1. Cách thức thu thập:

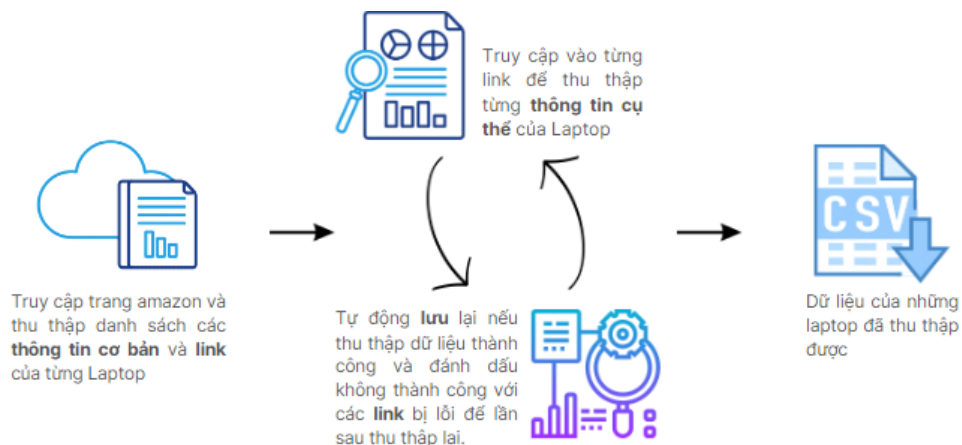
- Đối với 2 trang web để thu thập thông tin về thuộc tính CPU và Card đồ họa, chúng tôi sử dụng thư viện bs4 để trích xuất thông tin từ file HTML.
- Đối với trang web Amazon vì có tính chất phức tạp hơn nên chúng tôi sử dụng framework scrapy. Đây là framework chuyên về thu thập dữ liệu, giúp code có hệ thống hơn và có tốc độ vượt trội so với các thư viện khác còn lại như Selenium, Requests...
- Khi mới tiến hành thu thập dữ liệu, nhóm gặp lỗi “503 (Service Unavailable)” khi Amazon có hệ thống để phát hiện trình cào dữ liệu và chặn. Tận dụng sức mạnh của scrapy, để giải quyết vấn đề, chúng tôi tùy chỉnh phần Downloader Middlewares (phần xử lý các request được đẩy vào từ Engine và các response được tạo ra từ Downloader) để mỗi request gửi đi

đều có user-agent khác nhau – lấy ngẫu nhiên từ danh sách các user-agent giả mà chúng tôi tìm được.



Hình 1. Framework scrapy: Quy trình hệ thống

2.1.2. Quy trình thu thập:



Hình 2. Quy trình thu thập dữ liệu

Đầu tiên ta truy cập vào danh sách *Laptops* trên trang Amazon. Danh sách này bao gồm 2,107 laptop ứng với 2,107 đường link, được chia thành 263 trang. Truy cập từng link vừa rồi để lấy thông tin cấu hình từng máy. Tại một số thời điểm bất kỳ, trình cào gặp phần kiểm tra “Bạn có phải là robot?”, nên sẽ bị lỗi, nên những link này sẽ được lưu và đánh dấu để lần sau thực hiện lại. Cuối cùng, ta được data thô dạng dataframe có độ lớn 2,106 x 79 cột.

2.2. Tiền xử lý dữ liệu

Phần tiền xử lý gồm các bước:

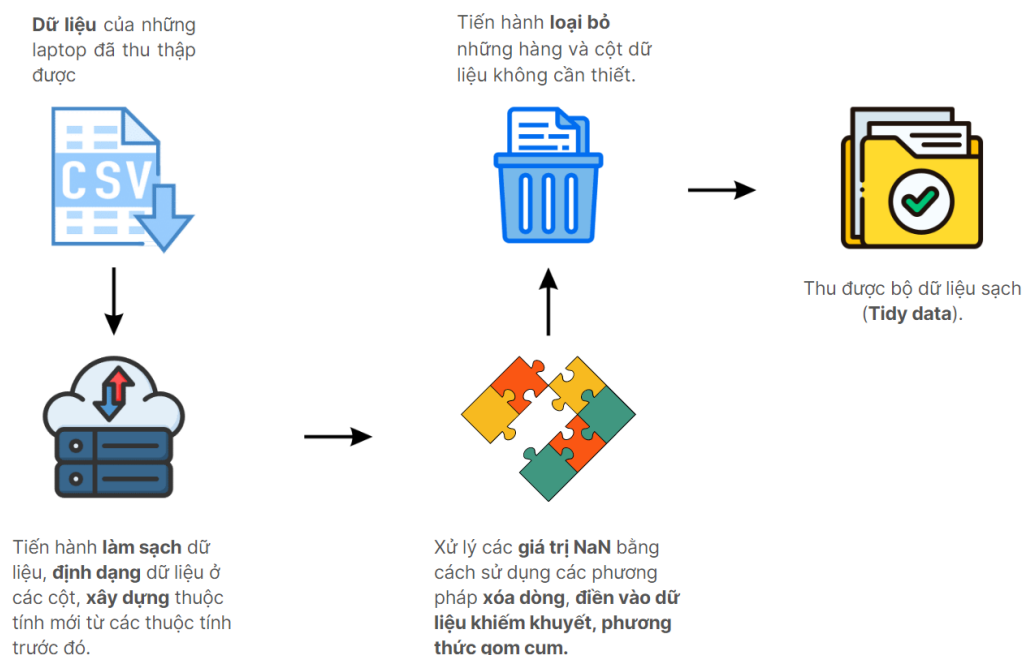
- Xóa cột dư thừa & cột có tỷ lệ missing data cao
- Trích thông tin cần thiết từ một số cột, tạo cột mới, feature engineering
- Điền các giá trị bị khuyết

