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

DỰ ĐOÁN GIÁ LAPTOP VÀ DESKTOP CŨ TRÊN CHỢ TỐT

Sinh viên thực hiện:			
STT	Họ tên	MSSV	Ngành học
10	Trương Thị Thanh Thanh	20520767	KHMT
33	Huỳnh Nguyễn Vân Khánh	20521446	KHMT
36	Nguyễn Thị Ngọc Nga	20521641	KHMT

TP. HỒ CHÍ MINH – 12/2022

1. GIỚI THIỆU

Trong đồ án môn học này, nhóm chúng tôi làm về dự đoán giá laptop và desktop cũ trên **Chợ Tốt**. Với đề tài, chúng tôi đã sử dụng một loạt các công cụ và giải pháp. Đầu tiên, chúng tôi đã sử dụng Google Colab để hỗ trợ trong quá trình lập trình và Google Drive để lưu trữ mã nguồn cũng như dữ liệu. Quá trình thu thập dữ liệu được thực hiện thông qua việc sử dụng các công cụ như Beautiful Soup và Selenium để crawl dữ liệu từ trang web Chợ Tốt. Sau đó, chúng tôi đã thực hiện tiền xử lý dữ liệu bằng cách loại bỏ các hàng có nhiều giá trị trống, thay thế các giá trị trống và giá trị nhiễu bằng giá trị mode của từng loại. Điều này giúp cải thiện chất lượng dữ liệu và chuẩn bị nền tảng cho việc xây dựng mô hình dự đoán giá.

Kết quả đo lường hiệu suất của mô hình đã được đánh giá bằng cách sử dụng các thang đo như R2 score, Mean Squared Error (MSE), Mean Absolute Error (MAE). Mô hình được nhóm sử dụng để tiến hành training là mô hình Histogram Gradient Boosting, kết quả đạt được R2 score là 0.7188, MAE là 2.253531e+6, MSE là 1.1750e+13, đồng nghĩa với việc mô hình có khả năng dự đoán khá tốt so với dữ liệu thực tế.

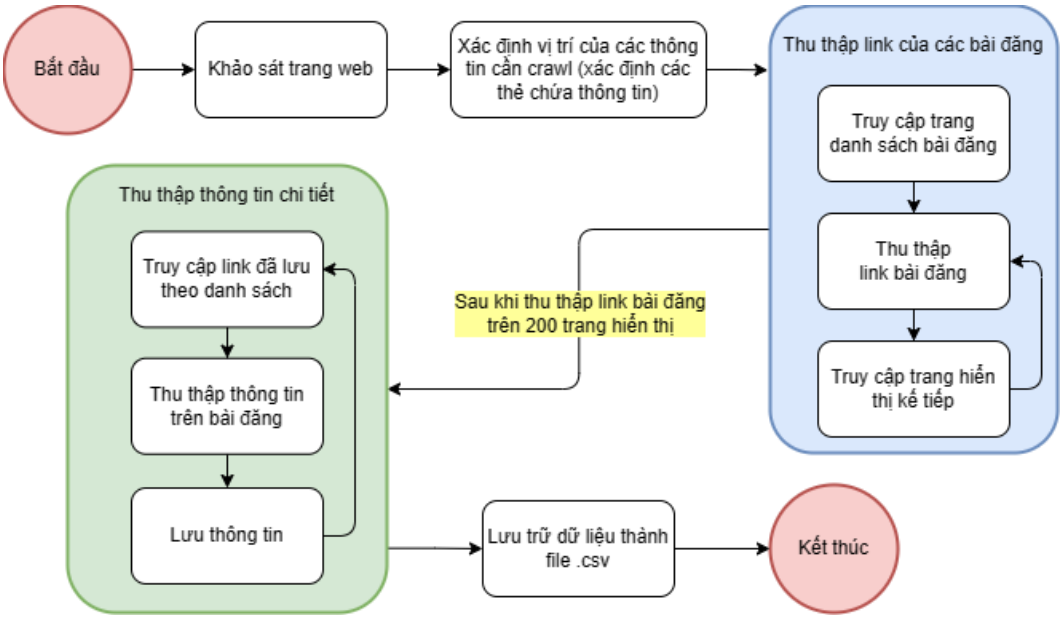
Chúng tôi cam kết minh bạch về đề tài và bộ dữ liệu. Bộ dữ liệu được thu thập tự động từ trang web Chợ Tốt và không phụ thuộc vào bất kỳ nguồn ngoại vi nào. Đồng thời, đề tài và bộ dữ liệu được xây dựng độc lập, không dựa trên bất kỳ đề tài nào khác.

2. MÔ TẢ BỘ DỮ LIỆU

Bộ dữ liệu được tự thu thập từ trang web chợ tốt trong khoảng thời gian từ ngày 22 đến 23 tháng 10 năm 2023, sử dụng các thư viện như Selenium, BeautifulSoup và Requests.

Dữ liệu được lấy từ hai danh mục chính trên trang chợ tốt, đó là Laptop và Máy tính để bàn. Thông tin từ mỗi dòng dữ liệu được thu thập từ các bài đăng riêng lẻ trong từng danh mục. Đối với mỗi danh mục, nhóm đã thu thập dữ liệu từ 200 trang đầu tiên, mỗi trang có khoảng 20-25 bài đăng.

Toàn bộ quy trình thu thập có thể tóm tắt như sau:



Hình 1. Quy trình thu thập dữ liệu

Dữ liệu thu thập có tổng cộng 8004 dòng và 13 cột, trong đó bao gồm biến mục tiêu. Trong 13 cột này, có 12 cột được lấy trực tiếp từ trang chợ tốt, còn một cột (classify - phân loại) được nhóm thêm vào trong quá trình thu thập để phân biệt giữa máy tính để bàn (desktop) và laptop.

