A white rectangular frame with black border

Description automatically generated

**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐAI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA: HỆ THỐNG THÔNG TIN**

----🙞🙜🕮🙞🙜-----

A blue logo with a black background

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**KHO DỮ LIỆU VÀ OLAP**

**Đề tài:**

**GIÁO VIÊN HƯỚNG DẪN:**

KS. Phạm Nguyễn Thanh Bình

**SINH VIÊN THỰC HIỆN:**

Phùng Thiên Phúc – 21521297

Phạm Mạnh Hùng – 21520901

Nguyễn Quang Lâm - 21521057

*Thành Phố Hồ Chí Minh, Tháng 5 Năm 2024*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

................................................................................................................................

……., ngày……...tháng……năm 2024

Người nhận xét

(Ký tên và ghi rõ họ tên)

# LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến toàn thể giảng viên trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM cũng như là nhà trường vì đã giúp nhóm em có những kiến thức cơ bản làm nền tảng để thực hiện đồ án môn học này.

Đặc biệt hơn nữa, chúng em xin gửi lời cảm ơn chân thành đến thầy Phạm Nguyễn Thanh Bình (Giảng viên thực hành môn Khai thác dữ liệu). Thời gian vừa qua, thày đã trực tiếp giảng dạy, truyền đạt kinh nghiệm, hướng dẫn chúng em một cách tận tình giúp nhóm chúng em hoàn thành tốt đồ án môn học của mình. Chúc thầy sẽ luôn dồi dào sức khỏe, tràn đầy nhiệt huyết để có thể tiếp tục giảng dạy, dìu dắt những thế hệ sinh viên tiếp theo.

Ngoài ra, chúng em cũng gửi lời cảm ơn đến tập thể lớp IS252.O21vì thời gian qua đã đồng hành cùng nhau. Cùng nhau học tập, tranh luận một cách sôi nổi để xây dựng bài học một cách tốt nhất. Cảm ơn các bạn đã cùng thảo luận, đánh giá và đóng góp ý kiến, cùng học hỏi nghiên cứu để thực hiện đồ án một cách tốt nhất có thể.

Mặc dù đã vận dụng tối đa những gì đã học được nhưng chúng em vẫn khó có thể tránh khỏi những sai sót. Chính vì vậy, nhóm em rất mong nhận được sự góp ý từ phía thầy để có thể hoàn thiện một cách tốt nhất có thể. Qua đó cũng tích lũy và học hỏi kinh nghiệm để làm hành trang cho tương lai.

Chúng em xin hết và xin một lần nữa gửi lời cảm ơn chân thành đến với thầy!

# Chương 1. Nhận diện bài toán khai thác dữ liệu

## Tổng quan về dữ liệu

### Lý do chọn đề tài

Trong những năm qua với sự phát triển của thị trường công nghiệp oto trên thế giới nói chung và Việt Nam nói riêng đã góp phần vào sự phát triển nền kinh tế toàn cầu. hiểu được vấn đề đó. Nhóm đã quyết định xây dựng chọn đề tài "Dự đoán giá bán xe thực tế trên thị trường"

Hơn nữa, trong thời đại số hóa hiện nay, việc ứng dụng các kỹ thuật khai thác dữ liệu và học máy để dự đoán thị trường không chỉ mang lại lợi ích về mặt lý thuyết mà còn có giá trị thực tiễn cao. Các doanh nghiệp có thể sử dụng những kết quả từ nghiên cứu này để điều chỉnh chiến lược kinh doanh, tối ưu hóa sản xuất và phân phối, cũng như cải thiện dịch vụ khách hàng.

Đồng thời, những thông tin dự đoán này cũng hỗ trợ người tiêu dùng trong việc đưa ra quyết định mua sắm thông minh và hiệu quả hơn.

Nhóm mong muốn đề tài này không chỉ mang lại những hiểu biết sâu sắc về thị trường ô tô mà còn cung cấp các công cụ phân tích mạnh mẽ để dự đoán và thích ứng với những thay đổi trong tương lai, qua đó góp phần vào sự phát triển bền vững của ngành công nghiệp ô tô.

### Giới thiệu nguồn dữ liệu

Dataset được lấy từ nền tảng Kaggle, được xây dựng bởi SYED ANWAR. Được cập nhật gần nhất vào tháng 2/2024.

Dataset gồm có 16 cột và 558844 dòng về thông tin, giá bán, hang sản xuất

Dataset car\_prices cung cấp một bộ sưu tập thông tin toàn diện liên quan đến các giao dịch bán các loại xe khác nhau. Bộ dữ liệu này bao gồm các chi tiết như năm, hãng sản xuất, mẫu xe, trang trí, loại thân xe, loại hộp số, số VIN (Số nhận dạng xe), trạng thái đăng ký, xếp hạng tình trạng, đọc đồng hồ đo đường, màu sắc ngoại thất và nội thất, thông tin người bán, Báo cáo thị trường Manheim ( MMR), giá bán và ngày bán.

Link tải dataset: <https://www.kaggle.com/datasets/syedanwarafridi/vehicle-sales-data>

## Mô tả thuộc tính

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên thuộc tính | Ý nghĩa | Kiểu dữ liệu |
| 1 | Year | Năm sản xuất |  |
| 2 | Make | Tên hãng sản xuất xe |  |
| 3 | Model | Mẫu xe cụ thể do hang sản xuất |  |
| 4 | Trim | Mỗi mẫu xe sẽ có các phiên bản khác nhau, mỗi phiên bản sẽ được thể hiện trong cột trim |  |
| 5 | Body | Loại than xe, vd như SUV, Sedan, coupe… |  |
| 6 | Transmission | Hộp số của xe. Có giá trị có thể là tự động (automatic) hoặc số sàn(manual) |  |
| 7 | Vin | Số nhận dạng xe (Vehicle Identification Number). Đây là mã số duy nhất cho mỗi xe và có thể dùng để tra cứu thông tin chi tiết về xe. |  |
| 8 | State | Bang hoặc khu vực nơi xe được bán. |  |
| 9 | Condition | Tình trạng của xe, ví dụ như mới, đã qua sử dụng, hoặc bị hư hỏng. |  |
| 10 | Odometer | Số dặm hoặc số km mà xe đã chạy. Odometer càng cao thường cho thấy xe đã qua sử dụng nhiều và có thể giảm giá trị. |  |
| 11 | Color | Màu sơn bên ngoài của xe. |  |
| 12 | interior | Màu hoặc tình trạng của nội thất xe. |  |
| 13 | Seller | Người bán hoặc đại lý bán xe. |  |
| 14 | Mmr | Giá trị thị trường dự đoán (Manheim Market Report) của xe. Đây là giá trị tham khảo để đánh giá giá trị thị trường của xe, giúp phân tích xu hướng và sự biến động của thị trường. |  |
| 15 | Sellingprice | Giá bán thực tế của xe trong giao dịch. Giá này cho biết mức giá mà người mua đã trả cho xe. |  |
| 16 | saledate | Ngày diễn ra giao dịch bán xe. Ngày này có thể dùng để phân tích xu hướng bán hàng theo thời gian. |  |

## Phát biểu bài toán

Dựa vào những thuộc tính của dataset, chúng ta sẽ dự đoán xu hướng mua xe của người tiêu dùng để giúp các đại lý bán xe đề ra chiến lược nhập hàng bán và cho các doanh nghiệp vạch ra chiến lược sản xuất

## Công cụ khai thác dữ liệu

Các công cụ khai thác dữ liệu:

* Google colab
* Anaconda – Jupiter notebook
* Python

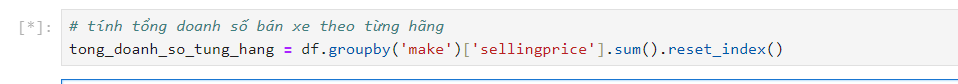
## Thư viện kèm theo

* **NumPy:** được sử dụng để thực hiện các phép tính số hiệu quả và làm việc với mảng đa chiều.
* **Pandas:** được sử dụng để thao tác, tiền xử lý và phân tích dữ liệu, cung cấp các cấu trúc và chức năng dữ liệu mạnh mẽ.
* **Seaborn:** được sử dụng để tạo trực quan hóa dữ liệu mang tính thông tin, chất lượng cao, xây dựng trên thư viện matplotlib.
* **Matplotlib:** là một thư viện toàn diện để tạo trực quan hóa dữ liệu một cách tĩnh, hoạt họa và tương tác trong môi trường Python.
* **Scikit-learn (sklearn):** là một thư viện máy học cung cấp nhiều thuật toán học có giám sát và không giám sát, cũng như các công cụ để tiền xử lý dữ liệu và đánh giá mô hình.

# Chương 2. Trực quan hóa dữ liệu

## Thống kê hãng xe nào có doanh số bán xe nhiều nhất

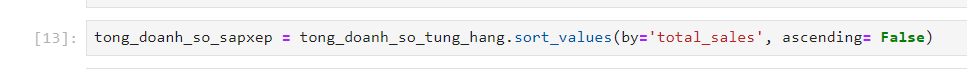
Tính tổng doanh số bán xe của từng hãng



Chúng ta thực hiện đổi tên cột make trong biến tong\_doanh\_so\_tung\_hang thành total\_sales để dễ hình dung



Thực hiện sắp xếp hang xe theo thứ tự giảm dần theo giá trị total\_sales



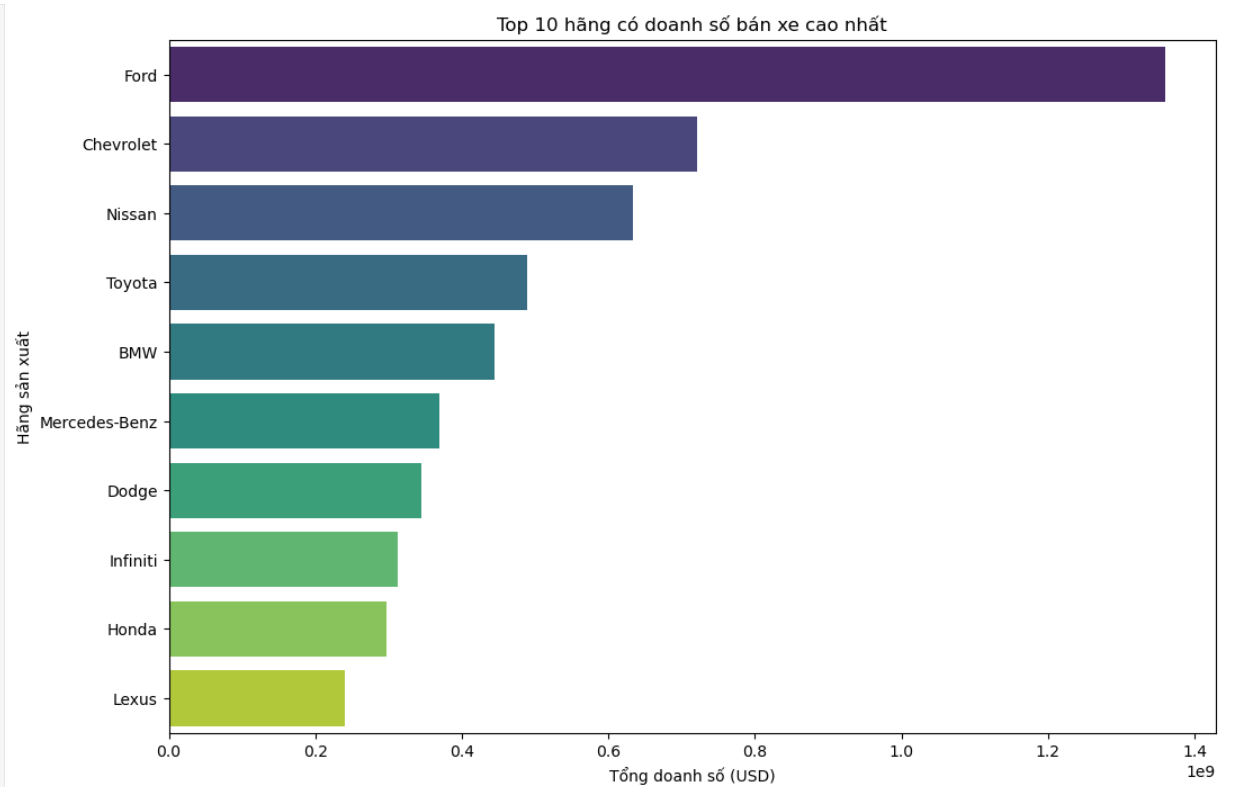
Gán 10 giá trị lớn nhất vào biến gia\_tri\_x



Thực hiện vẽ biểu đồ cột



Kết quả chúng ta có thể thấy hãng xe ford có doanh thu cao nhất(gấp đôi hãng Chevrolet đứng thứ 2). Đứng cuối bảng xếp hạng là Honda và Lexus



## Trực quan hóa được xu hướng mua xe của người tiêu dùng từ năm 2005 đến 2015

Trích xuất năm từ cột saledate



Xử lý các giá trị NaN trước khi chuyển đổi sang kiểu số nguyên



Loại bỏ các hàng có giá trị năm không hợp lệ (ví dụ: năm 0)



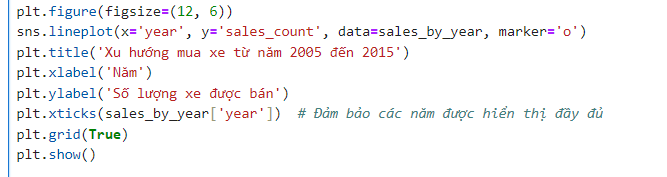
Lọc dữ liệu từ năm 2005 đến 2015



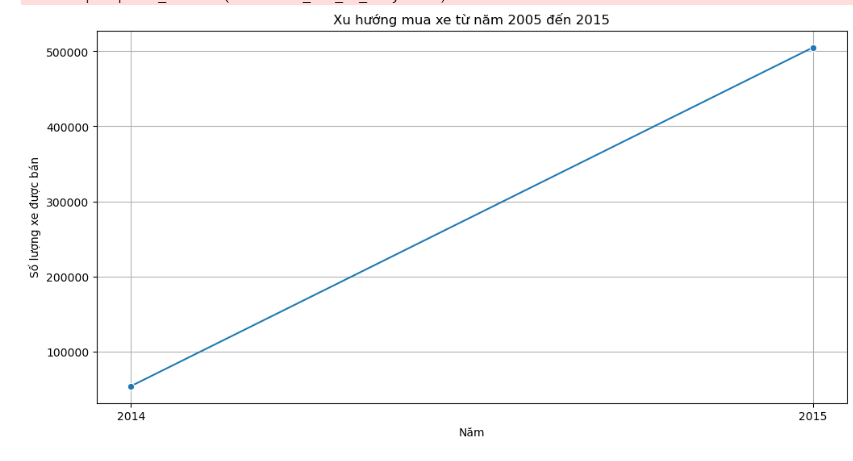
Nhóm dữ liệu theo năm và đếm số lượng xe được bán ra mỗi năm



Vẽ biểu đồ đường

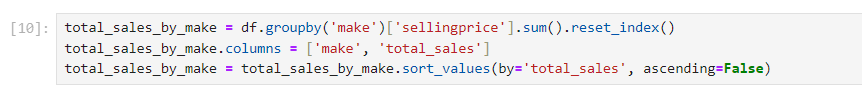


Kết quả



## Phân tích giá bán theo theo từng địa điểm của 10 hãng sản xuất oto có doanh số bán xe nhiều nhất

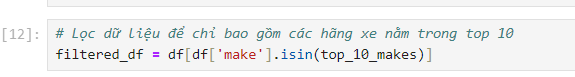
Tính doanh số bán xe theo từng hãng



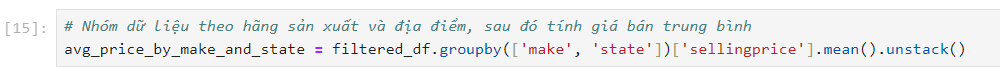
Chọn ra 10 hãng có doanh số cao nhất và lưu nó vào biến top\_10\_makes



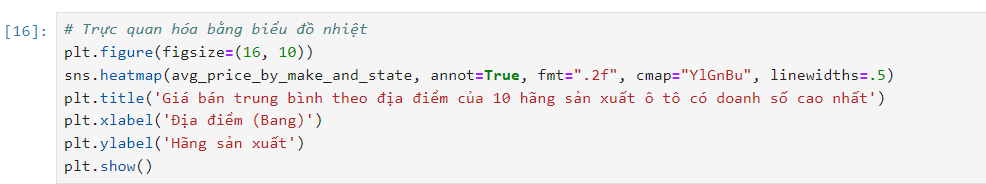
Lọc dữ liệu để chỉ bao gồm các hãng xe nằm trong top 10.