Có một thuộc tính quan trọng và khá phức tạp chúng tôi cần giải thích cụ thể.

- Tao cột CPU rank, và GPU performance:

- + Bộ dữ liệu không có đầy đủ thông tin CPU (Processor), nên chúng tôi dùng regex để trích xuất từ phần tên laptop ra. Sau khi có được, thì thông tin CPU vừa không chuẩn (thay vì “intel core i5-1135g7” thì là “intel i5 1135g7” hoặc “core processor 1135g7”) và có tận tới 503 unique value ở cột biến phân loại, nên lúc này chúng tôi tiến hành nối với dữ liệu CPU_rank đã thu thập trước đó, và ánh xạ theo thuật toán: Chẳng hạn như ta có “intel i7 5500u”, và chuẩn của nó là “intel core i7-5500u”, đầu tiên tìm kiếm riêng phần mã “5500u”, ta được 2 kết quả là “amd ryzen 5500u” và “intel core i7-5500u”. Tiếp theo sẽ tìm lần lượt từ “i7” và “core” trong 2 kết quả đó, kết quả nào khớp nhiều hơn thì chọn. Số lượng unique value sau đó còn 178.
 - + Tương tự để tạo cột GPU performance, nhưng thông tin GPU không có mã số như CPU, chủ yếu là tên, nên chúng tôi chuẩn hoá thông tin GPU với bộ dữ liệu GPU_performance bằng giải thuật tìm kiếm dựa trên bài toán quy hoạch động: xâu con chung dài nhất của hai chuỗi. Kết quả: “radeon vega 3”-> “amd radeon rx vega 3”. Số lượng unique value từ 191 còn 97.
 - + Hai cột CPU rank, và GPU performance sẽ là thang đo thay cho tên của CPU, và GPU khi dự đoán.
 - Xử lý tách ra thêm cột với những ô những dữ liệu chứa nhiều loại thông tin. Chẳng hạn:
 - + Với cột RAM có dạng “4 GB DDR4” tương ứng với dung lượng của ram, đơn vị, loại ram. Đầu tiên, tách “DDR4” để được “RAM Type”, sau đó thông nhất đơn vị ram chuyển về GB, tách phần số được “RAM Size”.
 - + Tách tên thương hiệu từ tên laptop
- Xử lý giá trị bị khuyết: Các phương pháp xử lý giá trị khiếm khuyết trong bộ dữ liệu được chúng tôi sử dụng:
- Xóa dòng: chúng tôi dùng trong trường hợp:
 - + Giá trị dự đoán không thể tự tiện điền dữ liệu: Price, Best Seller Rank.
 - + Giá trị khiếm khuyết không nhiều: Date First Available.
 - + Điền vào dữ liệu khiếm khuyết: có các phương thức:

- + Theo tỷ lệ các giá trị: dùng trong trường hợp dữ liệu là những con số hoặc giá trị văn bản cố định, không thể dùng giá trị tính toán mean, nội suy,...
Các cột được áp dụng: Number of USB 2.0 Ports, Number of USB 3.0 Ports, Optical Drive Type, Hard Drive Type, Memory Type, Laptop Type.
- Điền giá trị trung vị hoặc trung bình theo nhóm các biến chúng em cho là có thể dựa vào. Có thể áp dụng thêm phương thức forward fill / back fill để điền vào giá trị khiếm khuyết.
Các cột được áp dụng: Hard Drive Size, Memory Size, Memory Speed, Screen Size.
- Sử dụng phương thức gom cụm (KNN imputer). Các cột được áp dụng: Graphics Coprocessor, Processor. Sử dụng phương thức nội suy: có thể được tính theo nhóm biến khác và sử dụng thêm phương thức forward fill / back fill.
Các cột được áp dụng: Customer Reviews, Graphics Coprocessor perf, Processor Rank.
- Điền giá trị nhất định: chúng em hiểu rõ tại sao lại khuyết và tìm được giá trị khác để điền vào.
+ Các cột được áp dụng: Operating System, old_price.