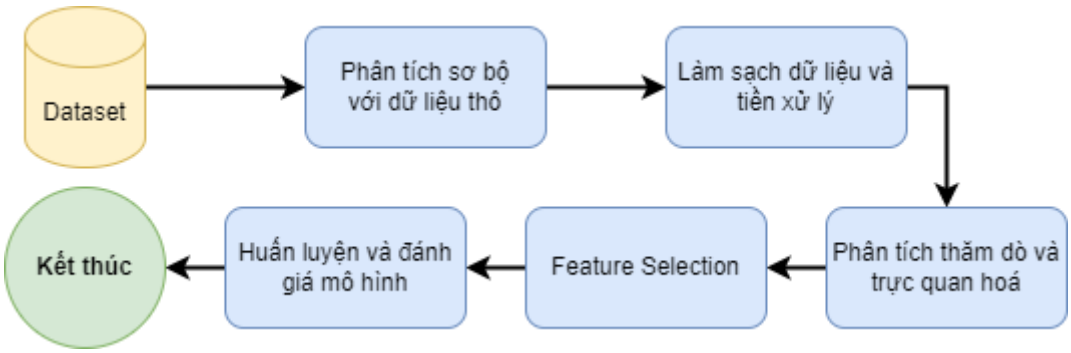
Mặc dù dữ liệu thu thập từ máy tính để bàn và laptop có đôi chút tương đồng, tuy nhiên, đối với máy tính để bàn, trang chợ tốt không cung cấp thông tin về tên hãng và dòng máy. Do đó, các giá trị trong cột pc_brand và pc_model của dữ liệu máy tính để bàn sẽ bị trống.

Bên dưới là bảng mô tả các feature trong bộ dữ liệu.

Tên biến	Loại biến	Số dữ liệu khuyết	Mô tả
pc_brand	Phân loại	4010	Nhãn hiệu của máy tính. Vd: DELL, Apple, Lenovo, …
pc_model	Phân loại	4010	Dòng máy. Vd: ThinkPad, …
elt_condition	Phân loại	115	Tình trạng sử dụng máy, là máy mới hay đã qua sử dụng, máy đã qua sửa chữa hay chưa.
elt_warranty	Phân loại	115	Thời gian bảo hành. Vd: 1 tháng, 2 tháng, hết bảo hành, …
desktop_ screen_size	Phân loại	2115	Thông tin về kích cỡ màn hình của laptop dựa trên màn hình tích hợp, trong khi đối với máy tính để bàn, việc có màn hình đi kèm hay không đi kèm phụ thuộc vào từng trường hợp cụ thể.
pc_cpu	Phân loại	1243	Dòng cpu của máy. Vd: Intel Core i5, Intel Core i3, …
pc_ram	Phân loại	1277	Dung lượng của RAM. Vd:128 GB, …
pc_vga	Phân loại	2365	Card đồ họa của máy. Vd: NVIDIA, Onboard, …
pc_drive_cap acity	Phân loại	1449	Dung lượng của ổ cứng.
elt_origin	Phân loại	115	Xuất xứ của máy. Vd: Việt Nam, …
usage_inform ation	Phân loại	115	Thông tin sử dụng (khác với tình trạng sử dụng)
classify	Phân loại	0	Phân loại là laptop hay máy bàn
price (biến mục tiêu)	Số	115	Giá bán của máy (tính trên đơn vị VND)

Bảng 1. Bảng mô tả dữ liệu

3. PHƯƠNG PHÁP PHÂN TÍCH



Hình 2. Quy trình phân tích dữ liệu

3.1. Phân tích sơ bộ với dữ liệu thô

Với bộ dữ liệu đã được chúng tôi thu thập trên trang chợ tốt có tổng cộng 8004 dòng và 13 cột dữ liệu. Quá trình phân tích thống kê đã được thực hiện để khám phá và hiểu các đặc điểm quan trọng của dữ liệu thu thập.

Trong quá trình phân tích, chúng tôi đã được xác định một số vấn đề chính trong dữ liệu. Đầu tiên, sự xuất hiện của dữ liệu thiếu đã được phát hiện. Điều này có thể làm giảm khả năng sử dụng của dữ liệu và đòi hỏi các biện pháp để xử lý các giá trị thiếu này. Thứ hai, sự trùng lặp trong dữ liệu đã được phát hiện, có nghĩa là có các bản ghi trùng nhau trong tập dữ liệu. Điều này có thể gây ra sai sót và ảnh hưởng đến kết quả phân tích. Cuối cùng, một số giá trị nhiễu đã được tìm thấy trong dữ liệu, có thể là kết quả của lỗi nhập liệu khi người dùng nhập liệu trên web Chợ Tốt.

Việc phát hiện và nhận biết những vấn đề này là quan trọng để đảm bảo rằng chúng tôi có một bộ dữ liệu nhất quán và chất lượng để sử dụng cho các bước phân tích tiếp theo.

3.2. Làm sạch dữ liệu và tiền xử lý

Sau khi tiến hành một quá trình thăm dò sơ bộ và thu được kết quả thống kê chi tiết về quy mô của bộ dữ liệu thô, nhóm chúng tôi tiến hành làm sạch và tiền xử lý dữ liệu để đảm bảo chất lượng và tính nhất quán của thông tin thu thập được.

Quá trình làm sạch dữ liệu bắt đầu bằng việc loại bỏ các dữ liệu trùng lặp, giúp đảm bảo tính chính xác và đồng nhất trong bộ dữ liệu. Tiếp theo đó, chúng tôi tiếp tục loại bỏ các hàng có giá trị "price" bị thiếu để giữ lại những dữ liệu có đầy đủ thông tin về giá trị quan trọng này. Đối mặt với tình trạng dữ liệu bị khuyết ở các trường như "pc_cpu", "pc_ram", và "pc_drive_capacity", chúng tôi quyết định loại bỏ những hàng dữ liệu mà cả ba giá trị trên đều bị thiếu.

