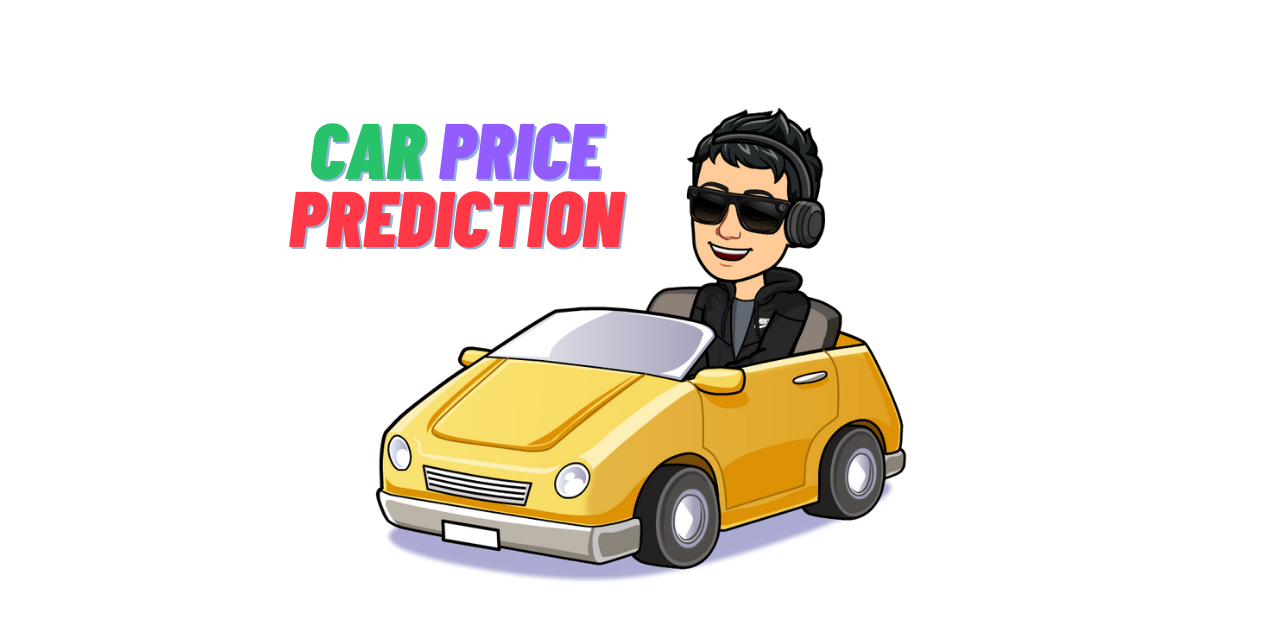
**TRƯỜNG ĐẠI HỌC BÁCH KHOA KHOA CÔNG NGHỆ THÔNG TIN**

BÁO CÁO CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU



DỰ ĐOÁN GIÁ XE HƠI

**GVHD: Phạm Công Thắng**

|  |  |  |  |
| --- | --- | --- | --- |
| HỌ VÀ TÊN SINH VIÊN | LỚP HỌC PHẦN | MÃ SINH VIÊN | |
| Huỳnh Đinh Hoàng Viên | 20Nh13 | 102200160 |
| Đặng Văn Nhật Minh | 20Nh13 | 102200139 |
| Trần Công Nguyên Hải | 20Nh13 | 102200127 |
| Phạm Văn Nhật Nam | 20Nh13 | 102200140 |

ĐÀ NẴNG, 05/2023

**TÓM TẮT**

Bộ dữ liệu bao gồm các thông tin, thông số kỹ thuật tổng quát, giá bán của xe hơi, được thu thập từ trang web bán các xe hơi ở Việt Nam. Bộ dữ liệu này được dùng để dự đoán giá xe hơi với các thông số kỹ thuật, mẫu mã được đưa ra. Dữ liệu sau khi thu thập là các chuỗi ký tự trong đó có các đại lượng cần dùng cho mô hình dự đoán. Khi làm sạch dữ liệu, cần trả về đúng kiểu dữ liệu, thay thế các giá trị dữ liệu trống và chọn lựa ra các đặc trưng phù hợp cho mô hình dự đoán. Mô hình dự đoán được xây dựng bởi thuật toán Random Forest Regression và cải tiến độ chính xác bằng mô hình XGBoost Regression. Đánh giá hiệu quả của mô hình dự đoán sử dụng RMSE, MAE và R2.