Hình 3. Quy trình xử lý dữ liệu

2.3. Khảo sát bộ dữ liệu

Sau khi thực hiện bước tiền xử lý dữ liệu, file dữ liệu sạch (Tidy Data) có kích thước 378 KB, bao gồm 1159 dòng dữ liệu và 24 thuộc tính. Thông tin cụ thể của các thuộc tính được thể hiện ở bảng sau:

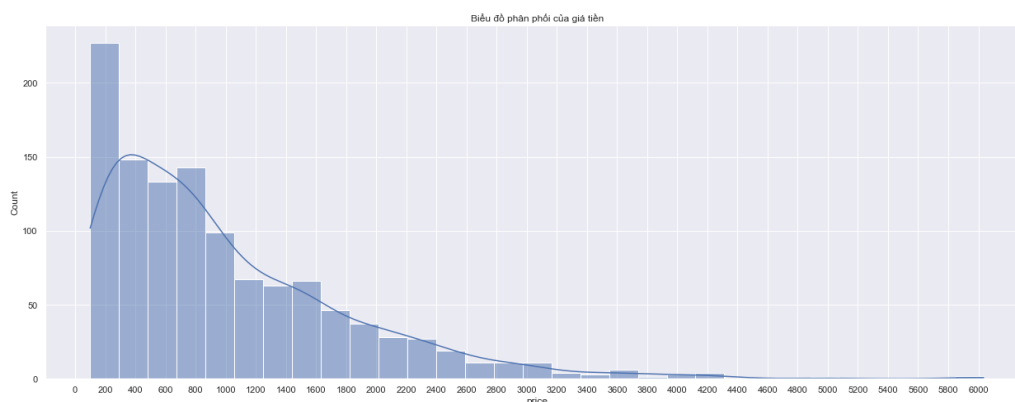
Bảng 1. Mô tả thông tin các thuộc tính

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Minh họa dữ liệu
1	name	Tên của laptop	object	acer aspire 5 a515...
2	Brand	Thương hiệu của laptop	object	acer, hp, samsung...
3	Best Sellers Rank	Mức xếp hạng việc bán chạy của laptop	float64	293, 250,270...
4	weight	Trọng lượng của laptop	float64	1.587, 1.905,..
5	Laptop type	Loại laptop	object	traditional, 2 in 1
6	Laptop purpose	Mục đích hướng đến của laptop	object	general, gaming
7	Screen Size	Kích thước của màn hình laptop	float64	15.6, 14 ...
8	Hard Drive Size	Kích thước của ổ cứng laptop	float64	128, 512,...
9	Hard Drive Type	Loại ổ cứng của laptop	object	Ssd, hdd,...
10	RAM speed	Tốc độ đọc của bộ nhớ RAM	float64	2.4, 2.6,...
11	RAM size	Kích thước của bộ nhớ RAM	float64	8, 16,...
12	RAM type	Các loại bộ nhớ RAM của laptop	object	DDR3, DDR4,...
13	CPU	Bộ xử lý trung tâm của laptop	object	intel corei5 1140g7,..
14	CPU rank	Mức xếp hạng của CPU	float64	2626, 3701,...
15	GPU	Bộ xử lý đồ họa của laptop	object	intel hd 400, nvidia geforce gtx 1650,...
16	GPU performance	Hiệu năng sử dụng của GPU	float64	2.3, 2.6
17	Optical Drive Type	Ổ đĩa bên ngoài của laptop	object	Yes/no
18	Operating System	Hệ điều hành của laptop	object	Windows, chrome os,...
19	Number of USB 3.0 Ports	Số cổng USB 3.0 của laptop	float64	0, 1, 2, ...

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Minh họa dữ liệu
20	Number of USB 2.0 Ports	Số cổng USB 2.0 của laptop	float64	0, 1, 2,...
21	Date First Available	Ngày hoạt động lần đầu của laptop	object	2017-09-21, 2021-08-15
22	Customer Reviews	Sự đánh giá của các người sử dụng	float64	4, 5, 4.5,...
23	old_price	Giá niêm yết của laptop	float64	186.99, 256,...
24	price	Giá hiện tại của laptop	float64	199, 207,...

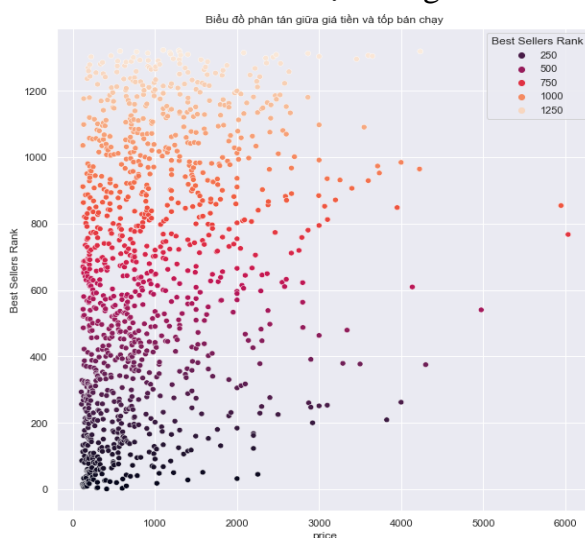
3. PHÂN TÍCH THẨM ĐÒ DỮ LIỆU

Đầu tiên, chúng tôi có cái nhìn tổng quan về biến target price. Chúng tôi thấy hiện nay laptop trên thị trường tập trung nhiều nhất ở phân khúc từ 200 đến dưới 1000 đô.