Chúng tôi tiếp tục giải quyết giá trị nhiễu trong các trường như "pc_vga", "elt_warranty", và "elt_origin". Đồng thời, chúng tôi chuẩn hóa thông tin về "pc_cpu" để giải quyết giá trị nhiễu "Khác" và cung cấp thông tin chi tiết về CPU cho các mô hình cụ thể như "MacBook Air M1", "MacBook Pro M2", "Surface Pro X", ...

Quá trình tiền xử lý dữ liệu bao gồm việc chuyển đổi đơn vị và kiểu dữ liệu, ánh xạ giữa các giá trị ban đầu và giá trị chuẩn hóa, cũng như label encoding cho các trường dữ liệu còn lại. Điều này giúp tối ưu hóa quá trình huấn luyện mô hình, tạo ra một biểu diễn số hóa của dữ liệu dễ hiểu và hiệu quả cho mô hình máy học.

3.3. Phân tích thăm dò và trực quan hoá dữ liệu

Sau giai đoạn làm sạch và tiền xử lý dữ liệu, nhóm của chúng tôi đã chuyển sang giai đoạn trực quan hóa dữ liệu và phân tích mối quan hệ giữa các biến. Điều này nhằm mục đích tìm ra các cấu trúc và xu hướng quan trọng trong dữ liệu, đồng thời xác định độ tương quan giữa các biến dạng số và giá tiền (target variable). Bằng cách này, chúng tôi hy vọng có thể chọn ra những thuộc tính quan trọng và có ảnh hưởng đáng kể đối với quá trình huấn luyện mô hình dự đoán giá máy tính cũ.

Quá trình thăm dò dữ liệu bao gồm việc tạo các biểu đồ để trực quan hóa sự phân phối của các biến và hiểu rõ hơn về mối quan hệ giữa chúng. Đồng thời, chúng tôi tập trung vào việc tìm hiểu về mối quan hệ giữa các biến phân loại và biến mục tiêu để xác định những thuộc tính quan trọng có thể đóng góp lớn cho việc dự đoán giá máy tính cũ.

Những kết quả thu được từ quá trình này sẽ là cơ sở để chọn lọc các đặc trưng quan trọng để có thể xây dựng một mô hình dự đoán hiệu quả.

3.4. Feature Selection

Dựa trên quá trình phân tích chuyên sâu của chúng tôi với dữ liệu đã được xử lý, chúng tôi đã quyết định chọn các đặc trưng phù hợp để xây dựng mô hình dự đoán giá của laptop và desktop cũ. Quá trình lựa chọn đặc trưng đã được thực hiện cẩn thận, tập trung vào những biến số có ảnh hưởng trực tiếp đến giá. Các đặc trưng được chọn đã được xem xét kỹ về mặt thống kê và ý nghĩa trong ngữ cảnh của thị trường máy tính cũ trên Chợ tốt. Quá trình này giúp chúng tôi tạo ra một mô hình hiệu quả và giảm thiểu vấn đề phức tạp và tiêu tốn tài nguyên không cần thiết. Đồng thời, việc chọn lựa thông minh về đặc trưng giúp mô hình linh hoạt trong việc áp dụng cho các tình huống thực tế và dự đoán giá một cách chính xác và đáng tin cậy.

3.5. Huấn luyện và đánh giá mô hình

Nhóm của chúng tôi đã tiến hành một loạt các thử nghiệm trên 8 thuật toán máy học khác nhau, bao gồm Histogram Gradient Boosting, XGBoost, Random Forest, KNN, Decision Tree, Linear Regression, SVM, và Neural Networks, sử dụng 11 mô hình khác nhau. Đối với mỗi thuật toán và mô hình, chúng tôi đã đánh giá hiệu suất dựa trên ba thang đo quan trọng là R2 score, Mean Absolute Error và Mean Squared Error.

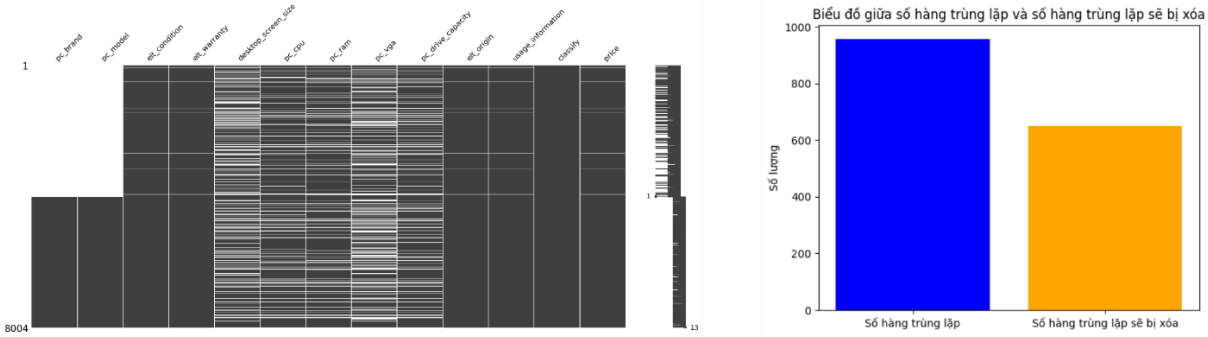
Trong quá trình thử nghiệm, chúng tôi đã giữ nguyên cài đặt mặc định của các thuật toán từ các thư viện, không thực hiện bất kỳ tinh chỉnh nào. Điều này nhằm mục đích đánh giá hiệu suất ban đầu của các mô hình mà không ảnh hưởng đến sự chủ quan từ việc điều chỉnh tham số.