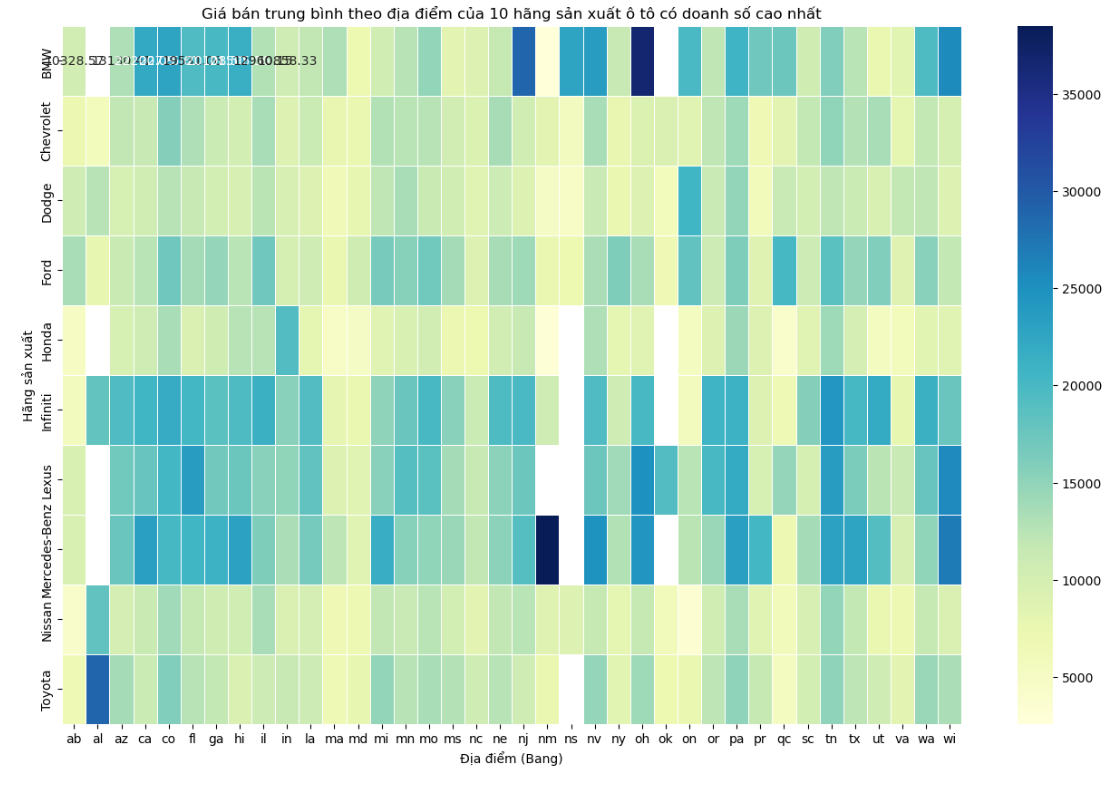
Nhóm dữ liệu theo hãng sản xuất và địa điểm, sau đó tính giá bán trung bình.



Trực quan hóa kết quả bằng biểu đồ hộp (box plot)



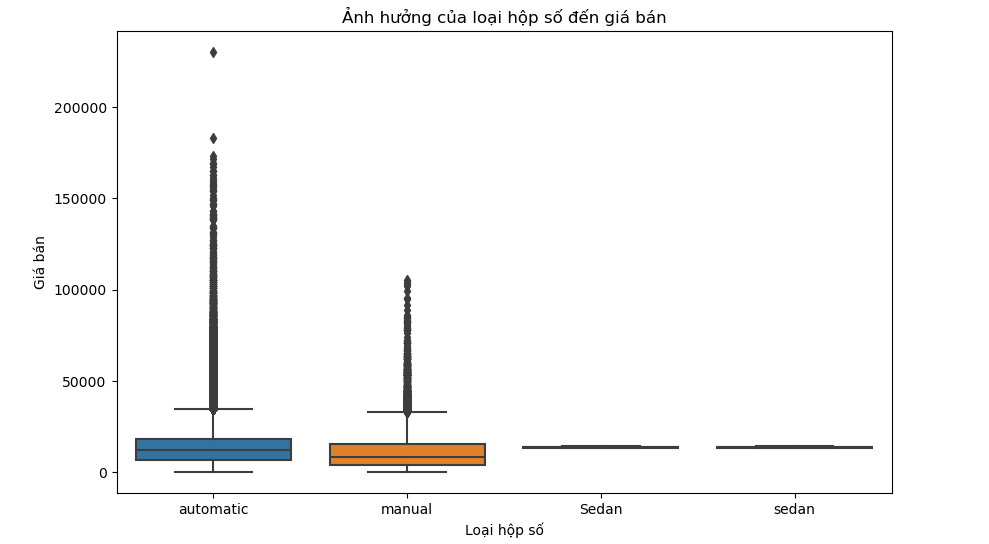
Kết quả



## Ảnh hưởng của loại hộp số đến giá bán

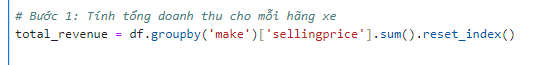
Vẽ biểu đồ boxplot trực quan hóa ảnh hưởng của odoometer 

Kết quả



## Giá trị thị trường dự đoán (MMR) so với giá bán thực tế

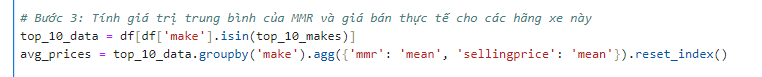
Tính tổng doanh thu cho mỗi hãng xe



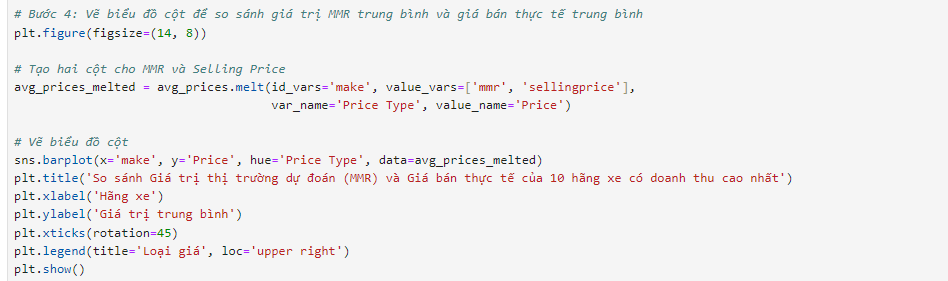
Lọc ra 10 hãng xe có doanh thu cao nhất



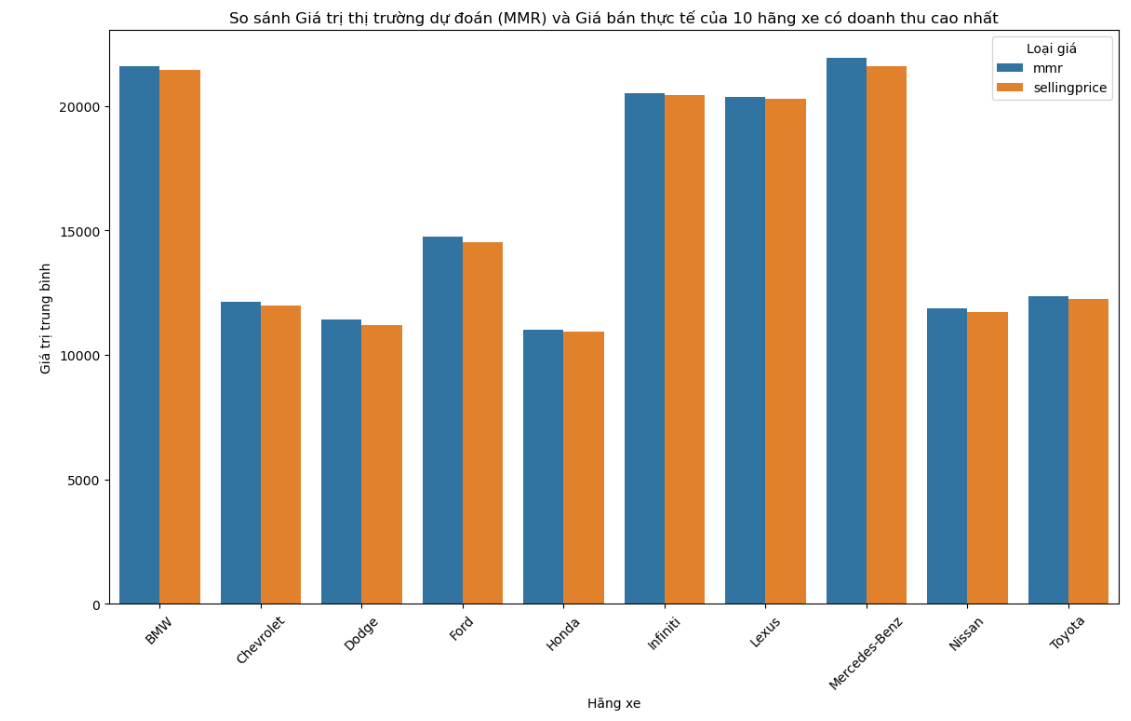
Tính giá trị trung bình của MMR và giá bán thực tế cho các hãng xe này



Vẽ biểu đồ cột để so sánh giá trị MMR trung bình và giá bán thực tế trung bình

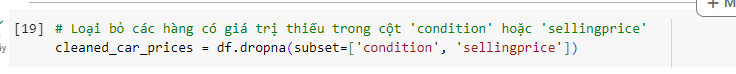


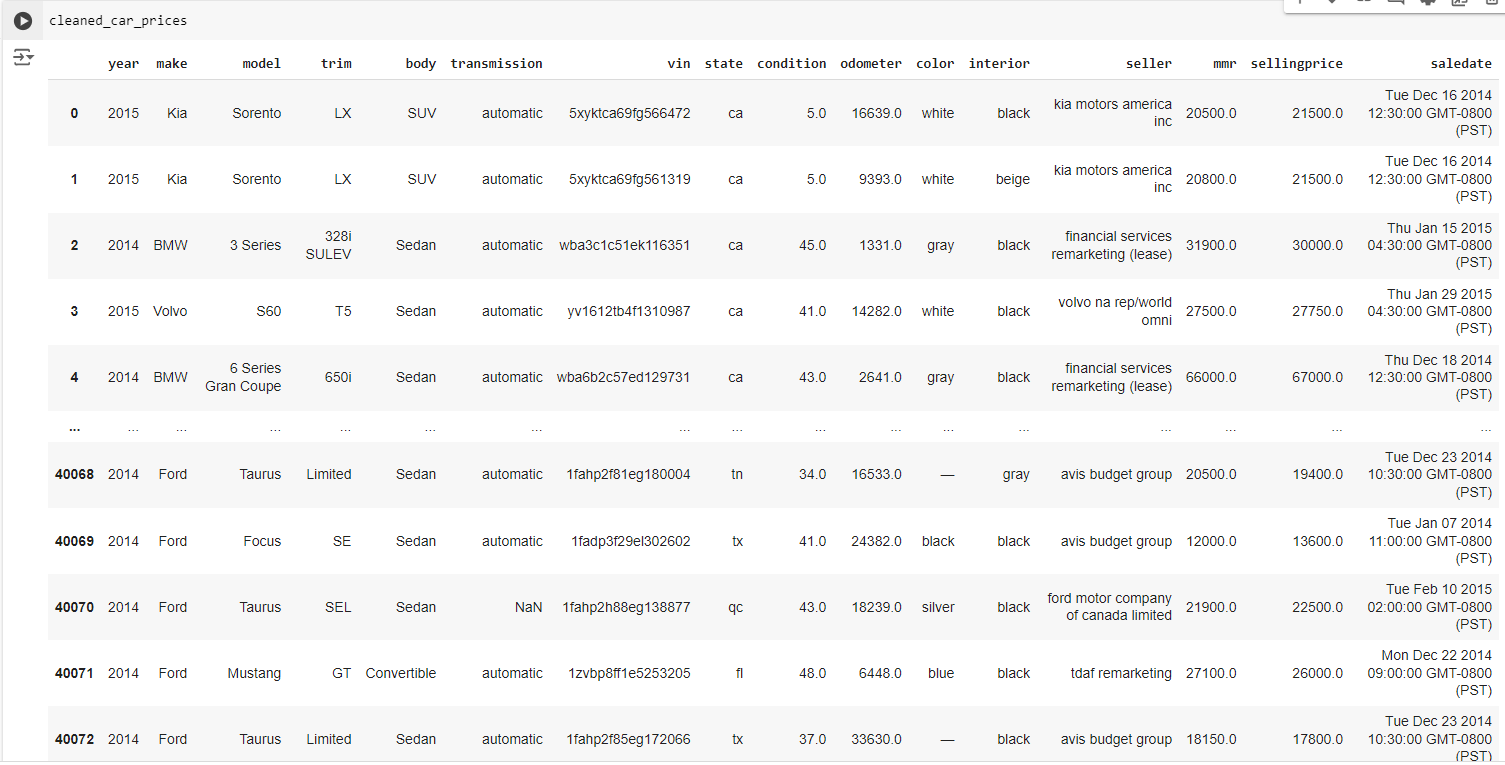
Kết quả



## Tình trạng xe ảnh hưởng đến giá bán:

Loại bỏ các hàng có giá trị thiếu trong cột ‘condition’ hoặc ‘sellingprice’

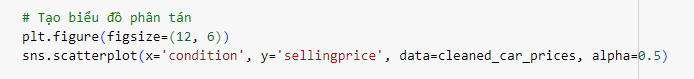
Kết quả

Vẽ biểu đồ phân tán để cho thấy ảnh hưởng của tình trạng xe đến giá bán

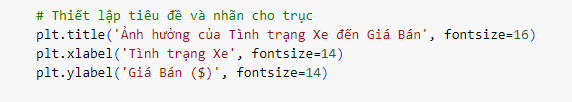
* Thiết lập kiểu của biểu đồ:



* Tạo biểu đồ phân tán



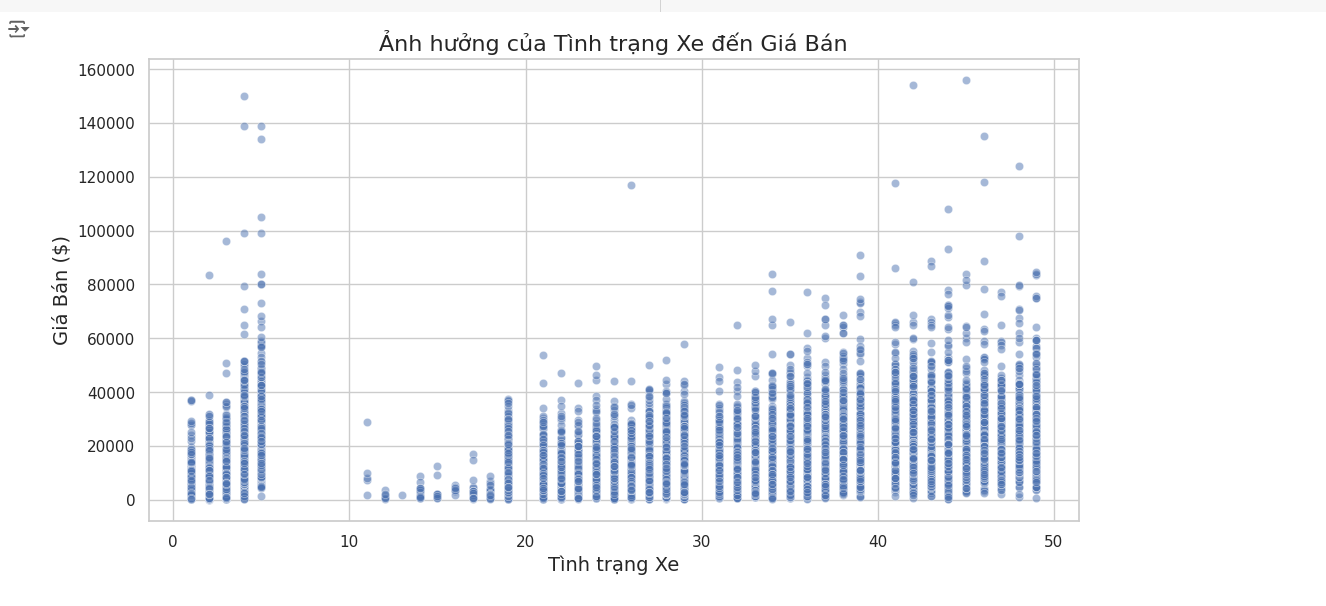
* Thiết lập tiêu đề và nhãn cho trục biểu đồ



* Hiển thị biểu đồ

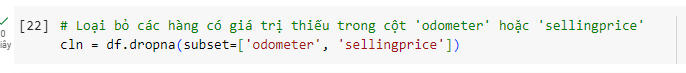


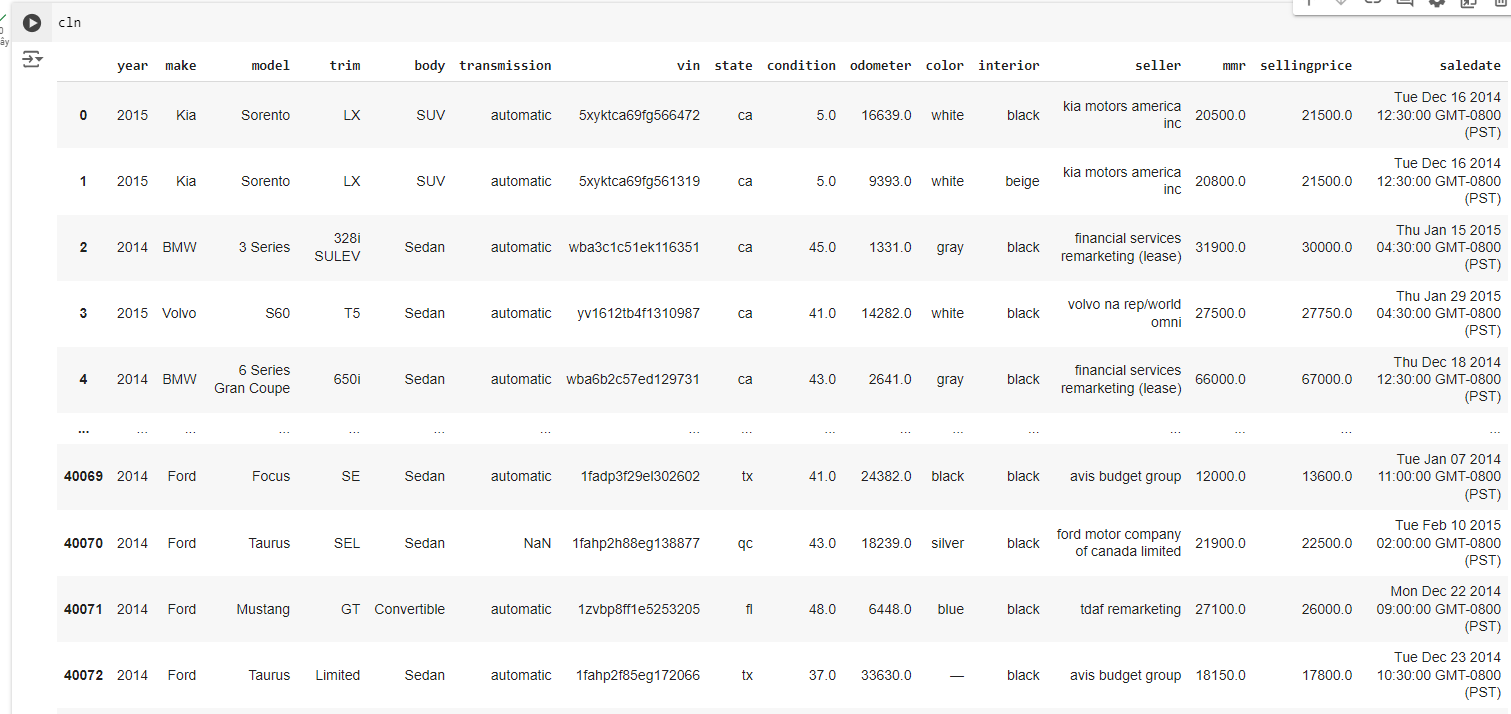
Kết quả



## Ảnh hưởng của số km đã đi (odometer) đến giá bán:

Loại bỏ các giá trị thiếu trong cột odometer và sellingprice

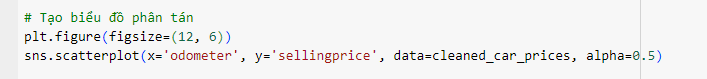
Kết quả

Vẽ biểu đồ biểu thị ảnh hưởng của số km đến giá bán

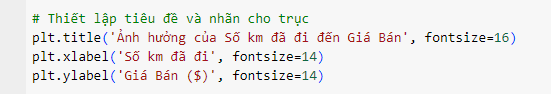
* Thiết lập kiểu của biểu đồ



* Tạo biểu đồ phân tán



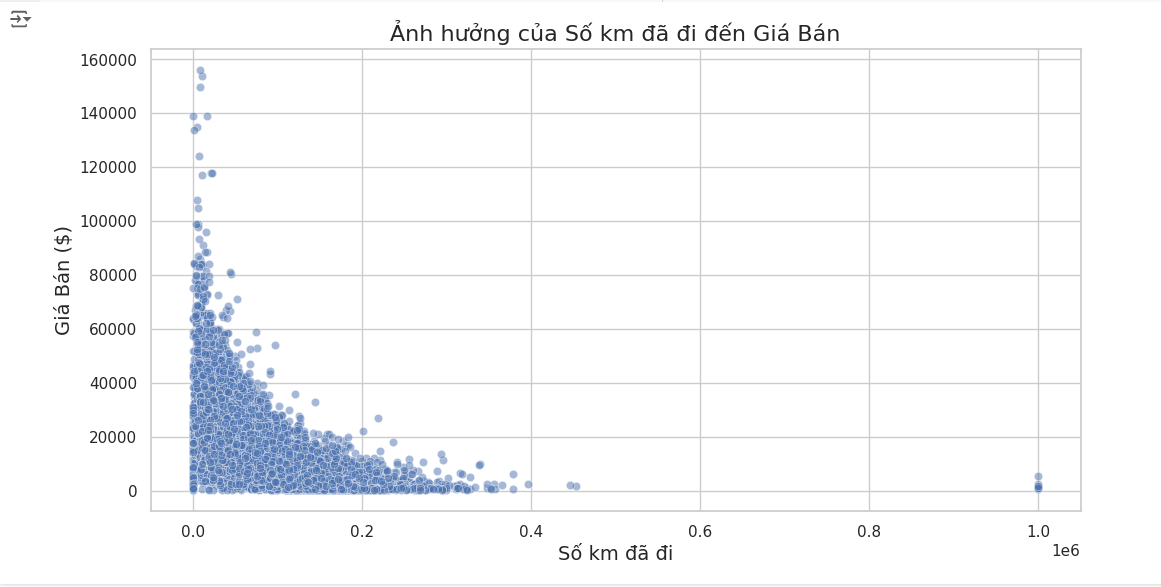
* Thiết lập tiêu đề và nhãn cho trục



* Hiển thị biểu đồ

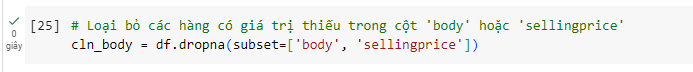


Kết quả

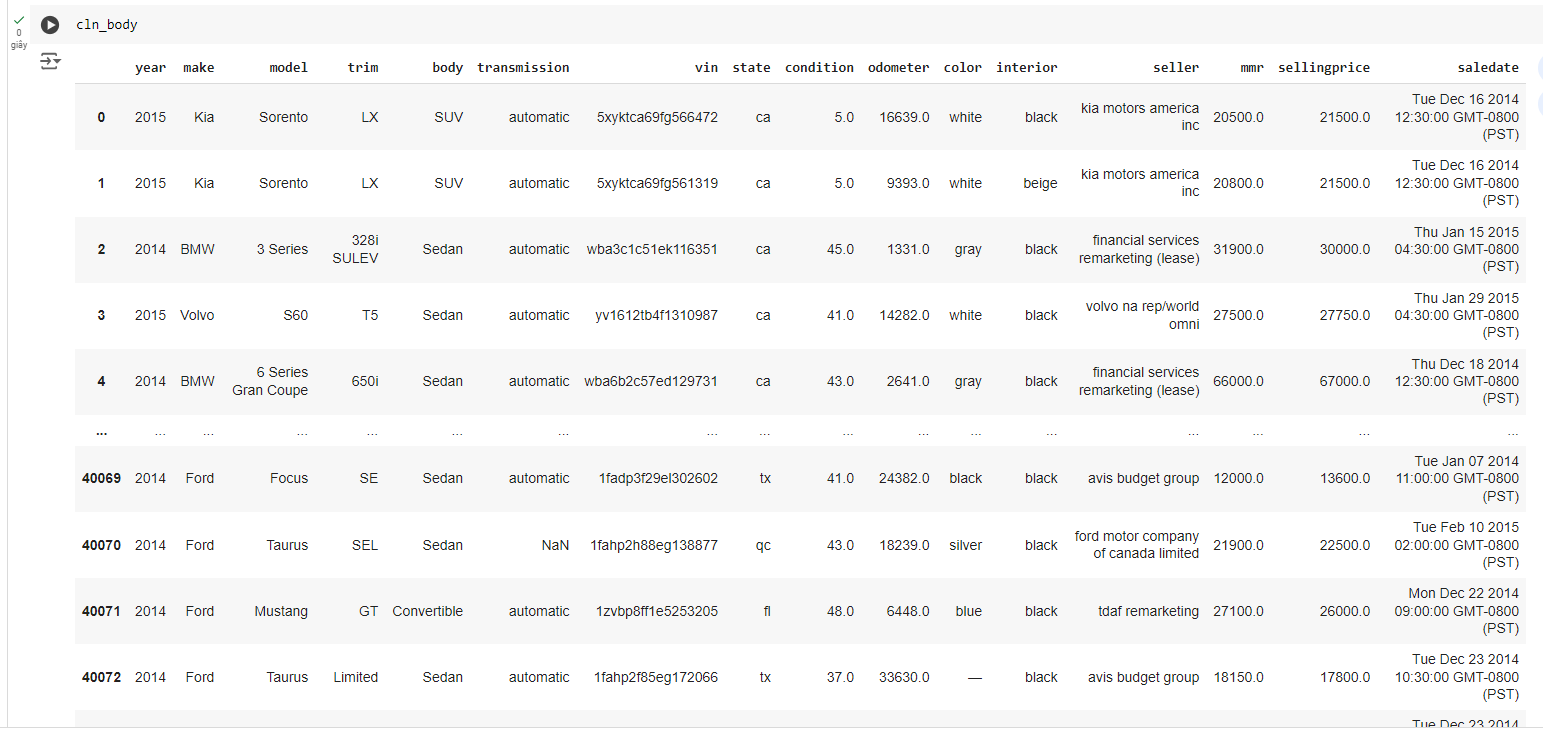


## Giá bán theo các loại thân xe (body):

Loại bỏ các giá trị thiếu trong cột body và sellingprice

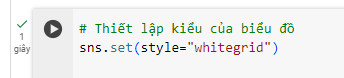


Kết quả

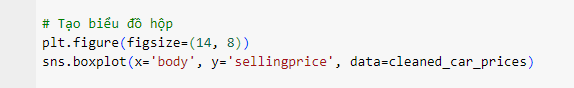


Vẽ biểu đồ hiển thị giá bán theo thân xe

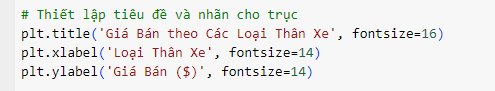
* Thiết lập kiểu biểu đồ



* Tạo biểu đồ hộp



* Thiết lập tiêu đề và nhãn cho trục



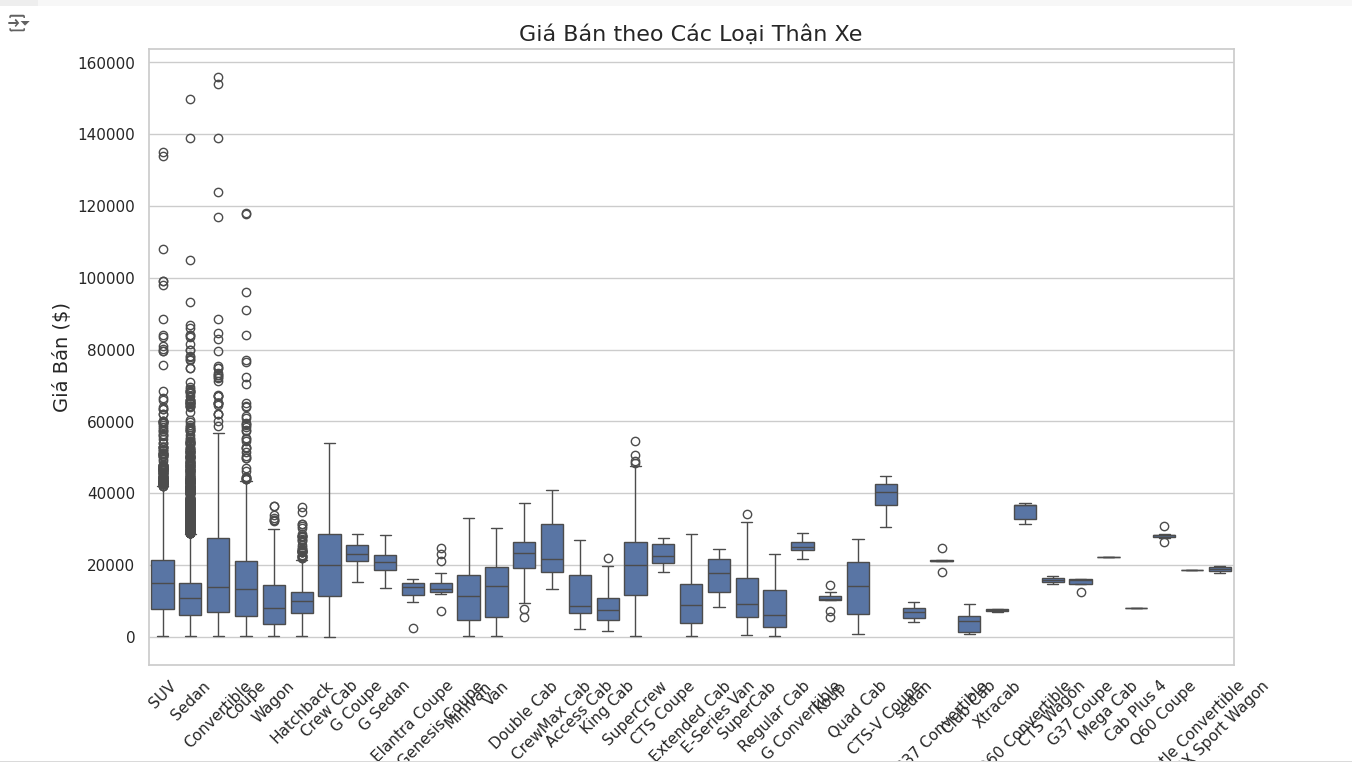
* Xoay nhãn trục x cho dễ cho đọc hơn ( Optional)