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **Các nhiệm vụ** | **Tự đánh giá theo 3 mức** |
| Huỳnh Đinh Hoàng Viên | * Thu thập dữ liệu * Làm sạch dữ liệu | * Đã hoàn thành * Đã hoàn thành |
| Đặng Văn Nhật Minh | * Làm sạch dữ liệu, xử lý dữ liệu trống * Chuẩn hóa dữ liệu | * Đã hoàn thành * Đã hoàn thành |
| Phạm Văn Nhật Nam | * Trực quan hóa dữ liệu * Mô ta dữ liệu | * Đã hoàn thành * Đã hoàn thành |
| Trần Công Nguyên Hải | * Xây dựng mô hình * Đánh giá thuật toán | * Đã hoàn thành * Đã hoàn thành |

**MỤC LỤC**

Mục lục

[1. Giới thiệu 6](#_Toc135735140)

[2. Thu thập và mô tả dữ liệu 6](#_Toc135735141)

[2.1. Thu thập dữ liệu 6](#_Toc135735142)

[2.1.1 Nguồn dữ liệu 6](#_Toc135735143)

[2.1.2 Công cụ thu thập 6](#_Toc135735144)

[2.1.3 Cách thức sử dụng công cụ 7](#_Toc135735145)

[2.1.4 Format & làm sạch dữ liệu 11](#_Toc135735146)

[2.2. Mô tả dữ liệu 12](#_Toc135735147)

[3. Trích xuất đặc trưng 13](#_Toc135735148)

[3.1. Loại bỏ các cột dữ liệu không cần thiết, thêm cột dùng cho dự đoán 13](#_Toc135735149)

[3.2. Làm sạch dữ liệu với đúng kiểu dữ liệu 14](#_Toc135735150)

[3.3. Xử lý dữ liệu trống 15](#_Toc135735151)

[3.4. Chuyển các dữ liệu phân loại thành dữ liệu dạng số (Encode Categorical) 15](#_Toc135735152)

[3.5. Chia dữ liệu thành tập input X, nhãn Y 15](#_Toc135735153)

[3.6. Chuẩn hóa dữ liệu 16](#_Toc135735154)

[3.7. Tương quan giữa các cột thuộc tính của dữ liệu 17](#_Toc135735155)

[4. Mô hình hóa dữ liệu 18](#_Toc135735156)

[4.1. Mô hình Random Forest Regression 18](#_Toc135735157)

[4.2. Mô hình Hồi quy của thư viện XGBoost - XGBoost Regression 21](#_Toc135735158)

[4.3. So sánh hiệu quả của các mô hình 22](#_Toc135735159)

[5. Dự đoán và kết luận 23](#_Toc135735160)

[5.1. Dự đoán giá một chiếc xe hơi bất kì 23](#_Toc135735161)

[5.2. Mô hình hóa dữ liệu 25](#_Toc135735162)

[6. Tài liệu tham khảo 25](#_Toc135735163)

**DANH SÁCH HÌNH ẢNH**

[Hình 1. Mô hình Web Scrapping sử dụng Beautiful Soup 6](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734866)

[Hình 2. Đường dẫn tới các trang chứa danh sách xe ô tô (300 trang) 7](#_Toc135734867)

[Hình 3. Dang sách sản phẩm theo từng trang 7](#_Toc135734868)

[Hình 4. Phân tích cấu trúc của website 8](#_Toc135734869)