Sau khi thu được kết quả đánh giá, chúng tôi đã lựa chọn mô hình tốt nhất để tiếp tục. Tiếp theo, chúng tôi thực hiện quá trình fine-tune trên mô hình đã chọn để tìm ra các tham số phù hợp nhất, nhằm đạt được kết quả dự đoán giá máy tính cũ tốt nhất có thể. Điều này giúp tối ưu hóa hiệu suất của mô hình và đảm bảo sự chính xác và độ tin cậy trong dự đoán giá sản phẩm.

4. PHÂN TÍCH SƠ BỘ

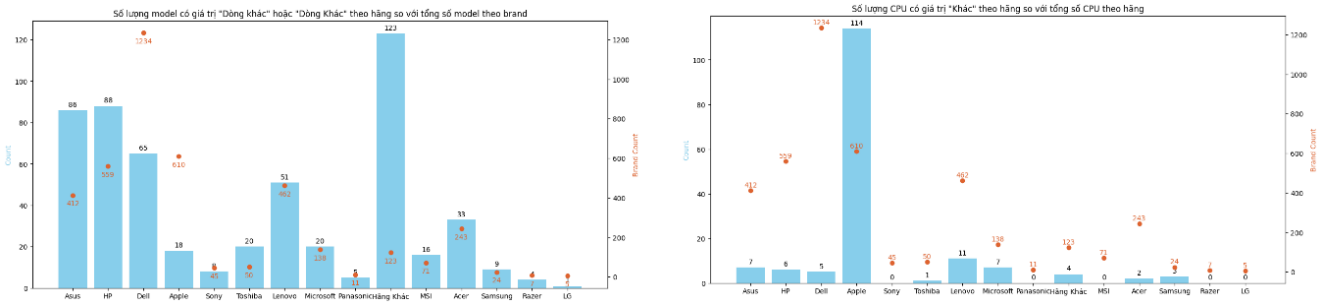
Sau quá trình thăm dò và phân tích, nhóm của chúng tôi đã ghi nhận một số phát hiện quan trọng như sau:

- **Dữ liệu thiếu nhiều:** Một vấn đề đáng chú ý là sự thiếu sót thông tin trong bộ dữ liệu. Có những dòng dữ liệu mà tất cả các cột đều bị thiếu, đặc biệt là những trường hợp mà link không thể truy cập được (link đã bị xóa).
- **Thiếu thông tin quan trọng ở các vị trí có ý nghĩa đặc biệt:** Phát hiện rằng có một số vị trí bị thiếu thông tin quan trọng, như pc_brand và pc_model, đặc biệt là trong các trường hợp thuộc danh mục 'desktop'. Điều này đặt ra thách thức trong việc xử lý và đánh giá dữ liệu liên quan đến máy tính để bàn.
- **Dữ liệu trùng lặp:** Bộ dữ liệu chứa một số dòng trùng lặp, nơi cần phải thực hiện quy trình loại bỏ để đảm bảo tính độc lập và độ tin cậy của dữ liệu. Cụ thể, có 957 dòng trùng lặp.



Hình 3. Biểu đồ biểu hiện mật độ thiếu sót của dữ liệu. Biểu đồ thể hiện sự tương quan giữa số hàng trùng lặp và số hàng trùng lặp sẽ bị xóa

- **Giá trị nhiễu trong pc_model và pc_cpu:** Giá trị "Dòng khác" hoặc "Dòng Khác" xuất hiện thường xuyên trong pc_model, gây nhiễu và khó khăn trong quá trình phân tích. Tương tự, giá trị "Khác" trong pc_cpu cũng gây ảnh hưởng lớn đối với các mô hình đặc biệt với thuộc pc_brand "Apple".



Hình 4. Biểu đồ thể hiện giá trị nhiễu trong pc_model và pc_cpu với tổng theo brand

Những phát hiện này đã định hình quá trình tiền xử lý dữ liệu của chúng tôi và là cơ sở quan trọng cho việc tối ưu hóa và hiệu chỉnh các mô hình dự đoán giá của chúng tôi.

5. LÀM SẠCH VÀ TIỀN XỬ LÝ DỮ LIỆU

Sau khi tiến hành một quá trình thăm dò sơ bộ và thu được kết quả thống kê chi tiết về quy mô của bộ dữ liệu thô, nhóm chúng tôi tiến hành giai đoạn làm sạch và tiền xử lý dữ liệu để đảm bảo chất lượng và tính nhất quán của thông tin thu thập được.

Trong quá trình làm sạch và tiền xử lý dữ liệu được tiến hành như sau:

Đối với quá trình làm sạch dữ liệu:

- Chúng tôi thực hiện bước xóa bỏ các dữ liệu trùng lặp để đảm bảo tính chính xác và đồng nhất trong bộ dữ liệu. Sau quá trình này, bộ dữ liệu của chúng tôi giảm xuống còn 7356 dòng.
- Tiếp theo, chúng tôi thực hiện việc loại bỏ những hàng có giá trị "*price*" bị thiếu, nhằm đảm bảo rằng chúng tôi chỉ giữ lại những dữ liệu có đầy đủ thông tin về giá trị quan trọng này. Kết quả là bộ dữ liệu hiện tại giảm xuống còn 7355 dòng.
- Đối mặt với tình trạng dữ liệu bị khuyết khá nhiều ở các giá trị quan trọng như "*pc_cpu*", "*pc_ram*", và "*pc_drive_capacity*", chúng tôi tiếp tục quá trình làm sạch bằng cách loại bỏ những hàng dữ liệu bị khuyết giá trị "*price*". Đặc biệt, chúng tôi quyết định loại bỏ những dòng dữ liệu mà cả ba giá trị "*pc_cpu*", "*pc_ram*", và "*pc_drive_capacity*" đều bị thiếu. Kết quả, bộ dữ liệu cuối cùng của chúng tôi hiện chỉ còn lại 6636 dòng, với mong muốn giữ lại những dữ liệu có đủ thông tin quan trọng cho quá trình phân tích và mô hình hóa tiếp theo.
- Chúng tôi nhận thức rằng giá trị nhiễu chiếm một tỷ lệ đáng kể trong bộ dữ liệu khi tiến hành thăm dò sơ bộ với dữ liệu thô. Để giải quyết vấn đề này, nhóm chúng tôi đã quyết định thực hiện quá trình thay thế giá trị nhiễu cũng như các giá trị bị khuyết còn lại trong bộ dữ liệu. Quá trình thay thế giá trị này không chỉ giúp nhằm duy trì tính toàn vẹn và chất lượng của bộ dữ liệu mà còn đảm bảo tính chính xác và đáng tin cậy của thông tin trong quá trình training mô hình dự đoán.
    + Đối với trường "*pc_vga*", chúng tôi đã thực hiện quá trình thay thế các giá trị trống bằng giá trị "`Onboard`". Đồng thời, để giải quyết vấn đề về giá trị nhiễu "`Khác`", chúng tôi cũng tiến hành ghi đè giá trị "`Onboard`" lên tất cả các ô chứa giá trị "`Khác`". Việc này nhằm mục đích đơn giản hóa và chuẩn hóa thông tin về card đồ họa, đồng thời giúp bảo đảm tính chính xác và nhất quán của dữ liệu trong quá trình phân tích và mô hình hóa.
    + Đối với trường "*elt_warranty*" với việc không có giá trị bị khuyết, chúng tôi quyết định thực hiện quá trình gán nhãn lại giá trị: "`Bảo hành hãng`" sẽ được chuyển thành " `Hết bảo hành`".
    + Đối với trường "*elt_origin*", chúng tôi thực hiện quá trình thay thế giá trị nhiễu "`Đang cập nhập`" bằng giá trị "`Nước khác`". Bước này giúp làm rõ và chuẩn hóa thông tin về xuất xứ.
    + Đối với các trường dữ liệu "*desktop_screen_size*", "*pc_ram*" và "*pc_drive_capacity*", chúng tôi nhận thấy sự khác biệt giữa trường dữ liệu "*classify*" là "`desktop`" và "`laptop`". Trong các cột này, không có sự xuất hiện của giá trị nhiễu. Do đó, chúng tôi quyết định điền giá trị khuyết dựa trên phân loại của từng mẫu dữ liệu.