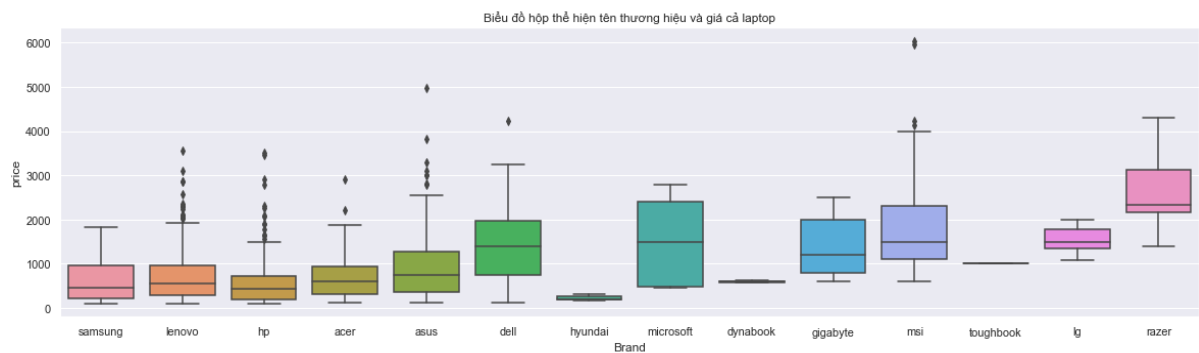
Hình 4. DisPlot thể hiện tương quan giữa số lượng laptop và 'Price'

Tuy nhiên, ta thấy có một số laptop có giá rất cao trên 5000 đô, thì đây hầu như là các máy chơi gaming, hoặc dành cho creator chuyên nghiệp. Bên cạnh các mẫu laptop giá rẻ chiếm đa số trên thị trường.



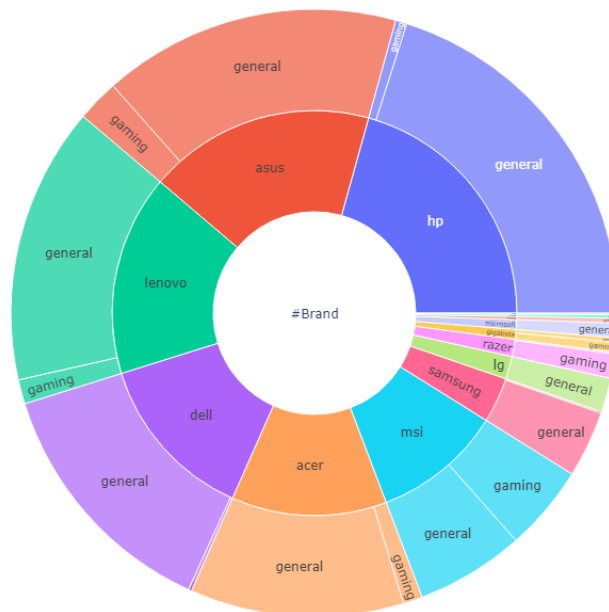
Hình 5. Scatter Plot thể hiện tương quan giữa biến 'Best Sellers Rank' và 'Price'

Dựa vào Hình 5 biểu đồ cho thấy: trong khoảng giá dưới 1000 đô, cũng tập trung nhiều sản phẩm nằm trong top mua sắm của người dùng nhất, đặc biệt là top đầu 200 hoàn toàn trội hơn so với các phân khúc cao hơn. Tuy nhiên ở đây cũng là miền có nhiều sản phẩm rẻ nhất và nằm chót về lượt mua sắm. Về sau, khi giá càng cao, các sản phẩm có xu hướng cân bằng, tức không nằm xếp hạng thấp nhất cũng không nằm xếp hạng top mua sắm.



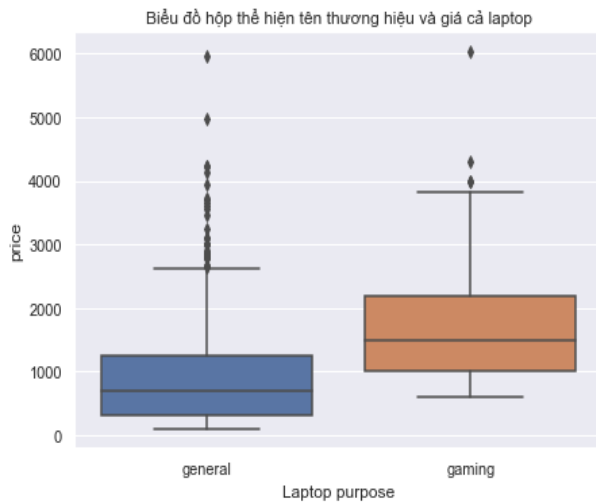
Hình 6. BoxPlot thể hiện tương quan giữa biến ‘Brand’ và ‘Price’

Ở Hình 6 thể hiện tên thương hiệu và giá cả laptop. Chúng tôi thấy rằng có một số box gần như overlap với nhau khi so sánh về giá, chẳng hạn Samsung, Lenovo, HP, Acer, Asus. Còn một số box thì lại ở phân khúc giá cao hơn hẳn các box còn lại như MSI, Razer.