[Hình 5. Tìm đường link chứa url dẫn tới trang chi tiết 8](#_Toc135734870)

[Hình 6. Lấy dữ liệu thông tin chi tiết của sản phẩm 9](#_Toc135734871)

[Hình 7. Lấy các class 9](#_Toc135734872)

[Hình 8. Trích xuất thông tin lưu vào file .csv 10](#_Toc135734873)

[Hình 9. Source code 10](#_Toc135734874)

[Hình 10. Format dữ liệu động cơ 11](#_Toc135734875)

[Hình 11. Format thông tin tên hãng 11](#_Toc135734876)

[Hình 12. Format tên xe 11](#_Toc135734877)

[Hình 13. Format đơn vị tiền 11](#_Toc135734878)

[Hình 14. Bảng mô tả các cột dữ liệu 12](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734879)

[Hình 15. Thông tin về các cột giá trị trong dữ liệu 12](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734880)

[Hình 16. Bảng phân bố số lượng xe theo giá tiền 13](#_Toc135734881)

[Hình 17. Dữ liệu sau khi thêm bớt các cột 13](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734882)

[Hình 18. Chuyển kiểu dữ liệu của cột “Giá tiền” 14](#_Toc135734883)

[Hình 19. Dữ liệu sau khi làm sạch 14](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734884)

[Hình 20. Dữ liệu sau khi chuyển các cột Object thành số bằng LabelEncoder của thư viện sk-learn 15](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734885)

[Hình 21. Khởi tạo tập dữ liệu X và nhãn Y 16](#_Toc135734886)

[Hình 22. Chia dữ liệu thành X\_train, Y\_train, X\_test, Y\_test 16](#_Toc135734887)

[Hình 23. Chuẩn hóa dữ liệu tập X\_train 17](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734888)

[Hình 24. Ma trận tương quan giữa các thuộc tính nhận dạng 18](#_Toc135734889)

[Hình 25. Sử dụng thuật toán RandomForest để xây dựng mô hình dự đoán giá 19](#_Toc135734890)

[Hình 26. Tương quan giữa tập test và predict khi sử dụng thuật toán RandomForest 19](#_Toc135734891)

[Hình 27. Khởi tạo bộ tham số cho mô hình XGBoost 21](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734892)

[Hình 28. Tương quan giữa tập test và predict khi sử dụng mô hình XGBoost 22](#_Toc135734893)

[Hình 29. Load file pkl 23](file:///C:\Users\ASUS-FPT\Downloads\BaoCaoKHDL.docx#_Toc135734894)

[Hình 30. Tạo một đối tượng xe hơi mới 23](#_Toc135734895)

[Hình 31. Sử dụng lớp OrdinalEncoder 24](#_Toc135734896)

[Hình 32. Sử dụng lớp StandardScaler 24](#_Toc135734897)

[Hình 33. Hàm dự đoán giá 24](#_Toc135734898)

[Hình 34. Kết quả dự đoán 24](#_Toc135734899)

**NỘI DUNG BÁO CÁO**

# Giới thiệu

Mô hình dự đoán giá xe hơi dựa trên các thông số kỹ thuật tổng quát, đặc trưng của xe hơi sẽ giúp các hãng xe ước tính được giá bán xe hơi phù hợp để cạnh tranh với các hãng khác. Ngoài ra, mô hình này cũng giúp cho khách hàng xác định được giá tốt nhất cho xe hơi họ muốn mua với các thông số cụ thể.