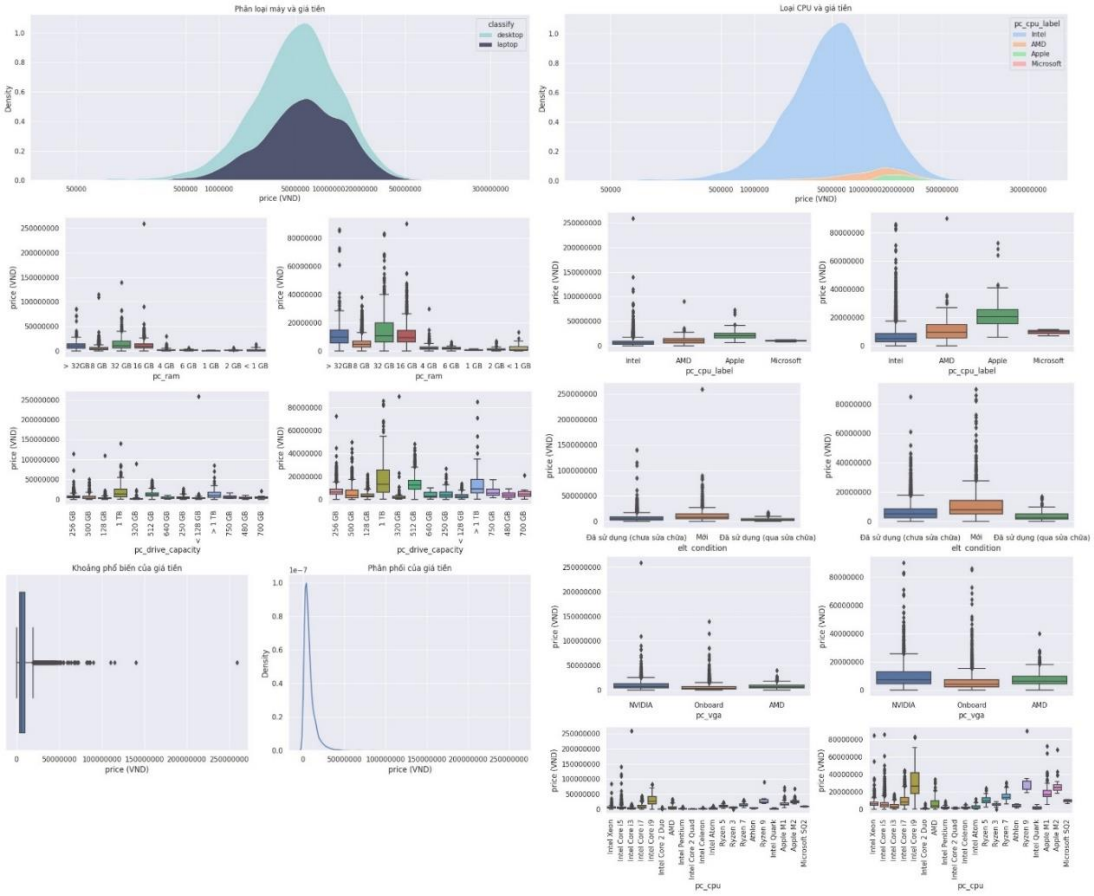
+ Cụ thể với trường "*desktop_screen_size*", chúng tôi thực hiện việc lấy giá trị xuất hiện nhiều nhất (mode) để thay thế các giá trị bị khuyết đối với "*classify*" là "`laptop`". Đối với giữa trường dữ liệu "*classify*" là "`desktop`", chúng tôi quyết định thay thế giá trị bị khuyết bằng "`Không bán kèm màn hình`". Giúp bảo đảm tính nhất quán và chất lượng của dữ liệu tùy thuộc vào phân loại cụ thể của từng mẫu.
+ Với "*pc_ram*" và "*pc_drive_capacity*" tương tự với "*desktop_screen_size*" nhóm tiến hành lấy giá trị xuất hiện xuất hiện nhiều nhất (mode) để thay thế các giá trị bị khuyết đối với cả 2 phân lớp.
+ Trong trường hợp của các trường "*pc_brand*" và "*pc_model*", chúng tôi nhận thấy rằng trong quá trình thu thập dữ liệu, trường dữ liệu "*classify*" là "`desktop`" không cung cấp giá trị cho các trường này, làm cho toàn bộ giá trị của "*pc_brand*" và "*pc_model*" ở phân loại này đều trở thành giá trị bị khuyết. Vì vậy, quyết định của chúng tôi là thay thế tất cả các giá trị bị khuyết ở "*pc_brand*" và "*pc_model*" bằng giá trị "`desktop`".
+ Ở "*pc_brand*" giá tri bị nhiễu là "`Hãng khác`" nhưng vì không có phương pháp thay thế hợp lý nên nhóm sẽ không thay đổi giá trị này.
+ Đối với trường "*pc_model*" với trường dữ liệu "*classify*" là "`laptop`", chúng tôi đã nhận thấy sự xuất hiện của giá trị nhiễu như "`Dòng khác`" và "`Dòng Khác`". Do đó, quyết định của chúng tôi là thay thế tất cả các giá trị "`Dòng Khác`" bằng "`Dòng khác`" để giữ tính nhất quán trong dữ liệu. Với mỗi "*pc_brand*", do có sự đa dạng giữa các "*pc_model*", chúng tôi tiến hành thay thế các giá trị nhiễu và giá trị bị khuyết bằng giá trị "*pc_model*" xuất hiện nhiều nhất (mode) trong từng nhóm "*pc_brand*". Trong trường hợp "*pc_model*" xuất hiện nhiều nhất là "`Dòng khác`", chúng tôi sẽ thay thế bằng giá trị xuất hiện nhiều thứ hai. Đối với "*pc_brand*" là "`Hãng khác`", giá trị "*pc_model*" sẽ được giữ nguyên là "`Dòng khác`".
+ Trong quá trình thăm dò sơ bộ với dữ liệu thô, chúng tôi nhận thấy giá trị nhiễu "`Khác`" trong trường "*pc_cpu*" có ảnh hưởng lớn đối với các mô hình, đặc biệt là khi thuộc "*pc_brand*" là "`Apple`". Để làm sạch dữ liệu trong trường "*pc_cpu*", chúng tôi đã thực hiện việc thay thế các giá trị nhiễu và giá trị bị khuyết bằng giá trị "*pc_cpu*" xuất hiện nhiều nhất (mode) đối với mỗi "*pc_model*". Trong trường hợp giá trị "*pc_cpu*" xuất hiện nhiều nhất là "`Khác`", chúng tôi thay thế nó bằng giá trị xuất hiện nhiều thứ hai. Tuy nhiên, để đảm bảo tính chính xác đối với các "*pc_model*" cụ thể như "`MacBook Air M1`", "`MacBook Pro M1`", "`MacBook Pro M1 Touch Bar`", "`MacBook Air M2`", "MacBook Pro M2", "`MacBook Pro M2 Touch Bar`", "`Surface Pro X`" và "`Notebook 7`", chúng tôi đã gán giá trị "*pc_cpu*" một cách chi tiết. Cụ thể, với "`MacBook Air M1`", "`MacBook Pro M1`", và "`MacBook Pro M1 Touch Bar`", giá trị "*pc_cpu*" được gán là "`Apple M1`". Đối với "`MacBook Air M2`", "`MacBook Pro M2`", và "`MacBook Pro M2 Touch Bar`", giá trị "*pc_cpu*" được gán là "`Apple M2`". Đối với "`Surface Pro X`", giá trị "*pc_cpu*" được gán là "`Microsoft SQ2`", và đối với "`Notebook 7`", giá trị "*pc_cpu*" được gán là "`Intel Core i5`". Những giá trị này được lựa chọn để đảm bảo độ chính xác và thống nhất trong việc mô phỏng thông tin về CPU của từng "*pc_model*".

Đối với quá trình tiền xử lý dữ liệu:

Sau quá trình làm sạch dữ liệu, để tối ưu hóa quá trình huấn luyện mô hình dự đoán, nhóm của chúng tôi đã tiến hành bước tiền xử lý dữ liệu. Điều này nhằm tạo điều kiện thuận lợi và chuẩn bị dữ liệu đầu vào sao cho các thuật toán máy học có thể hiệu quả hơn trong việc học và dự đoán giá máy tính cũ. Quá trình tiền xử lý này đóng vai trò quan trọng trong việc nâng cao chất lượng và hiệu suất của mô hình dự đoán cuối cùng.