Hình 7. Sunburst Plot thể hiện tương quan giữa biến ‘Brand’ và ‘Laptop purpose’

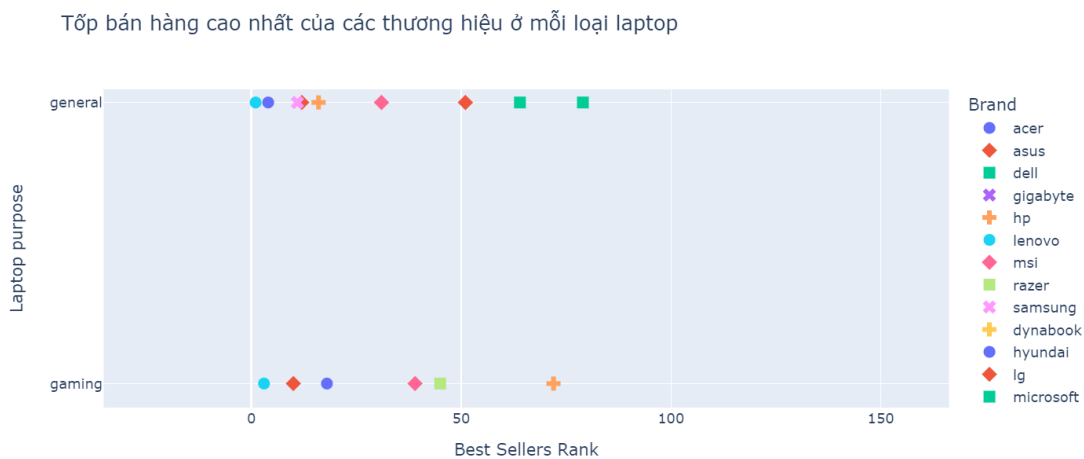
Để làm rõ hơn, chúng tôi phân tích về chênh lệch các kiểu laptop mà các hãng sản xuất, có hai loại là laptop *gaming* và laptop *đại trà*. Dễ thấy với Samsung, Lenovo, HP, Acer hay Asus tập trung sản xuất máy *đại trà* và trội hơn nhiều so với *gaming*. Ngược lại với Razer, hãng này lại thiên về sản xuất máy *gaming*.



Qua đó, chúng tôi thực hiện phân tích quan hệ giữa kiểu laptop và giá, thì giá máy *gaming* hầu như là cao hơn hẳn so với *đại trà*. Từ đó chúng tôi giải thích được vì sao giá một số thương hiệu lại cao hơn một số thương hiệu khác.

Tuy nhiên, điều này còn tùy vào định hướng mục tiêu sản xuất của mỗi thương hiệu: có những thương hiệu muốn cân bằng cả hai loại, nhưng cũng có những thương hiệu muốn tập trung vào một dòng máy, như nhắc tới *Razer* là nhắc tới máy *gaming*.

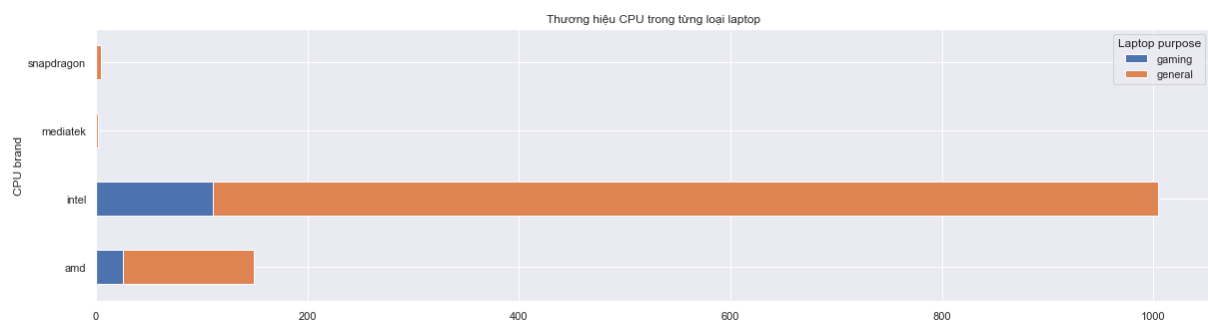
Hình 8. Box Plot thể hiện tương quan giữa biến ‘Laptop purpose’ và ‘Price’



Hình 9. Dot Plot - Tương quan giữa biến ‘Laptop purpose’ và ‘Best Sellers Rank’