* Hiển thị biểu đồ

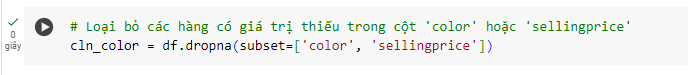


Kết quả



## Phân bố giá bán theo màu sắc xe:

Loại bỏ các giá trị thiếu trong cột color và sellingprice

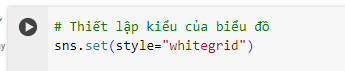


Kết quả

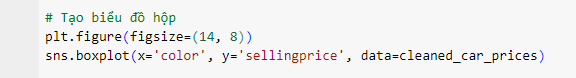


Vẽ biểu đồ phân bố giá bán theo màu sắc xe

* Thiết lập kiểu biểu đồ



* Tạo biểu đồ hộp



* Thiết lập tiêu đề và nhãn cho trục



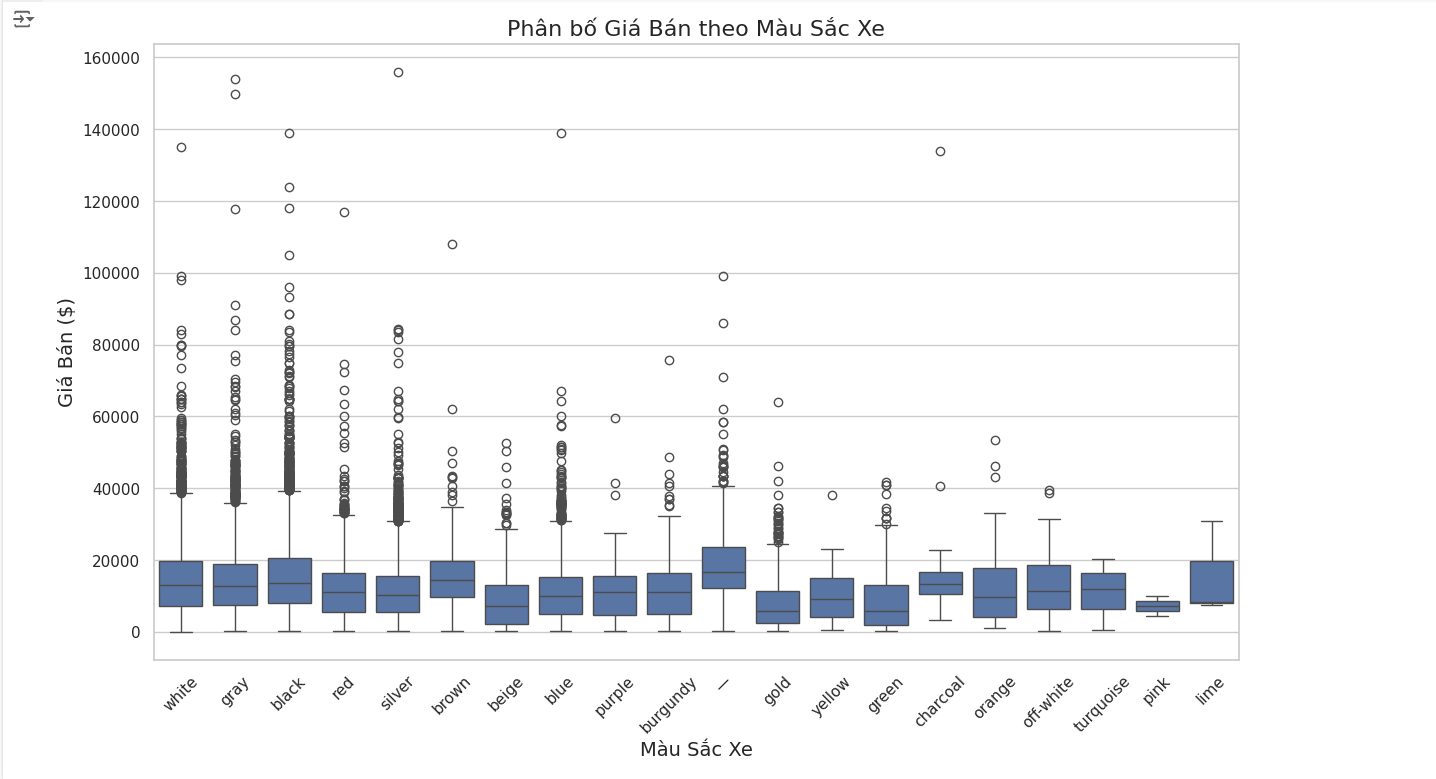
* Xoay nhãn trục x cho dễ cho đọc hơn ( Optional)



* Hiển thị biểu đồ



Kết quả

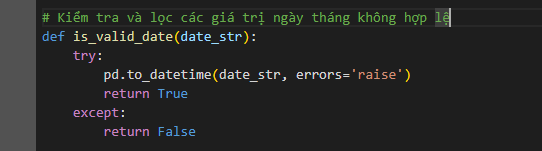


## Xu hướng giá bán theo ngày bán (saledate):

Loại bỏ các giá trị thiếu trong cột saledate và sellingprice

Chuyển cột saledate sang định dạng datetime

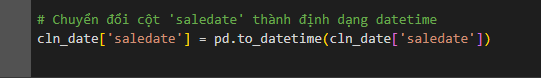
* Kiểm tra và lọc các giá trị ngày tháng không hợp lệ



* Lọc các giá trị ngày tháng hợp lệ

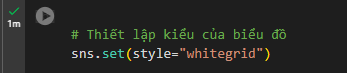


* Chuyển đổi cột saledate sang định dạng datetime

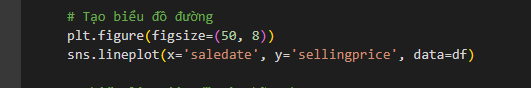


Vẽ biểu đồ thể hiện giá bán theo ngày bán

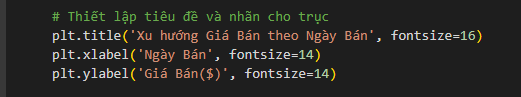
* Thiết lập kiểu biểu đồ



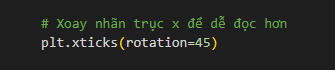
* Tạo biểu đồ đường



* Thiết lập tiêu đề và nhãn cho trục



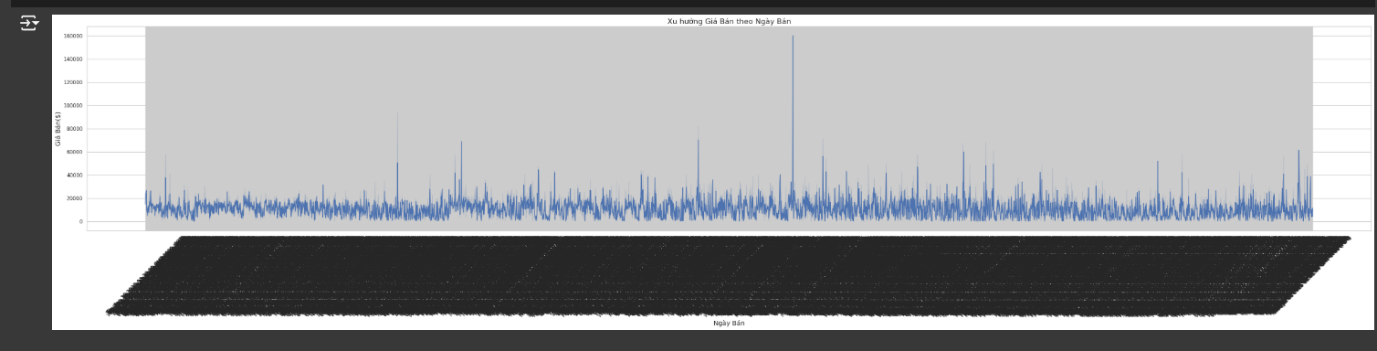
* Xoay nhãn trục x để dễ đọc hơn



* Hiển thị đồ thị

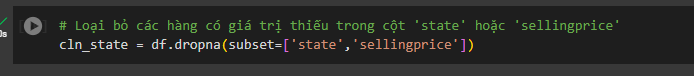


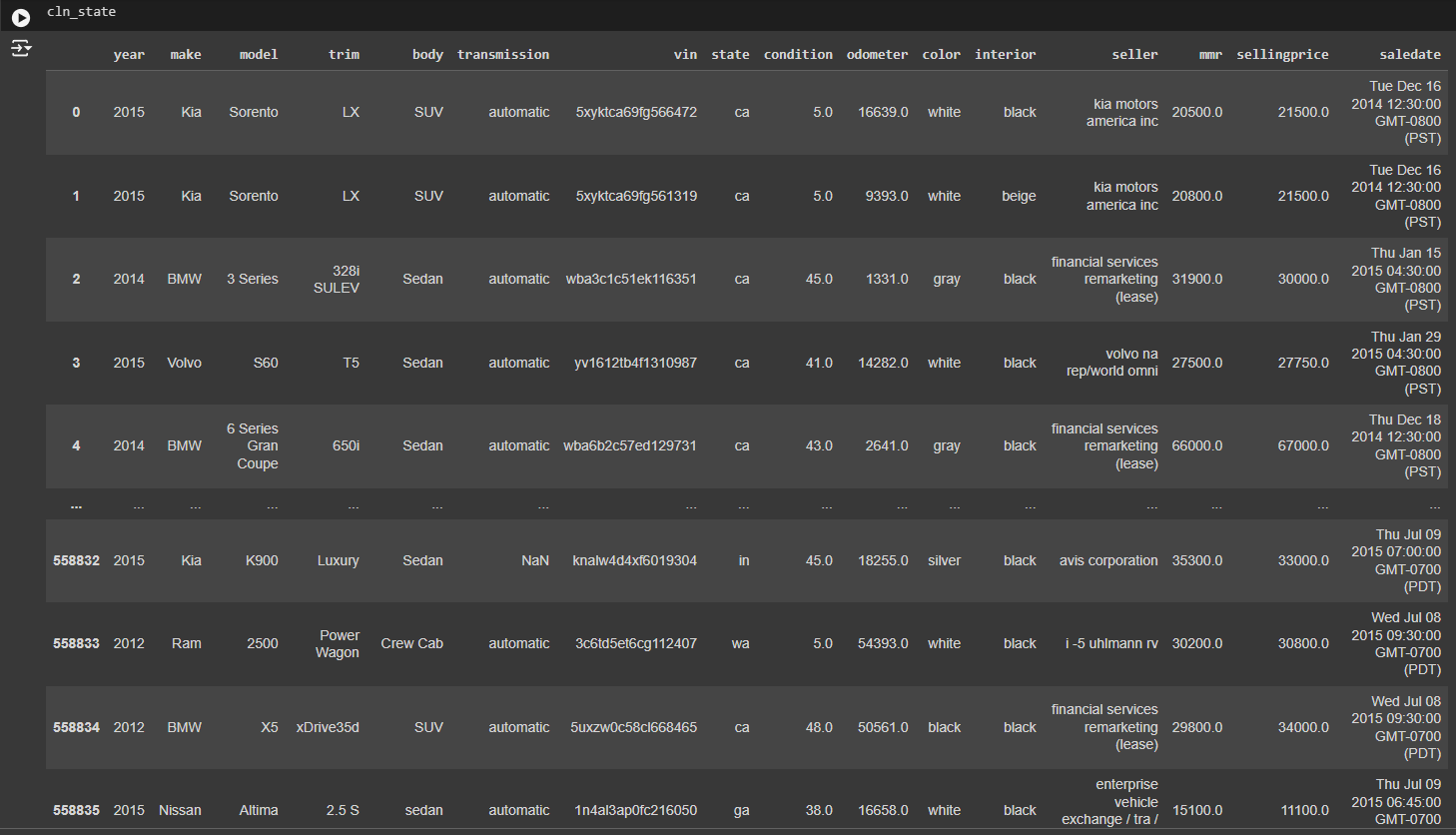
Kết quả



## Phân bố giá bán theo bang

Loại bỏ các giá trị thiếu trong cột state và sellingprice

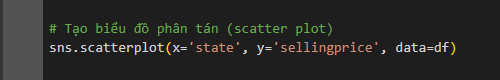
Kết quả

Vẽ biểu đồ hiển thị giá bán phân bố theo bang:

* Thiết lập kiểu biểu đồ



* Tạo biểu đồ phân tán



* Thêm tiêu đề và nhãn trục



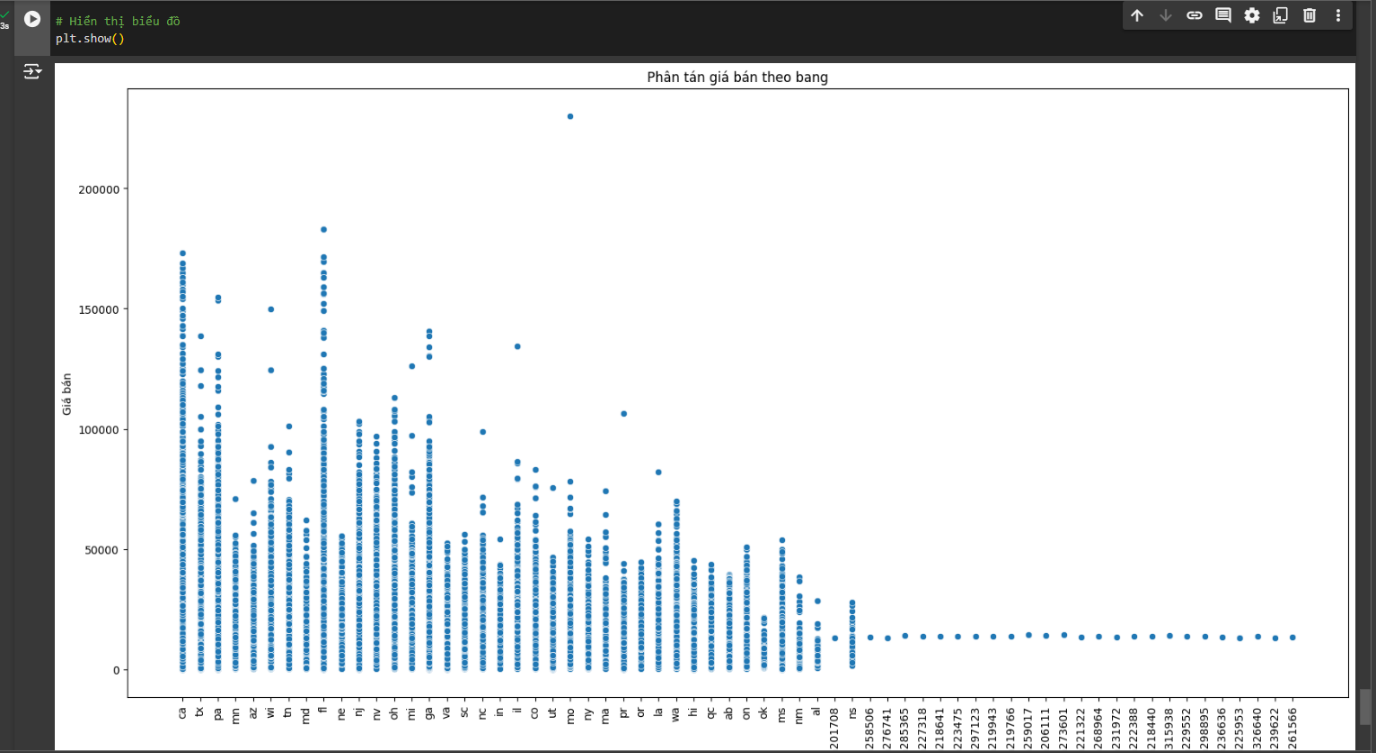
* Xoay nhãn trục x để dễ đọc (Optional)