- Xóa cột "*usage_information*" vì cột này chỉ có một giá trị duy nhất và không ảnh hưởng đến "*price*".
- Thêm cột "*pc_cpu_label*" để có thể phân tích sự ảnh hưởng của loại cpu đối với "*price*".
- Với trường "*price*". Chúng tôi thực hiện các thao tác như loại bỏ các ký tự không cần thiết, loại bỏ chuỗi "đ" và dấu chấm ngăn cách phần nghìn trong giá trị. Sau đó, chúng tôi chuyển đổi kiểu dữ liệu của trường "*price*" thành "*float64*". Cuối cùng, chúng tôi đổi tên cột từ "*price*" sang "*price (VND)*" để thể hiện đơn vị tiền tệ.
- Chúng tôi đã tiến hành label encoding cho các trường dữ liệu còn lại, nhằm chuyển đổi các giá trị chuỗi thành các giá trị số nguyên. Việc này giúp chúng tôi tạo ra một biểu diễn số hóa của dữ liệu, làm cho thông tin có thể được hiểu bởi mô hình máy học một cách hiệu quả hơn, từ đó tăng khả năng hiểu của mô hình đối với dữ liệu, giúp tối ưu hóa quá trình huấn luyện, tạo ra một mô hình dự đoán giá máy tính cũ chính xác và hiệu quả.

6. PHÂN TÍCH THĂM DÒ VÀ TRỰC QUAN HOÁ DỮ LIỆU



Hình 5. Kết quả thăm dò dữ liệu

Dựa trên biểu đồ "Khoảng phổ biến của giá tiền" và "Phân phối của giá tiền" chúng ta thấy rõ phân phối của giá tiền bị lệch nhiều về phía bên trái. Phần lớn tập trung ở khoảng từ 5 triệu VND đến 20 triệu VND. Đặc biệt, xét trên giá tiền, có thể thấy dữ liệu có khá nhiều giá trị ngoại lệ (outliers). Hầu hết giá trị ngoại lệ nằm trong khoảng từ 30 triệu VND đến khoảng 65 triệu VND. Một số ngoại lệ có giá tiền thậm chí còn lớn hơn 100 triệu VND (đây là các trường hợp có khả năng cao là các giá trị nhiễu).

Biểu đồ "Phân loại máy và giá tiền" cho thấy không có sự khác biệt rõ rệt về hình dạng của phân phối về giá của máy bàn và máy laptop.

Một điều thú vị khi quan sát biểu đồ "Phân loại CPU và giá tiền", đó là phân phối theo giá của các máy có CPU thuộc dòng Intel có hình dạng gần giống như phân phối theo giá của loại máy. Ở biểu đồ này cũng có sự khác biệt đáng kể giữa phân phối của các loại CPU, theo đó CPU loại AMD và Apple có phân phối dẹp hơn, chứng tỏ các máy với CPU loại này có khoảng giá rộng hơn so với máy có CPU Intel.

Bên cạnh đó, nhóm còn tiến hành khảo sát khoảng giá của máy tính dựa trên bộ nhớ RAM, ổ cứng, loại CPU (Intel, AMD,…), dòng CPU (Intel Core i3, Intel Core i5,…), card màn hình (NVIDIA, Onboard,…), tình trạng sử dụng (máy cũ, máy mới, máy cũ đã qua sửa chữa),… và các biến phân loại đó (do một vài outliers có giá trị khá lớn, dẫn đến biểu đồ trực quan bị lệch, nên chúng tôi đã tạo thêm một biểu đồ khác sau khi đã loại bỏ các outlier có giá lớn hơn 100 triệu VND, việc này nhằm giúp cho việc phân tích dữ liệu được dễ dàng hơn). Trong đó, có một số phát hiện tiêu biểu như sau:

- Độ lớn của RAM: độ lớn của RAM có ảnh hưởng mạnh đến giá của máy, cụ thể máy có RAM là 8 GB, 16GB, 32 GB và > 32 GB có phân khúc giá cao hơn hẳn. Trong đó, máy có RAM là 32 GB có khoảng giá rộng hơn so với các máy khác.
- Độ lớn của ổ cứng: máy có ổ cứng 512 GB, 1 TB và > 1 TB có mức giá cao trung bình cao hơn tất cả các máy còn lại. Máy có ổ cứng là 1 TB có khoảng giá khá rộng, chứng tỏ có sự đa dạng về giá. Tuy nhiên, có một điều bất thường, đó là máy có ổ cứng > 1 TB lại có mức giá trung bình thấp hơn máy có ổ cứng 1 TB.
- Loại CPU: CPU loại Apple có khoảng giá cao hơn và cũng có ít outliers hơn. Bên cạnh đón CPU loại Intel cũng có số outliers đáng kể so với các loại CPU còn lại.
- Loại dòng CPU: các dòng CPU cao cấp (như Intel Core i9, Ryzen 9, Apple M1, Apple M2) có mức giá cao hơn các dòng CPU khác.
- Loại Card màn hình: không ngoài dự đoán, máy có card màn hình rời có mức giá nhỉnh hơn so với card màn hình Onboard.
- Tình trạng sử dụng của máy: máy mới có xu hướng có giá cao hơn, và máy đã qua sửa chữa sẽ có giá thấp hơn so với máy chưa qua sửa chữa

Bên cạnh phân tích dựa trên trực quan dữ liệu, nhóm cũng có tiến hành tính toán một số thông số để cũng cố thêm giả thuyết phân tích.

	Variable name	F-test	P-value
0	pc_ram	236.886649	0.000000e+00
1	elt_condition	189.865192	6.553893e-81
2	pc_drive_capacity	173.238715	0.000000e+00
3	pc_cpu_label	171.129723	7.332339e-107
4	pc_vga	170.881872	4.321303e-73
5	pc_cpu	120.844553	0.000000e+00
6	classify	92.396042	9.826792e-22
7	elt_warranty	82.308319	3.730625e-115
8	desktop_screen_size	40.420181	1.725495e-63
9	pc_brand	25.754897	8.159431e-71
10	elt_origin	23.521912	5.470760e-40
11	pc_model	16.620947	8.723770e-249

Bảng 2. Bảng độ ảnh hưởng của các biến phân loại (sử dụng ANOVA)

Dựa vào bảng trên, ta thấy giá máy tính có thể bị chi phối mạnh bởi độ lớn của RAM, tình trạng sử dụng, độ lớn bộ nhớ của ổ cứng, loại cpu, dòng cpu và card màn hình. Điều này là đồng thuận với kết quả phân tích bằng trực quan.