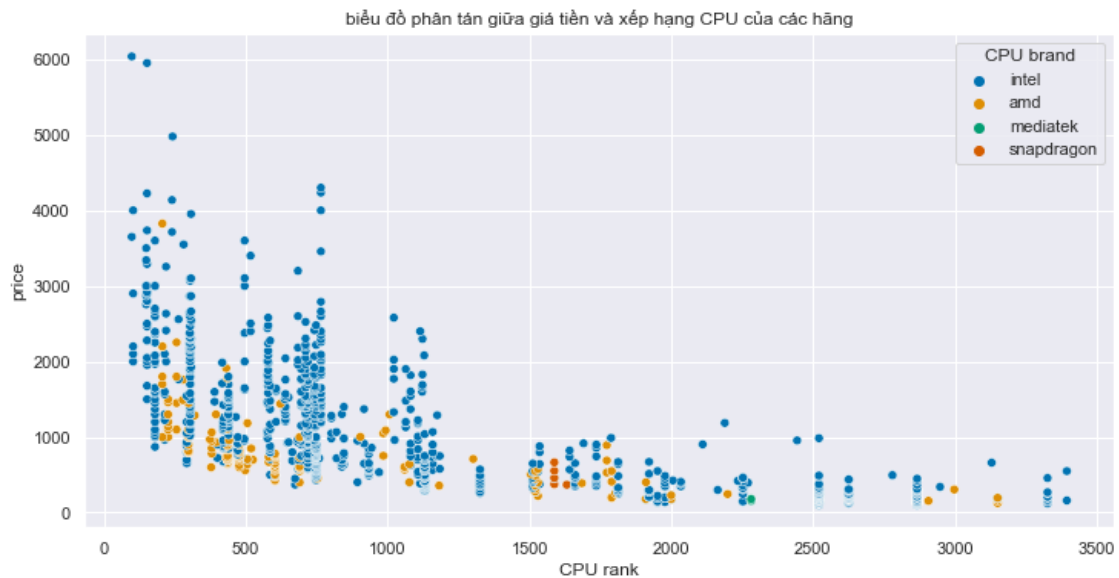
Thông qua Hình 9, chúng tôi nhận thấy với loại máy *đại trà*, thương hiệu đang được mua nhiều nhất là *Lenovo, Acer, Samsung, Asus, HP*. Còn với máy *gaming*, thương hiệu được mua nhiều nhất là *Lenovo, Asus, Acer, MSI, Razer*

Một trong những yếu tố quyết định mục đích sử dụng laptop như vừa rồi, là nằm ở CPU trong laptop. Vì vậy, với laptop mục đích *đại trà* và laptop *gaming*, các CPU cung ứng của nó tới từ những thương hiệu nào?



Hình 10. BarPlot - Tương quan giữa biến ‘Laptop purpose’ và ‘CPU Brand’

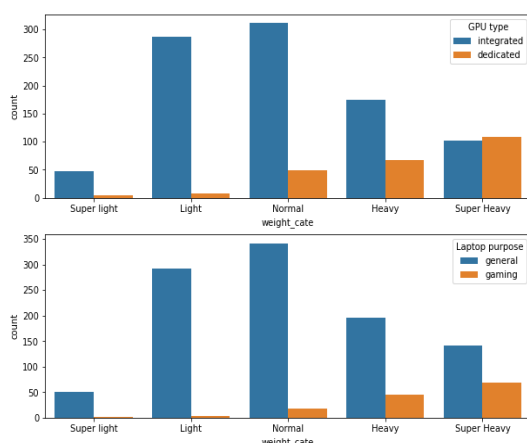
Chúng tôi thấy hiện nay thị trường *CPU gaming* không quá phổ biến với phần còn lại. Đồng thời, chúng ta biết rằng có hai thương hiệu phổ biến nhất là *Intel* và *AMD*.



Hình 11. Scatter Plot - Tương quan giữa biến 'CPU rank' và 'Price'

Dựa trên Hình 11 - biểu đồ phân tán giữa giá tiền và xếp hạng *CPU* ('CPU rank' càng thấp càng mạnh) của các hãng, chúng tôi thấy *CPU AMD* tuy cung cấp không đa dạng bằng *Intel*, nhưng hãng này vẫn đuổi kịp về hiệu năng so với *Intel*, đồng thời, những laptop chạy chip *AMD* cũng có giá mềm hơn rất nhiều và cho mức hiệu năng ngang ngửa với *intel*.

Có thể giải thích thị phần *AMD* không cao bằng *Intel*, là do hãng hay gặp một số lỗi liên quan đến xung đột phần mềm, cũng như việc *AMD* đã chìm trên thị trường gần một thập kỷ và hãng này vừa vực dậy từ năm 2016 nhờ kiến trúc Zen đột phá, cho tốc độ tốt, và giá thành mềm hơn. Có thể nói *AMD* đang từng bước thu phục thị trường bên cạnh ông lớn *Intel*.