* Hiển thị biểu đồ



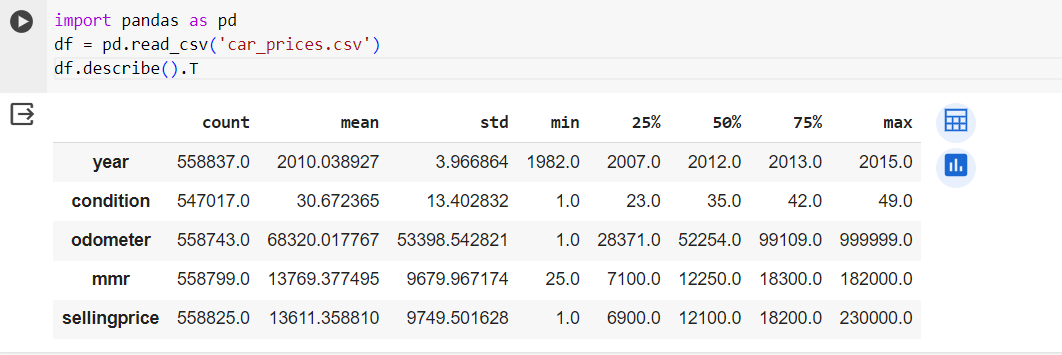
Kết quả



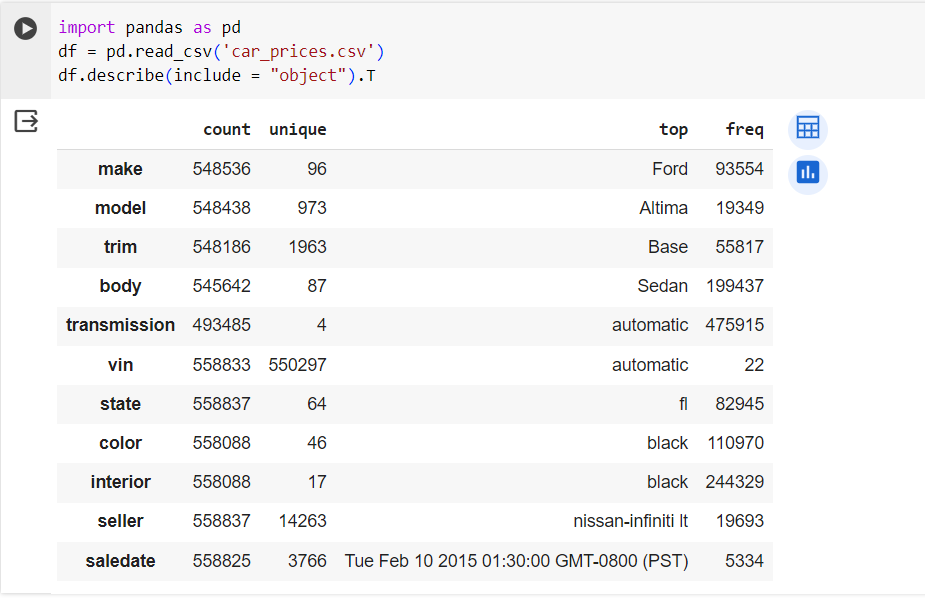
# Chương 3. Tiền xử lý dữ liệu

## Mô tả dữ liệu

Thống kê các chỉ số định tính và định lượng có trong bộ dữ liệu  
Định lượng: giá trị lớn nhất, nhỏ nhất, giá trị trung bình, độ lệch chuẩn và tứ phân vị

Thống kê thông tin chi tiết dữ liệu định lượng

Định tính: tính số giá trị khác nhau trong bộ dữ liệu và giá trị có tần suất cao nhất và tần số của giá trị đó

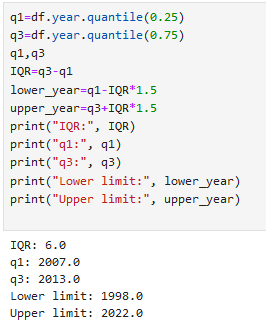
Thống kê thông tin chi tiết dữ liệu định tính

## Làm sạch dữ liệu

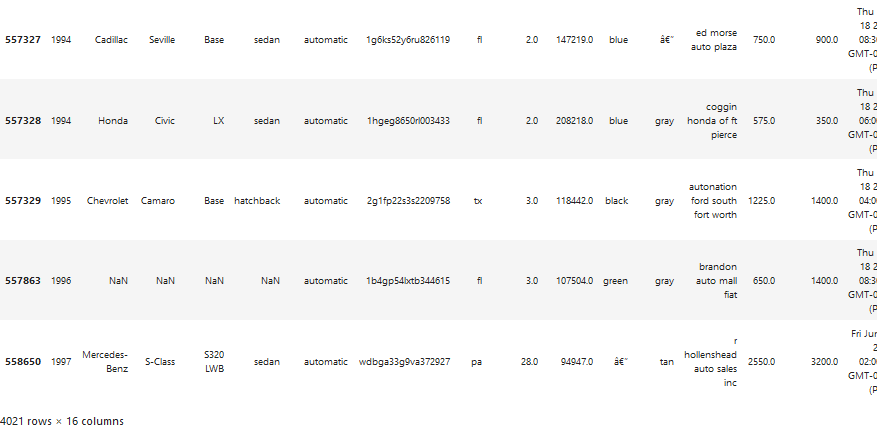
### 2.1. Nhận diện và xử lý các dữ liệu bị nhiễu

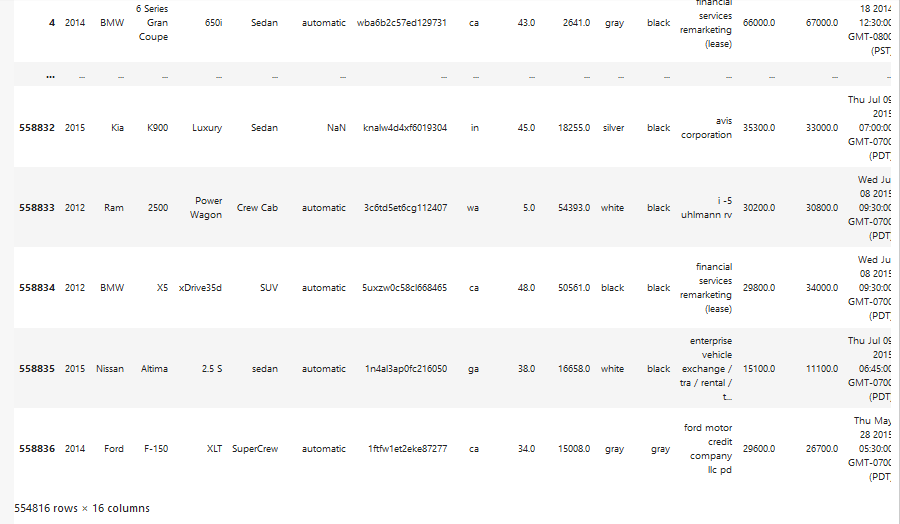
Sử dụng phương pháp IQR để tìm các ngoại lệ có trong các thuộc tính định lượng bằng cách tìm ra phân vị tứ thứ nhất và thứ ba rồi tính ra giới hạn giá trị trên và giới hạn giá trị dưới của từng thuộc tính. Cuối cùng là loại bỏ các dòng dữ liệu không nằm trong vùng giá trị vừa tạo

**Thuộc tính year:**

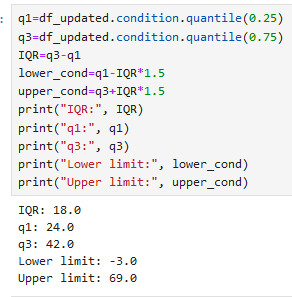


Giới hạn trên và dưới của thuộc tính year

Các dòng dữ liệu ngoại lệ của thuộc tính year

Các dòng dữ liệu còn lại

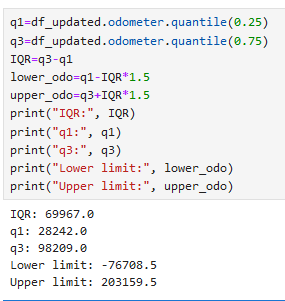
Thuộc tính condition



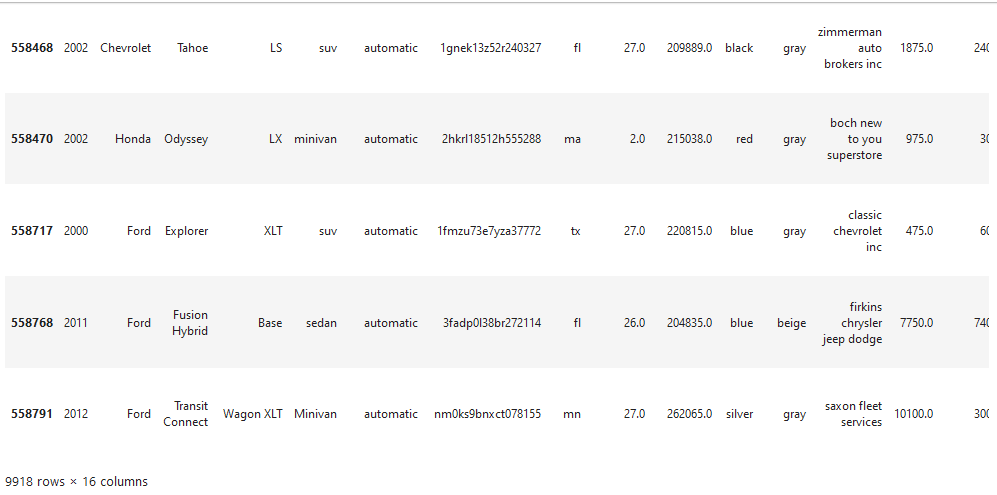
Giới hạn trên và dưới của thuộc tính condition

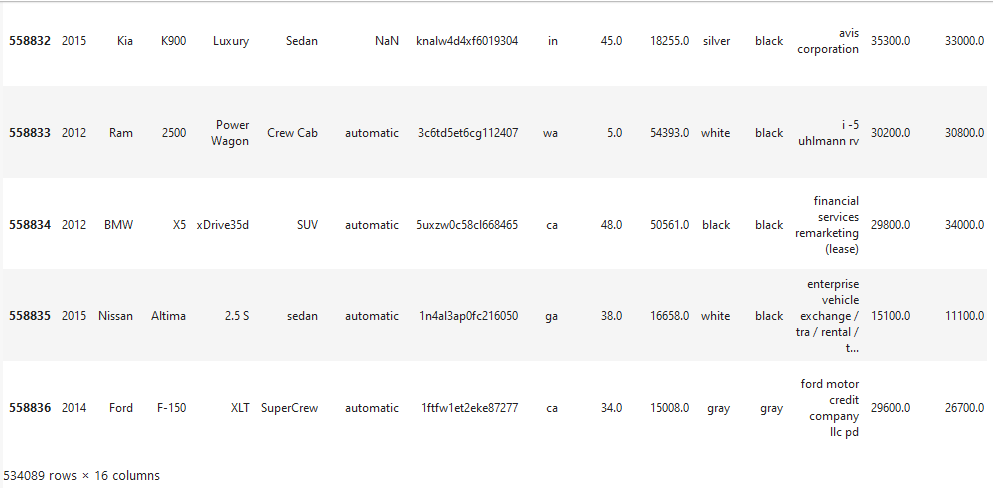
Các dòng dữ liệu ngoại lệ của thuộc tính condition

T**huộc tính odometer**

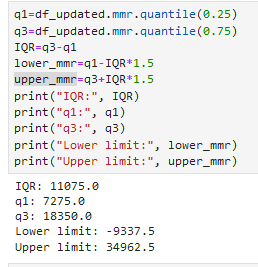


Giới hạn trên và dưới của thuộc tính odometer

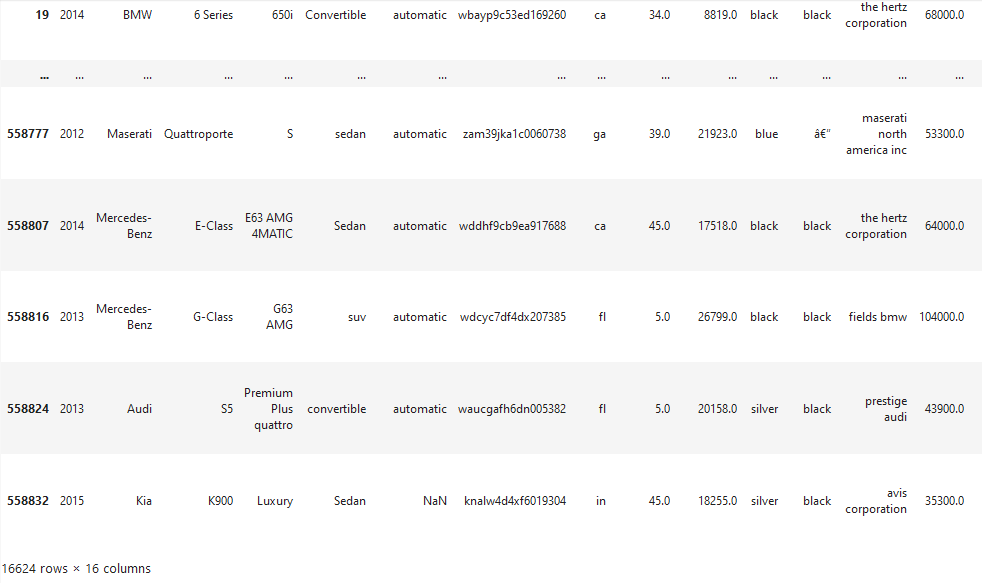
Các dòng dữ liệu ngoại lệ của thuộc tính odometer