7. KẾT QUẢ PHÂN TÍCH

Loại bỏ một số dòng (outliers) với mức giá lớn hơn 65 triệu VND.

Sau tiến hành thực nghiệm, nhóm thu được kết quả như sau:

	model	r2	mse	mae
0	Histogram Gradient Boosting	0.709438	1.214390e+13	2.247683e+06
1	XGBoost	0.704299	1.235868e+13	2.246087e+06
2	Random Forest	0.642799	1.492903e+13	2.440168e+06
3	KNN	0.545041	1.901478e+13	2.673050e+06
4	multi_mmscale	0.261507	3.086497e+13	3.723290e+06
5	multi_stscale	0.261507	3.086497e+13	3.723290e+06
6	multi	0.261507	3.086497e+13	3.723290e+06
7	Decision Tree	0.244301	3.158409e+13	3.116065e+06
8	multi_norm	0.221013	3.255738e+13	3.879834e+06
9	SVR	-0.050502	4.390522e+13	4.175250e+06
10	Neural Network	-1.032693	8.495543e+13	6.558531e+06

Bảng 3. Kết quả đối với mỗi model sau khi training

Kết quả cho thấy rằng cả ba thuật toán tốt nhất đều thuộc loại Ensemble, trong đó hai thuật toán hàng đầu đều sử dụng Gradient Boosting. Hiệu suất của chúng không có sự chênh lệch đáng kể. Boosting, một kỹ thuật thuộc nhóm ensemble, thay vì xây dựng cùng lúc nhiều cây như Random Forest, thì các thuật toán boosting sẽ lần lượt thêm từng cây vào ensemble và sửa lỗi dự đoán của các cây trước đó. Gradient Boosting là một biến thể tổng quát hóa của Boosting, kết hợp việc sử dụng hàm mất mát (loss function) nhất định. Do đó, các thuật toán Gradient Boosting ensemble thường được ưa chuộng trong bài toán dự đoán với dữ liệu dạng bảng.

Trong số các thuật toán được thử nghiệm, Histogram Gradient Boosting cho kết quả tốt nhất. Như đã đề cập thì Histogram Gradient Boosting là một thuật toán sử dụng Greadient Boosting. Từ Histogram ở đây chỉ việc "rời rạc hóa" (binning) các giá trị liên tục, qua đó giúp cải thiện hiệu xuất của mô hình.

Histogram Gradient Boosting không yêu cầu phải scale dữ liệu về chung một khoảng giá trị. Bên cạnh đó thuật toán này, được thiết kết để có thể sử dụng tốt cho cả biến phân loại và biến dạng số. Do đó, thuật toán này có thể phù hợp với bài toán đang được đặt ra.

Sau khi thực hiện fine tune, hiệu suất của mô hình đã được cải hiện so với cài đặt ban đầu. Với kết quả đạt được như sau:

- R2: 0.7188616525480944
- MSE: 11750040359137.404
- MAE: 2253531.969568334

Như vậy, mô hình dự đoán giá máy với mức chênh lệch trung bình là khoảng 2 triệu 253 VND.

Bên cạnh đó, nhóm phát hiện rằng việc tỉ lệ hóa các tham số về cùng một khoảng giá trị không mang lại lợi ích đáng kể trong việc xây dựng mô hình dự đoán, đặc biệt là khi dữ liệu số không có sự chênh lệch quá nhiều về giá trị. Thực nghiệm trên thuật toán Linear Regression cho thấy không có sự khác biệt đáng kể giữa việc sử dụng MinMaxScaler, StandardScaler và việc không tỉ lệ hóa dữ liệu.

8. KẾT LUẬN

Dữ liệu được thu thập từ trang Chợ Tốt bao gồm thông số của cả laptop và desktop. Sau quá trình thăm dò sơ bộ, chúng tôi thực hiện quá trình làm sạch và tiền xử lý dữ liệu. Tiếp theo, chúng tôi tiến hành thăm dò chi tiết trên dữ liệu đã xử lý và lựa chọn các biến số phù hợp nhất để xây dựng mô hình dự đoán.

Sử dụng 12 biến để huấn luyện mô hình, đó là: pc_brand (nhãn hiệu), pc_model (dòng máy), elt_condition (tình trạng sử dụng), elt_warranty (thời gian bảo hành), desktop_ screen_size (kích thước màn hình), pc_cpu (dòng CPU), pc_ram (RAM), pc_vga (card màn hình) pc_drive_capacity (ổ cứng), elt_origin (xuất xứ), classify (loại máy), pc_cpu_label (loại CPU).

Sau khi tiến hành huấn luyện mô hình, thu được kết quả khá khả quan với R2 là 0.71, và với mỗi máy, mô hình dự đoán ra kết quả chênh lệch khoảng 2 triệu 253 VND (MAE là 2253531.96).

TÀI LIỆU THAM KHẢO

[1] Link crawl data laptop

[2] Link crawl data desktop

[3] Tài liệu tham khảo Selenium

[4] Tài liệu tham khảo Pandas

[5] Tài liệu tham khảo Seaborn

PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
10	Trương Thị Thanh Thanh	− Code làm sạch và tiền xử lý dữ liệu. − Code phân tích dữ liệu sơ bộ − Viết báo cáo.
33	Huỳnh Nguyễn Vân Khánh	− Code huấn luyện và đánh giá mô hình. − Code phân tích dữ liệu sơ bộ. − Code thăm dò và trực quan dữ liệu. − Viết báo cáo.
36	Nguyễn Thị Ngọc Nga	− Code crawl dữ liệu từ chợ tốt với cả desktop và laptop. − Viết báo cáo. − Thiết kế slide thuyết trình.