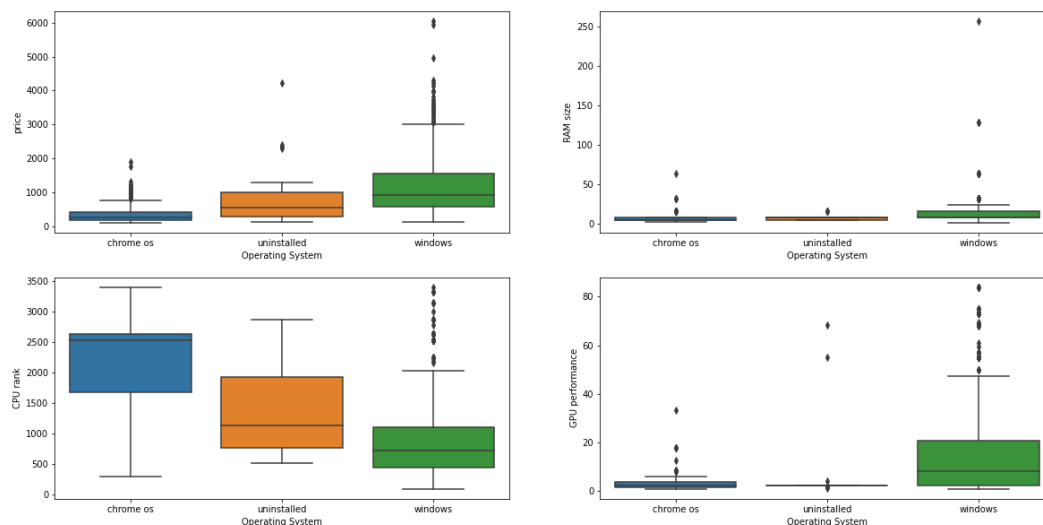


Hình 12. BarPlot - Tương quan giữa biến 'Weight' với 'GPU' và 'Laptop purpose'

Xét quan hệ giữa trọng lượng laptop và giá tiền. Khác với dự đoán là laptop càng nhẹ thì giá càng cao, thì ngược lại, máy càng nặng thì giá tiền lại càng cao.

Chúng tôi có thể thấy, khi trọng lượng càng tăng số lượng máy có card đồ họa rời càng tăng (máy có card rời cũng sẽ bao gồm card tích hợp) mà điều này cũng dẫn tới số lượng máy *gaming* cũng tập trung nhiều hơn, nên có thể giải thích được vì sao máy có trọng lượng lớn lại thường giá cao.

Sau khi phân tích về yếu tố thương hiệu, và phần cứng, thử kiểm tra xem yếu tố phần mềm là hệ điều hành có liên quan gì tới giá không.



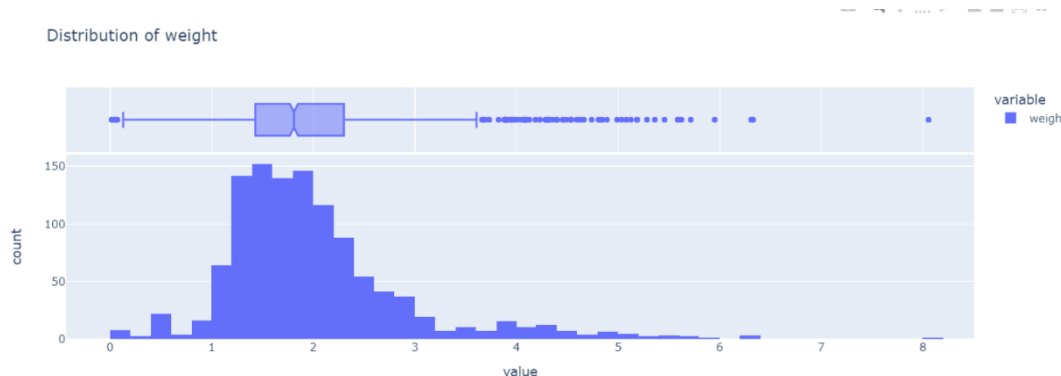
Hình 13. Box Plot - Tương quan giữa biến ‘Operating System’ với ‘price’, ‘RAM size’, ‘CPU rank’ và ‘GPU performance’

Chúng tôi thấy những laptop được hệ điều hành *chrome os* có giá rẻ hơn *windows*, và từ phân tích trên ta thấy hệ điều hành *chrome os* được cài ở những máy có cấu hình rất thấp nên kết luận khác với *windows*, *chrome os* là hệ điều hành nhẹ cài trên máy không quá mạnh, từ đó giá thường rẻ.

4. PHƯƠNG PHÁP MÁY HỌC

4.1. Loại thuộc tính phù hợp với mô hình

- Đối với biến định lượng, chúng tôi sử dụng độ đo Pearson correlation để lọc ra những biến có khả năng phù hợp với dữ liệu nhất. Sau đó chúng tôi tiến hành kiểm tra kỹ lưỡng các biến đó có thực sự phù hợp với mô hình hay không thông qua quy trình sau:
- + Trực quan hình ảnh phân phối dữ liệu (histogram) kiểm tra phân phối và tình trạng của outlier.



Hình 14. Histogram – Phân phối dữ liệu biến ‘Weight’

- + Trục quan regression plot kết hợp với độ đo Pearson và Residual Plot để kiểm tra về mức độ tương quan của biến đó với biến dự đoán.

Khi thông qua 3 bước trên, chúng tôi có thể phân nào xem xét rõ các biến định lượng nào thật sự ảnh hưởng đến biến dự đoán và có thể dùng được.