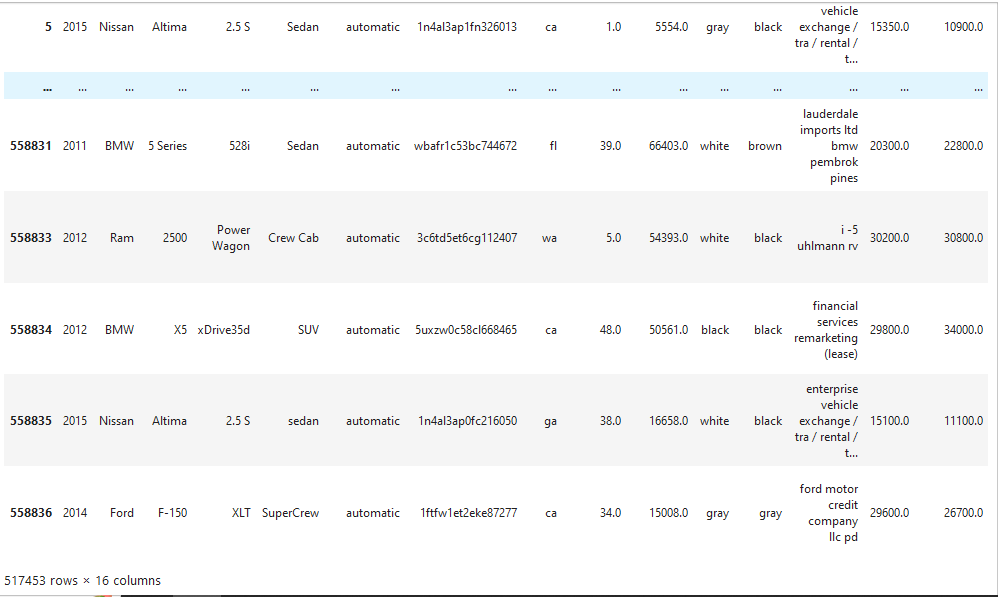
Các dòng dữ liệu còn lại

**Thuộc tính mmr**

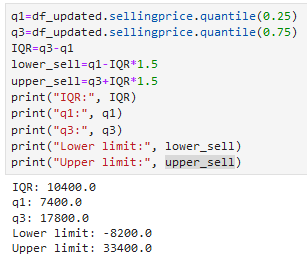


Giới hạn trên và dưới của thuộc tính mmr

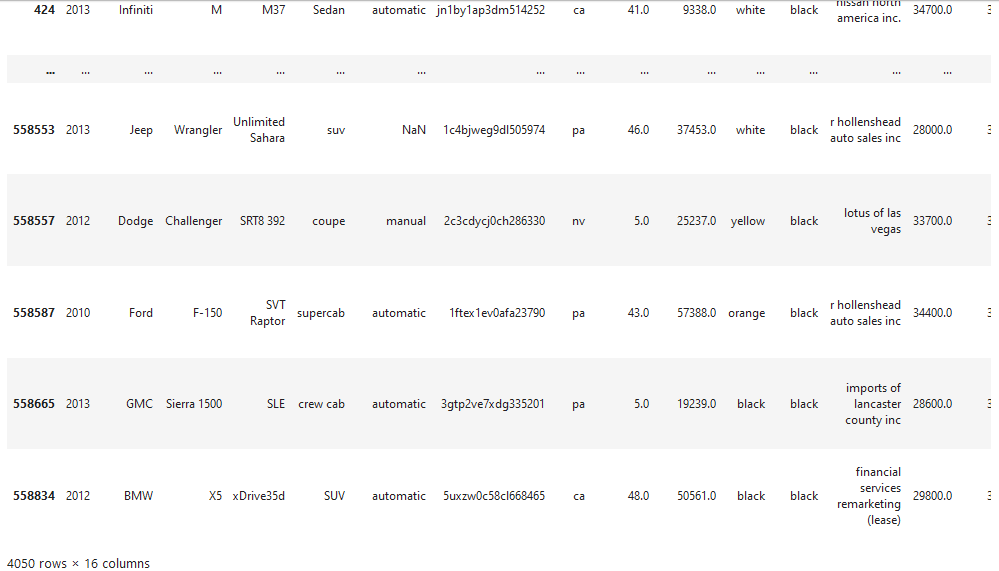
Các dòng dữ liệu ngoại lệ của thuộc tính odometer

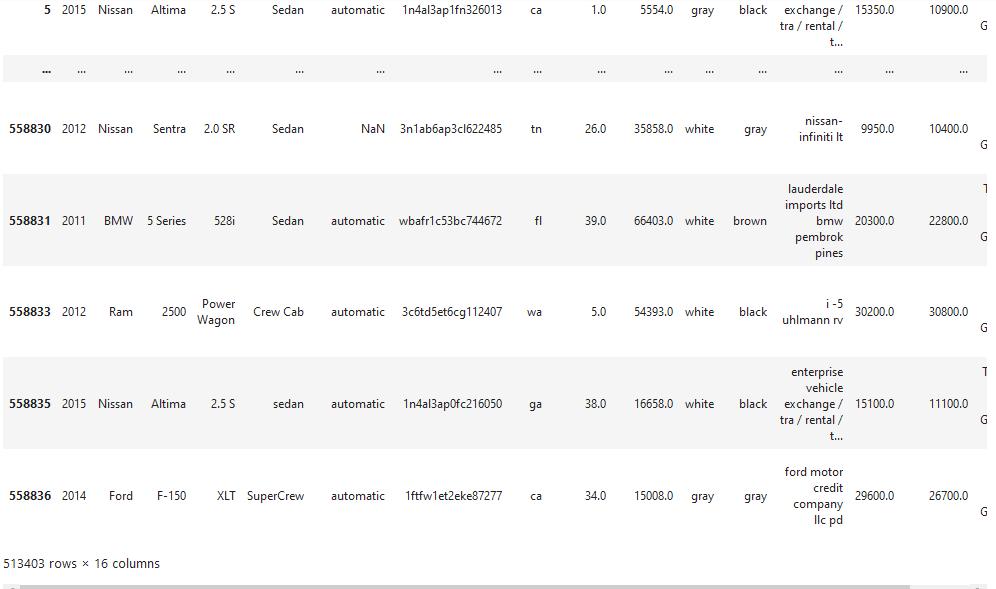
Các dòng dữ liệu còn lại

**Thuộc tính sellingprice**



Giới hạn trên và dưới của thuộc tính sellingprice

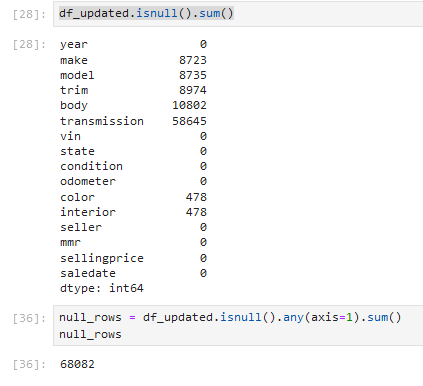
Các dòng dữ liệu ngoại lệ của thuộc tính sellingprice

Các dòng dữ liệu còn lại

### 2.2 Nhận biết các dữ liệu bị thiếu

Trong quá trình xử lý dữ liệu bị nhiễu, các giá trị ở dạng số cũng đã được xử lý nên khi tìm các dòng dữ liệu null thì chỉ còn ở trong các cột dữ liệu dạng object.

Vì vậy sử dụng hàm isnull để tìm các giá trị dạng object có chứa giá trị null sau đó là hàm sum để tính tổng các dòng tồn tại rỗng trong mỗi cột



Tính tổng số dòng null bằng axis=1 để tránh trường hợp 1 dòng có 2 giá trị null làm nhiễu dữ liệu

A screenshot of a computer

Description automatically generated

**Kết luận**: bộ dữ liệu chứa tới hơn 68000 dòng dữ liệu rỗng, chiếm gần 8% bộ dữ liệu

### 2.3 Xử lý ý nghĩa của dữ liệu

### 2.4 Xử lý dữ liệu bị thiếu

Để xử lý dữ liệu null trong các cột định tính, ta có thể thay thế điều này bằng giá trị Unknown và gom nhóm các dữ liệu rỗng vào đây cho phép ta vẫn sử dụng được các dòng dữ liệu này dù bị khiếm khuyết dữ liệu.

Kết quả là trừ đi hết các dòng dữ liệu rỗng.

A screenshot of a computer program

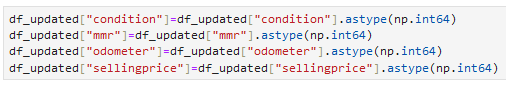
Description automatically generated

### 2.5 Kiểm tra thuộc tính của dữ liệu

A screenshot of a computer code

Description automatically generated

Nhìn nhận các cột dữ liệu ta thấy dữ liệu tồn tại ở 3 dạng là int64, float64 và object. Nhằm làm cho việc tương tác với dữ liệu đơn giản và dễ nhìn ta đổi thuộc tính về kiểu dư liệu int



Đồng thời, đối với cột saledate thì chỉ cần lưu ở dạng dd/mm/yyyy nên sẽ bỏ bớt các yếu tố giờ:phút:giây và múi giờ.  
Do hàm ép kiểu parser chỉ ứng dụng đươc cho từng chuỗi đơn nên cần chạy vòng lặp for để ép kiểu dữ liệu cho cột dữ liệu mới mang tên saledate\_formatted

A white rectangular object with black text

Description automatically generated

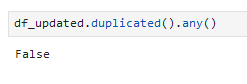
A screenshot of a computer

Description automatically generated

Kết quả cột dữ liệu mới như trên:

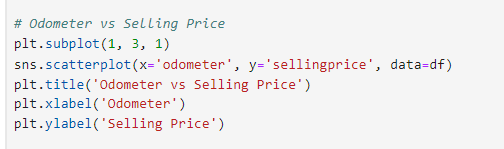
### 2.6 Xử lý các dữ liệu bị trùng lặp

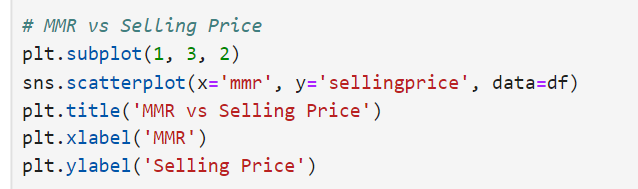
Các dòng dữ liệu trùng có thể bị phát hiện bằng hàm duplicated().any().  
Tùy thuộc vào kết trả về mà thay đổi phương pháp xử lý cho phù hợp.  
True: Xóa dữ liệu trùng bằng hàm drop\_duplicates  
False: Không có dữ liệu trùng nên tiếp tục bình thường

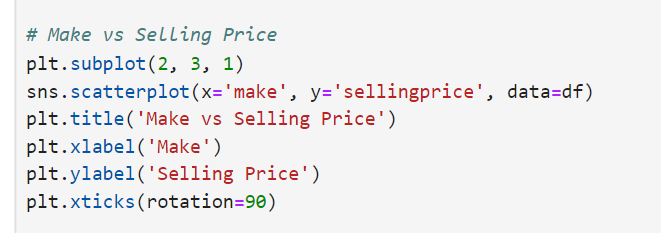


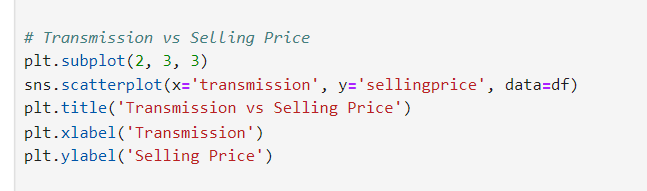
## Tích hợp dữ liệu

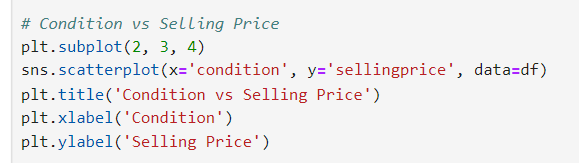
Tương quan giữa các thuộc tính và thuộc tính quyết định Sellingprice

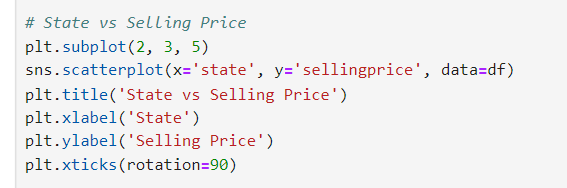


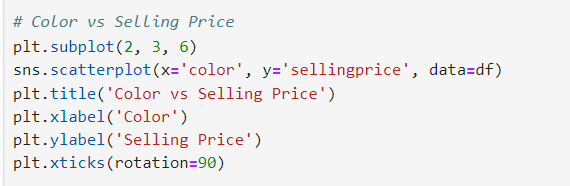




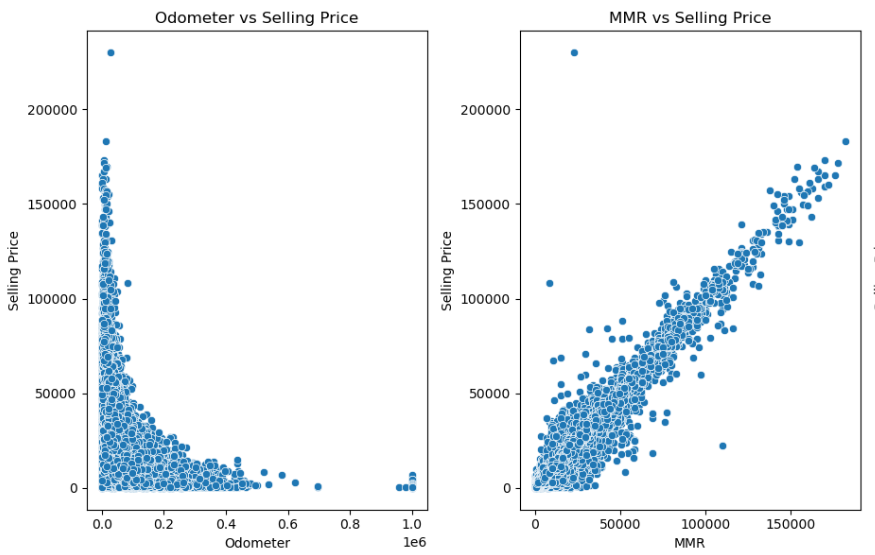




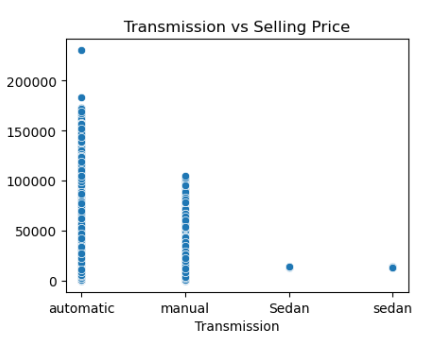


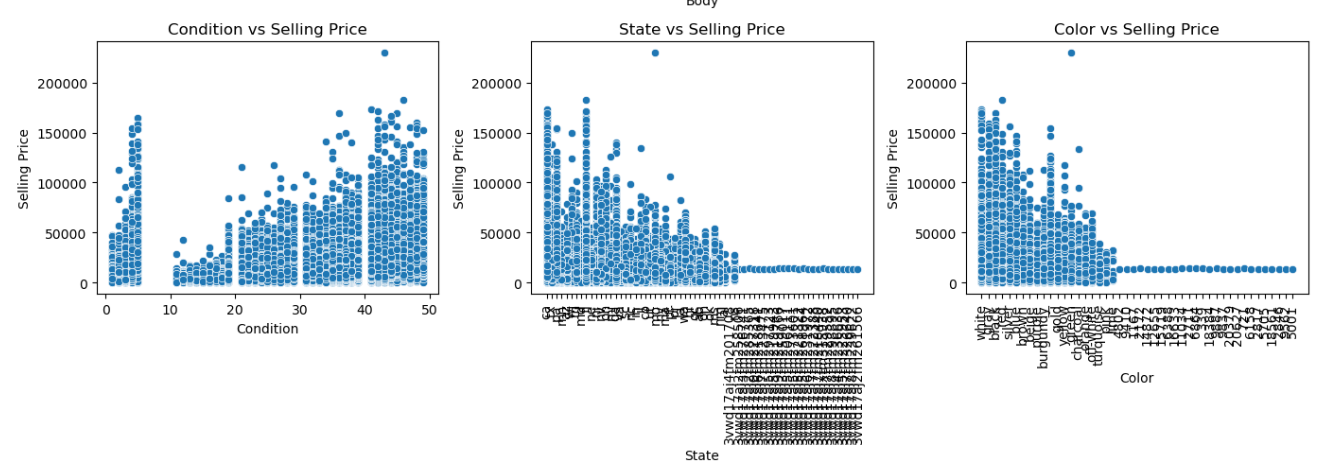


Từ đó chúng ta có các kết quả là biểu đồ scatterplot tương ứng









Nhận xét:

* Với thuộc tính odometer: Biểu đồ phân tán cho thấy một mối quan hệ nghịch giữa số km đã đi (odometer) và giá bán (sellingprice). Khi số km đã đi tăng lên, giá bán của xe có xu hướng giảm. Điều này là hợp lý vì xe có số km đã đi ít thường có giá trị cao hơn do ít hao mòn.
* Với thuộc tính mmr: Biểu đồ phân tán cho thấy một mối quan hệ tuyến tính dương giữa giá trị thị trường dự đoán (MMR) và giá bán (sellingprice). Điều này có nghĩa là MMR là một ước tính tốt cho giá bán thực tế, và giá trị thị trường cao thì giá bán cũng cao.
* Với thuộc tính make: Biểu đồ phân tán cho thấy giá bán của các xe từ các hãng khác nhau có sự biến động lớn. Một số hãng như Mercedes-Benz và BMW có xu hướng có giá bán cao hơn
* Với thuộc tính transmission: Biểu đồ phân tán cho thấy có sự khác biệt về giá bán giữa các loại hộp số. Xe hộp số tự động (automatic) có xu hướng có giá bán cao hơn so với xe hộp số tay (manual).
* Với thuộc tính condition: Biểu đồ phân tán cho thấy một mối quan hệ dương giữa tình trạng xe (condition) và giá bán (sellingprice). Xe có tình trạng tốt hơn (điểm condition cao hơn) có xu hướng có giá bán cao hơn.
* Với thuộc tính state: Biểu đồ phân tán cho thấy sự khác biệt về giá bán giữa các bang (state).
* Với thuộc tính color: Biểu đồ phân tán cho thấy màu sắc xe (color) có ảnh hưởng nhất định đến giá bán. Một số màu sắc có xu hướng có giá bán cao hơn, đứng đầu là white rồi tới gray, black

## .Biến đổi dữ liệu và thu giảm dữ liệu

### Thu giảm dữ liệu

* Xóa các thuộc tính không cần thiết:
  + Xóa thuộc tính saledate vì đã xử lý sang saledate\_formatted (chuyển sang đoạn datetime)
    - Trước:

A screenshot of a computer

Description automatically generated

* + - Sau:

A screenshot of a computer

Description automatically generated

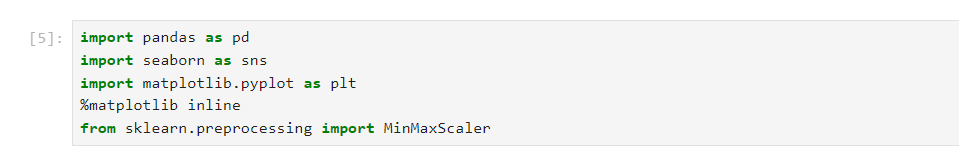
* + Xóa khoảng trắng thừa

A screenshot of a computer

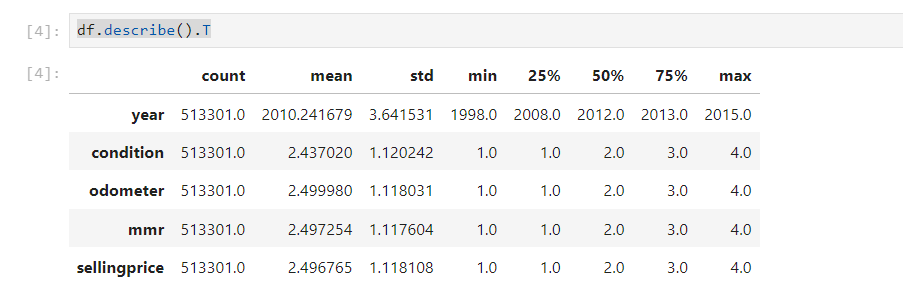
Description automatically generated

### Chuẩn hóa dữ liệu

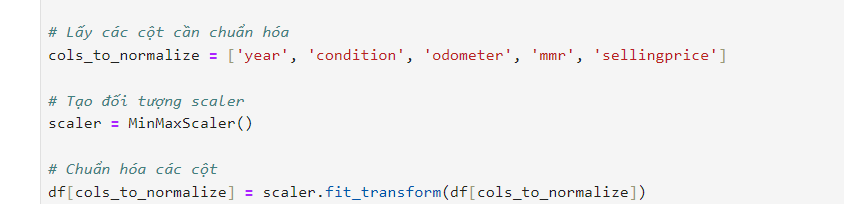
* Import MinMaxScaler từ thư viện sklearn



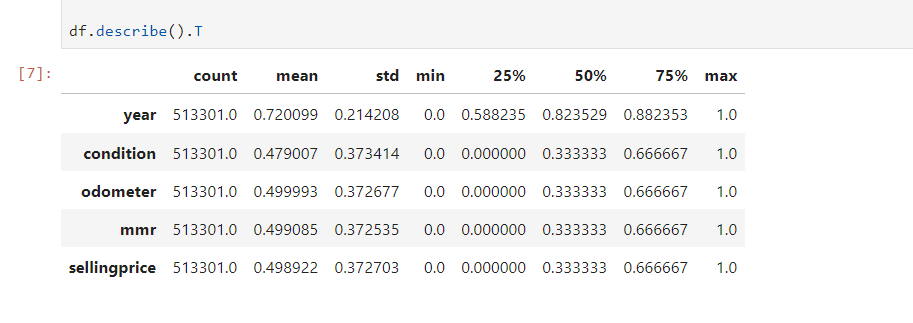
* Thống kê các thuộc tính định lượng trong dataframe



* Tiến hành chuẩn hóa dữ liệu bằng phương pháp Min\_Max Normalization



* Sau khi chuẩn hóa, độ lệch chuẩn std của các thuộc tính định lượng đã thay đổi (nhỏ hơn 1)



* Sau khi chuẩn hóa các giá trị trong DataFrame bằng phương pháp min-max normalization, ta có thể nhận thấy:
* Tất cả các cột đều có giá trị nằm trong khoảng từ 0 đến 1. Điều này cho thấy rằng phương pháp chuẩn hóa đã được áp dụng thành công.
* Các giá trị ở cột "mean" và "std" đều nằm trong khoảng từ 0 đến 1, phản ánh đúng bản chất của min-max normalization là đưa các giá trị về cùng một phạm vi.
* Các percentile như 25%, 50%, 75% cũng nằm trong khoảng từ 0 đến 1, cho thấy phân bố dữ liệu đã được chuẩn hóa.
* Giá trị min và max đều là 0 và 1 tương ứng, phù hợp với bản chất của min-max normalization.

## 5.Biểu diễn dữ liệu

Nhìn lại thông tin bộ dữ liệu sau quá trình tiền xử lý thay đổi ra sao

A screenshot of a computer code

Description automatically generated

Ta có thể thấy được rằng tổng các dòng dữ liệu của các cột đã đồng nhất nhờ xử lý dữ liệu nhiễu và rỗng. Đồng thời các cột kiểu float cũng đã bị biến đổi thành int.

A screenshot of a computer

Description automatically generated

## 6.Thu giảm số chiều dữ liệu

Nhằm tránh tình trạng dữ liệu có tính tương cao tồn tại trong bộ dữ liệu, ta dựng lên biểu đồ nhiệt hay còn gọi là correlation matrix để tìm và loại bỏ các cột có đột tương quan trên 0,5.

A red and blue squares

Description automatically generated

Kết quả thu được cho thấy thuộc tính mmr và sellingprice có mức tương cao. Vì vậy ta sẽ loại bỏ cột mmr để làm tăng hiệu suất làm việc và giảm sự thừa thải trong dữ liệu.

# Chương 4. Ứng dụng giải thuật phân lớp vào tập dữ liệu

## Cây quyết định ID3

**Khái niệm Cây quyết định ID3:**

ID3 (Iterative Dichotomiser 3) là một thuật toán học cây quyết định (decision tree) được sử dụng để xây dựng mô hình phân loại. Nó được phát triển bởi Ross Quinlan vào năm 1986 và là một trong những thuật toán cây quyết định cổ điển và quan trọng nhất.

Nguyên lý hoạt động của ID3 như sau:

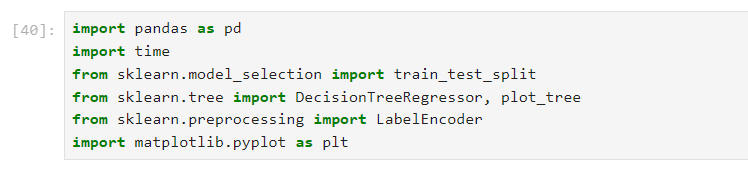
* Xây dựng cây quyết định theo hướng top-down, bắt đầu từ gốc và phân chia dần các nút con.
* Tại mỗi nút, ID3 sẽ chọn thuộc tính tốt nhất (dựa trên một tiêu chí đo lường độ lợi ích như Entropy hoặc Gain Information) để phân chia dữ liệu tại nút đó.
* Tiêu chí đo lường độ lợi ích được sử dụng là Entropy và Gain Information:
* Entropy đo lường mức độ hỗn loạn/không chắc chắn của dữ liệu.
* Gain Information đo lường lượng thông tin thu được khi sử dụng một thuộc tính để phân chia dữ liệu.
* ID3 sẽ tiếp tục chia nhánh cho đến khi đạt được các nút lá (leaf nodes) có giá trị dự đoán duy nhất.

**Ưu điểm:**

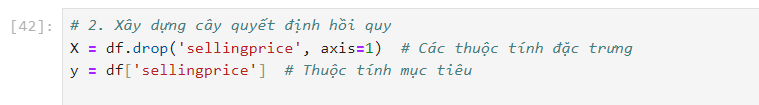
* Dễ hiểu và dễ diễn giải, tạo ra mô hình cây quyết định trực quan.
* Không yêu cầu giả định về phân phối dữ liệu.
* Hiệu quả về tính toán, đặc biệt với dữ liệu nhỏ và trung bình.
* Có thể xử lý các thuộc tính liên tục và không liên tục.
* Có khả năng xử lý dữ liệu bị thiếu và nhiễu.

**Thuật toán Decision Tree ID3:**

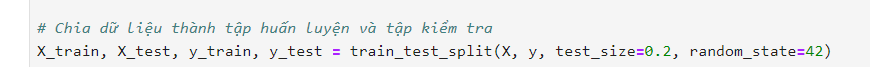
Import các thư viện cần thiết



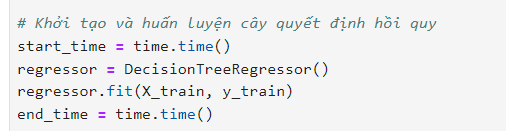
Xây dựng cây quyết định hồi quy



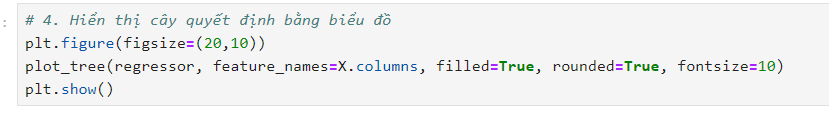
Chia dữ liệu thành tập huấn luyện và kiểm tra



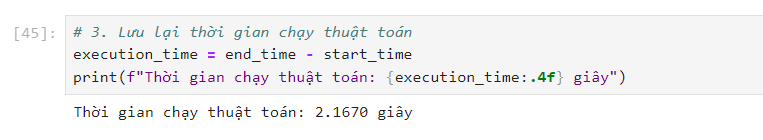
Khởi tạo và huấn luyện cây quyết định



Hiển thị cây quyết định



Thời gian chạy thuật toán



Kết quả



## KNN(K – Nearest Neighbour)

**Khái niệm KNN:**

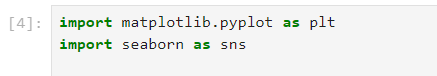
* KNN là một thuật toán học máy phổ biến được sử dụng trong các bài toán phân loại và hồi quy.
* Nó hoạt động dựa trên ý tưởng rằng các điểm dữ liệu gần nhau trong không gian đặc trưng thường có cùng nhãn hoặc giá trị.
* Để phân loại một điểm dữ liệu mới, KNN tìm k điểm dữ liệu gần nhất trong tập huấn luyện, sau đó gán nhãn hoặc giá trị dự đoán dựa trên đa số hoặc trung bình của các điểm lân cận.

**Ưu điểm**

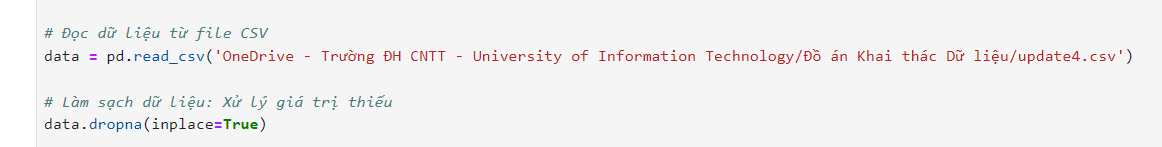
* Đơn giản và dễ hiểu: KNN là một thuật toán đơn giản và dễ hiểu, không yêu cầu quá nhiều tiền xử lý dữ liệu.
* Không yêu cầu huấn luyện mô hình: KNN không cần xây dựng một mô hình phức tạp, thay vào đó nó chỉ cần lưu trữ tập huấn luyện và tính toán khoảng cách đến các điểm dữ liệu mới.
* Hiệu quả với dữ liệu không tuyến tính: KNN hoạt động tốt với các bài toán có dữ liệu không tuân theo các giả định tuyến tính như hồi quy tuyến tính.
* Dễ dàng triển khai và áp dụng: KNN có thể được triển khai và áp dụng một cách nhanh chóng, đặc biệt là với các tập dữ liệu nhỏ hoặc trung bình.
* Không yêu cầu giả định về phân phối: KNN không yêu cầu bất kỳ giả định nào về phân phối dữ liệu, điều này làm cho nó trở thành một lựa chọn phù hợp khi không biết phân phối của dữ liệu.
* Dễ dàng giải thích: Việc xác định nhãn hoặc giá trị dự đoán của KNN dựa trên các điểm lân cận là rất dễ hiểu và giải thích.