- Đối với biến định tính, chúng tôi sử dụng quy trình khác:
 - + Trục quan hình ảnh phân phối dữ liệu (histogram) kiểm tra phân phối và tình trạng của outlier.
 - + Trục quan Box plot kết hợp cùng kiểm tra Anova để kiểm tra mức độ chắc chắn về sự ảnh hưởng của biến định tính đó với biến dự đoán.

Khi thông qua 3 bước trên, chúng tôi có thể phân nào xem xét rõ các biến định tính nào thật sự ảnh hưởng đến biến dự đoán và có thể dùng được.

Sau khi hoàn thành quy trình trên, chúng tôi đã chọn ra được 9 thuộc tính bao gồm: ‘weight’, ‘CPU rank’, ‘GPU performance’, ‘Brand’, ‘Laptop type’, ‘Laptop purpose’, ‘RAM type’ và ‘Operating System’.

4.2. Mô hình sử dụng

Bộ dữ liệu được chia thành 2 tập Train và Test (tỷ lệ 8-2) và sử dụng các mô hình máy học sau:

- Decision Tree
- Random Forest
- Ridge
- Support Vector Machine - SVM

4.3. Kết quả mô hình

Chúng tôi sử dụng độ đo đánh giá là: **R²**

Kết quả mô hình thể hiện trong bảng sau

Bảng 1. Kết quả chạy mô hình máy học

Mô hình	Kết quả
Decision Tree	0.571920
Random Forest	0.758227
Ridge	0.578673
Support Vector Machine - SVM	0.581698

Từ các kết quả trên ta có thể rút ra một số nhận xét:

- Mô hình cho kết quả tốt nhất là Random Forest
- Kết quả ở những mô hình máy học khác chưa được tốt
- Sau khi tiền xử lý đã làm cho bộ dữ liệu đã bị giảm đi đáng kể dẫn đến kết quả khi áp dụng các mô hình không tốt.

5. KẾT LUẬN

5.1. Kết quả đạt được

- Qua đồ án này, chúng tôi đã đạt mục tiêu đã được đề ra từ đầu, đó là tự thu thập và xây dựng, phân tích và trục quan được bộ dữ liệu.
- Xây dựng được các mô hình áp dụng cho bộ dữ liệu
- Tìm hiểu, sử dụng được một số phương pháp tiền xử lý, phân tích và trục quan, cũng như xây dựng các mô hình máy học áp dụng cho bộ dữ liệu đã xây dựng.
- Nắm được kiến thức cơ bản, có cái nhìn cơ bản tổng thể về Phân tích và trục quan dữ liệu cũng như về Machine Learning.

5.2. Khó khăn gặp phải

- Thiếu kinh nghiệm giải quyết các vấn đề gặp phải khi tự thực hiện thu thập dữ liệu
- Thiếu kinh nghiệm trong việc tìm hiểu các thư viện mới, các cách xử lý mới trong phân tích và trục quan dữ liệu
- Thiếu kinh nghiệm trong bài toán xử lý dữ liệu có nhiều thuộc tính nói riêng và các bài toán Machine Learning nói chung dẫn đến việc gặp nhiều khó khăn trong quá trình thu thập cũng như xử lý dữ liệu.
- Bộ dữ liệu chưa đủ lớn, dẫn đến kết quả mô hình chưa cao như mong muốn. Nhiều mô hình chạy demo cho kết quả sai
- Hạn chế về kiến thức nền khiến cho việc tiếp cận các phương pháp xử lý dữ liệu cũng như tinh chỉnh mô hình trong python và xây dựng app demo bằng streamlit

TÀI LIỆU THAM KHẢO

- [1] Book: Hands-On Web Scraping with Python: Perform Advanced Scraping Operations Using Various Python Libraries and Tools Such as Selenium, Regex, and Others
- [2] On the Selection of Decision Trees in Random Forests (Simon Bernard, Laurent Heutte, Sebestien Adam) <https://ieeexplore.ieee.org/document/5178693>
- [3] Hands-on Machine Learning with Scikit-Learn, Keras Tensor-Flow, 2nd Edition(Aurélien Geron)
- [4] <https://plotly.com/python/basic-charts/>
- [5] <https://docs.scrapy.org/en/latest/>
- [6] <https://seaborn.pydata.org/>
- [7] <https://www.geeksforgeeks.org/>
- [8] <https://scrapeops.io/python-scrapy-playbook/>

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Trịnh Thế Hiển	Tiền xử lý dữ liệu Trực quan dữ liệu Train model Hỗ trợ viết báo cáo
2	Nguyễn Hùng Trung Hiếu	Thu thập dữ liệu Tiền xử lý dữ liệu Trực quan dữ liệu Hỗ trợ train model Hỗ trợ viết báo cáo
3	Lưu Quang Tiến Hoàng	Tiền xử lý dữ liệu Hỗ trợ trực quan dữ liệu Chuẩn bị slide Hỗ trợ viết báo cáo
4	Nguyễn Thành Nhân	Hỗ trợ trực quan dữ liệu Train model Làm Demo Viết báo cáo Hỗ trợ làm slide