**Thuật toán K – Nearest Neighbour:**

Import các thư viện cần thiết



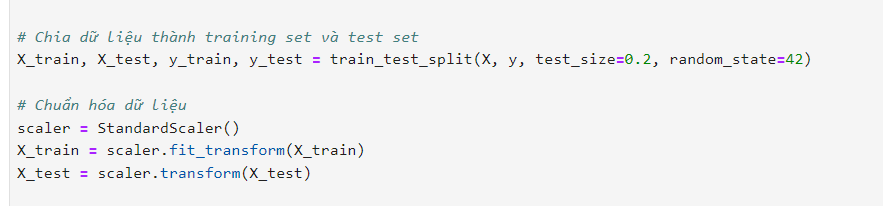
Đọc dữ liệu vào dataframe df



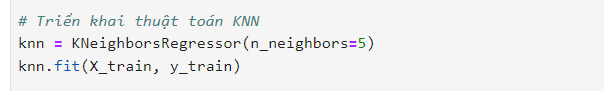
Thực hiện mã hóa các thuộc tính phân loại



Thực hiện chuẩn bị dữ liệu, chia dữ liệu và chuẩn hóa dữ liệu



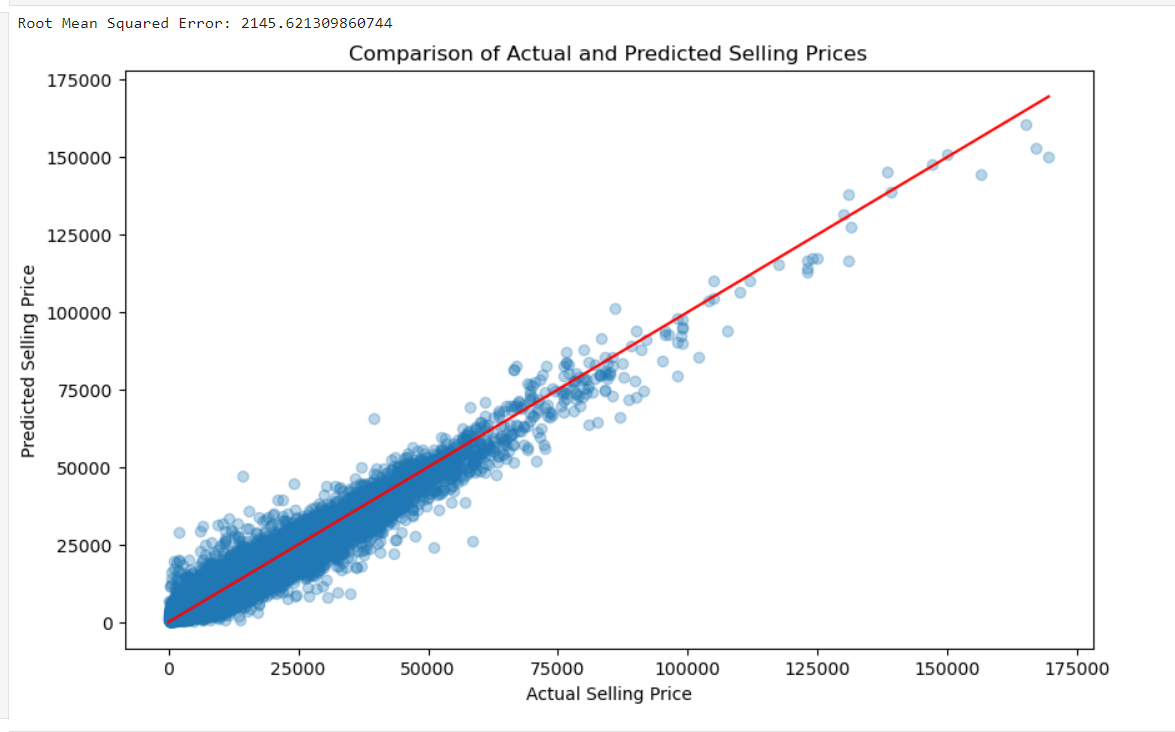
Thực hiện khởi tạo và huấn luyện knn



Dự đoán và đánh giá thuật toán



Hiển thị kết quả bằng biểu đồ scatter



Nhận xét về biểu đồ:

* Đường chéo màu đỏ: Đường chéo màu đỏ là đường 𝑦=𝑥y=x, biểu thị trường hợp lý tưởng khi giá trị dự đoán hoàn toàn trùng khớp với giá trị thực tế. Các điểm nằm trên đường này cho thấy mô hình dự đoán chính xác giá trị thực tế.
* Các điểm dữ liệu (chấm xanh): Đa số các điểm dữ liệu nằm gần đường chéo màu đỏ, cho thấy mô hình KNN dự đoán khá chính xác giá bán của các xe. Tuy nhiên một số điểm dữ liệu nằm xa đường chéo màu đỏ, cho thấy có những trường hợp mô hình dự đoán có sai số lớn.
* Phân bố các điểm dữ liệu: Phần lớn các điểm dữ liệu tập trung ở khu vực giá bán thấp (dưới 50,000), cho thấy phần lớn các xe trong tập dữ liệu có giá bán trong khoảng này. Các điểm dữ liệu ở khu vực giá bán cao (trên 100,000) thưa thớt hơn, nhưng vẫn cho thấy mô hình hoạt động khá tốt với các mức giá cao hơn.

## Naïve Bayes

**Khái niệm Naïve Bayes**

Naïve Bayes là phương pháp phân loại dựa vào xác suất được sử dụng rộng rãi trong lĩnh vực máy học và nhiều lĩnh vực khác như trong các công cụ tìm kiếm, các bộ lọc mail. Mục đích chính là làm sao tính được xác xuất Pr(Cj,d’), xác suất để tài liệu d’ nằm trong lớp Cj.

**Ưu điểm**

* + Dễ dàng áp dụng và dễ hiểu
  + Hiệu suất tốt với không gian đặc trưng và dữ liệu lớn.
  + Hữu dụng trong bài toán phân loại văn bản

**Thuật toán Naïve Bayes**

* + Chạy thuật toán bằng những dòng lệnh sau:

A screen shot of a computer program

Description automatically generated

A computer screen with colorful text

Description automatically generated

A computer screen shot of a program

Description automatically generated

* + Kết quả thời gian huấn luyện, thời gian dự đoán và độ chính xác của mô hình

A screenshot of a computer screen

Description automatically generated

## 4.Support Vector Machine

**Khái niệm:**

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồi thị dữ liệu là các điểm trong n chiều ( ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (Hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

**Ưu điểm:**

Đây là thuật toán hoạt động hiệu quả với không gian cao chiều (high dimensional spaces). Thuật toán tiêu tốn ít bộ nhớ vì chỉ sử dụng các điểm trong tập hỗ trợ để dự báo trong hàm quyết định. Chúng ta có thể tạo ra nhiều hàm quyết định từ những hàm kernel khác nhau.

**Thuật toán Support Vector Machine (SVM)**

Chạy thuật toán bằng các dòng lệnh sau:

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer program

Description automatically generated

Kết quả thời gian huấn luyện, thời gian dự đoán và độ chính xác:

A screenshot of a computer

Description automatically generated

## 5.Random Forest Model

**Khái niệm:** Mô hình được huấn luyện dựa trên sự phối hợp giữa luật kết hợp (ensembling*)* và quá trình lấy mẫu tái lặp(boostrapping). Cụ thể thuật toán này tạo ra **nhiều cây quyết định** mà mỗi cây quyết định được huấn luyện dựa trên **nhiều mẫu con** khác nhau và kết quả dự báo là bầu cử (voting) từ toàn bộ những **cây quyết định.**

**Ưu điểm:**

Random Forest cho thấy hiệu quả hơn so với thuật toán phân loại thường được sử dụng vì có khả năng tìm ra thuộc tính nào quan trọng hơn so với những thuộc tính khác.

Trên thực tế, nó còn có thể chỉ ra rằng một số thuộc tính là không có tác dụng trong cây quyết định

khắc phục được hiện tượng quá khớp (overfitting) hay gặp ở cây quyết định

**Thuật toán Random Forest:**

* + Chạy thuật toán bằng những dòng lệnh sau:

**A screenshot of a computer program

Description automatically generated**

Kết quả dự báo

**A white rectangular object with black numbers

Description automatically generated**

Xây dựng biểu đồ cột tầm quan trọng của thuộc tính được sử dụng trong thuật toán

**A graph of blue rectangular bars with white text

Description automatically generated**

Trực quan hóa kết quả thuật toán

**A pair of white objects

Description automatically generated with medium confidence**

## 6.Bagging Regressor

**Khái niệm:**

Bagging (Bootstrap Aggregating) là một phương pháp Ensemble Learning (học tập kết hợp) được sử dụng trong Machine Learning để nâng cao độ chính xác của mô hình dự đoán. Trong Bagging, chúng ta tạo ra nhiều mô hình dự đoán độc lập với nhau từ các tập dữ liệu con được lấy mẫu ngẫu nhiên với sự thay thế từ tập dữ liệu huấn luyện ban đầu. Sau đó, chúng ta kết hợp các dự đoán của các mô hình này để đưa ra dự đoán cuối cùng

**Ưu điểm:**

Giảm phương sai bằng cách đào tạo máy học trên các tập hợp con dữ liệu khác nhau, bagging mang lại sự đa dạng giữa các mô hình. Khi các mô hình đa dạng này được kết hợp, các lỗi sẽ bị loại bỏ, dẫn đến các dự đoán ổn định và đáng tin cậy hơn.

Đóng bao giúp chống lại việc quá khớp bằng cách giảm phương sai của mô hình. Bằng cách tạo ra nhiều tập hợp con của dữ liệu huấn luyện đảm bảo tập trung vào các khía cạnh hơi khác nhau của dữ liệu. Sự đa dạng này giúp nhóm khái quát hóa dữ liệu chưa nhìn thấy tốt hơn.

Bagging huấn luyện nhiều mô hình trên các tập con dữ liệu khác nhau nên nó có xu hướng ít nhạy cảm hơn với các điểm ngoại lệ và điểm dữ liệu nhiễu. Các ngoại lệ ít có khả năng tác động đến dự đoán tổng thể khi nhiều mô hình được kết hợp đáng kể.

**Thuật toán Bagging Regressor**

* + Chạy thuật toán bằng những dòng lệnh sau:

**A screenshot of a computer program

Description automatically generated**

**A graph of a graph

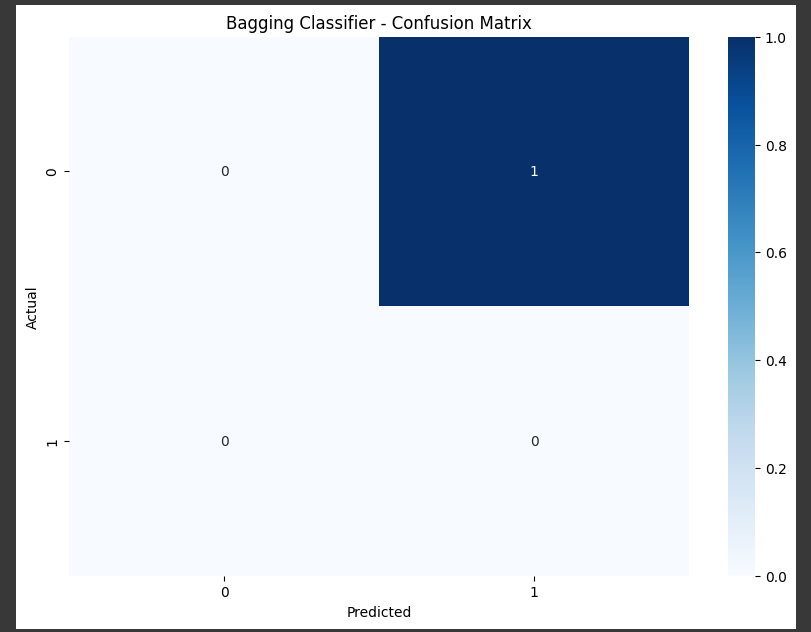
Description automatically generated with medium confidence**

Từ biểu đó trên kết luận được rằng:

* Model Bagging có khả năng học từ dữ liệu huấn luyện, như được thể hiện qua điểm số huấn luyện cao trên 0.85
* Điểm số xác thực tăng dần khi kích thước tập huấn luyện tăng lên, cho thấy mô hình có thể tổng quát hóa tốt với tập xác thực. Điều này chứng tỏ rằng mô hình không bị quá khớp với dữ liệu huấn luyện.
* Sự giảm nhẹ trong điểm số huấn luyện và tăng trong điểm số xác thực cho thấy mô hình cân bằng tốt giữa độ lệch và phương sai. Mô hình không bị ảnh hưởng nhiều bởi tình trạng thiếu khớp hoặc quá khớp đáng kể.

# Chương 5. Phân tích đánh giá các thuật toán và dự báo

## Ma trận nhầm nhẫn



A blue square with white text

Description automatically generated

A blue square with white text

Description automatically generated

A blue square with white text

Description automatically generated

A blue square with white text

Description automatically generated

A blue square with white text

Description automatically generated

## Đánh giá về thời gian chạy thuật toán

A graph with blue and white bars

Description automatically generated with medium confidence

* Thời gian chạy thuật toán Random Forest là lớn nhất.
* Hai thuật toán SVM và Naïve Bayes có thời gian chạy gần như = 0.

## Đánh giá về độ chính xác của thuật toán

A graph of a graph

Description automatically generated

* Thuật toán Bagging Regressor và Random forest thường có độ chính xác cao nhất với chỉ số MAE thấp, cho thấy rằng chúng dự đoán rất gần với giá trị thực tế.
* Thuật toán Decision Tree và KNN có thể có MAE cao hơn một chút, chỉ ra rằng chúng có thể ít chính xác hơn trong một số trường hợp.
* Thuật toán SVM và Naïve Bayes cũng hoạt động tốt, nhưng hiệu suất của SVM phụ thuộc nhiều vào việc chọn đúng tham số, trong khi Bayesian Ridge mang lại sự ổn định trong dự đoán.

Khi so sánh các mô hình này, chỉ số MAE thấp hơn đồng nghĩa với độ chính xác cao hơn, do đó, lựa chọn mô hình có MAE thấp nhất là lựa chọn tốt nhất cho mục tiêu dự đoán chính xác.

## Đánh giá thuật toán

### So sánh thời gian chạy

A graph with numbers and a bar

Description automatically generated

Ta thấy được rằng thuật toán SVM tốn nhiều thời gian nhất. Điều này do rằng thuật toán không phù hợp với bộ dữ liệu quy mô lớn vì biến đổi chiều không gian O(n^2) và thời gian O(n^3) [1]

### 2.So sánh độ chính xác của thuật toán

Cây quyết định ID3



Knn:



Random Forest

A number of numbers and symbols

Description automatically generated with medium confidence

SVM:



Naïve Bayes



Bagging Regressor:



Ta thấy rằng thuậ toán Bagging Regressor và Random Forest đem lại độ chính xác cao nhất nhờ vào độ phức tạc của dữ liệu cũng như đặt tính tập kết hợp và chống quá khớp của thuật toán.

# Phân chia công việc

|  |  |  |  |
| --- | --- | --- | --- |
|  | Phùng Thiên Phúc - 21521297 | Nguyễn Quang Lâm - 21521057 | Phạm Mạnh Hùng - 21520901 |
| Chọn bộ dữ liệu và ý tưởng đề tài | Hỗ trợ | Hỗ trợ | X |
| Chương 1: Nhận diện bài toán | Hỗ trợ | Hỗ trợ | X |
| Chương 2: Trực quan hóa dữ liệu | Hỗ trợ | X | X |
| Chương 3: Tiền xử lý dữ liệu |  | | |
| Mô tả dữ liệu, Làm sạch dữ liệu, Rời rạc hóa dữ liệu, Biểu diễn dữ liệu, Thu giảm số chiều dữ liệu | X | Hỗ trợ | Hỗ trợ |
| Tích hợp dữ liệu | Hỗ trợ | Hỗ trợ | X |
| Biến đổi dữ liệu và thu giảm dữ liệu | Hỗ trợ | X | Hỗ trợ |
| Chương 4. Ứng dụng giải thuật phân lớp vào tập dữ liệu |  | | |
| Cây quyết định ID3 và KNN | Hỗ trợ | Hỗ trợ | X |
| Naïve Bayes và Support Vector Machine | Hỗ trợ | X | Hỗ trợ |
| Random Forest Model và Bagging | X | Hỗ trợ | Hỗ trợ |
| Chương 5. Phân tích đánh giá các thuật toán và dự báo |  | | |
| Ma trận nhầm lẫn và đánh giá tỉ lệ TPR và FPR | Hỗ trợ | X | Hỗ trợ |
| Đánh giá thời gian chạy và độc chính xác của thuật toán | Hỗ trợ | Hỗ trợ | X |
| Đánh giá thuật toán | X | Hỗ trợ | Hỗ trợ |

# TÀI LIỆU THAM KHẢO

[1] <https://typeset.io/papers/fast-svm-training-using-edge-detection-on-very-large-4xo9qvfp6l>