# Thu thập và mô tả dữ liệu

### Thu thập dữ liệu

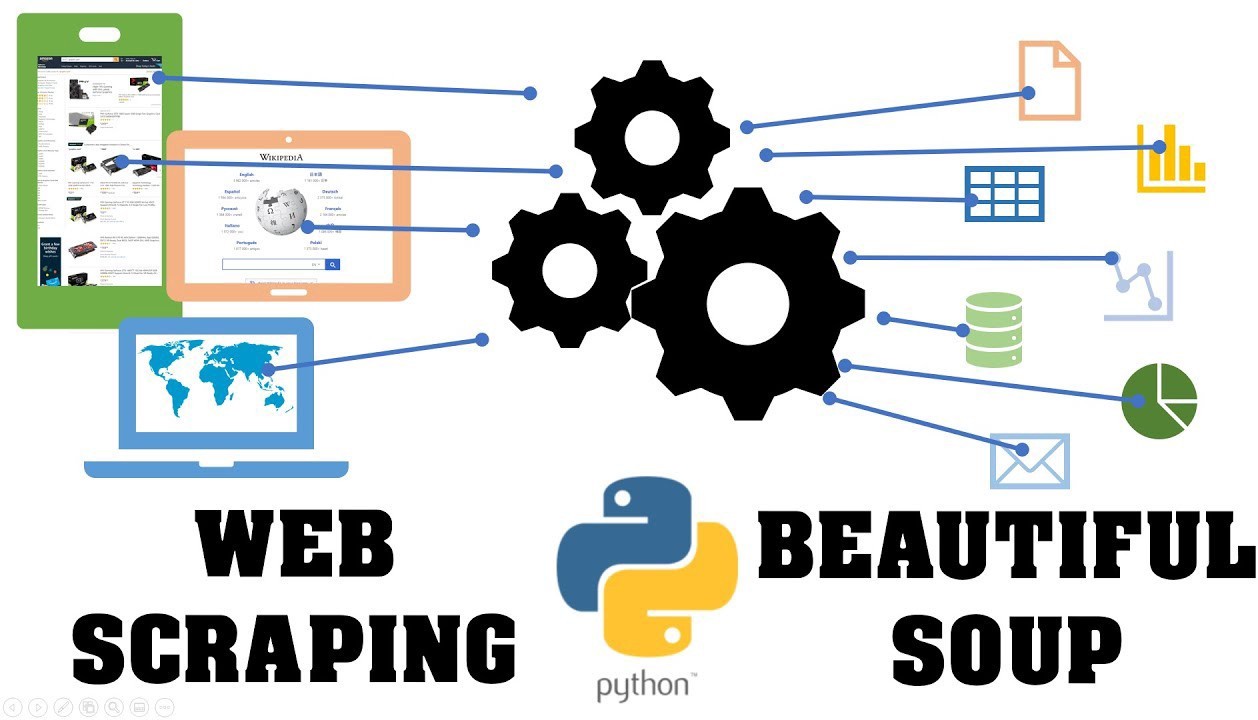
### Nguồn dữ liệu

Dữ liệu được thu thập từ trang web bán xe hơi lớn ở Việt Nam, các xe được lựa chọn có mức giá < 1 tỉ 600 triệu:

https://bonbanh.com/oto-xe-moi-gia-duoi-1600-trieu/page

### Công cụ thu thập

Thu thập dữ liệu từ web sử dụng thư viện **Beautiful Soup**. Đây là một thư viện Python dùng để chuyển đổi HTML và XML. Nó tạo ra một cây phân tích của trang web dùng để trích xuất dữ liệu từ HTML giúp ích cho việc cào dữ liệu.

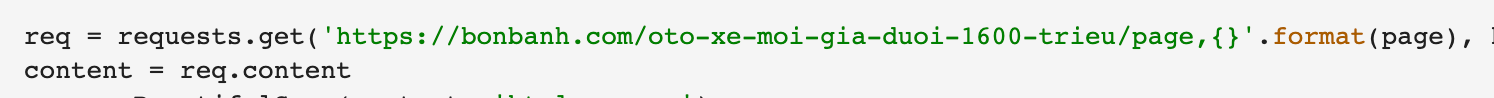


Hình 1. Mô hình Web Scrapping sử dụng Beautiful Soup

### Cách thức sử dụng công cụ

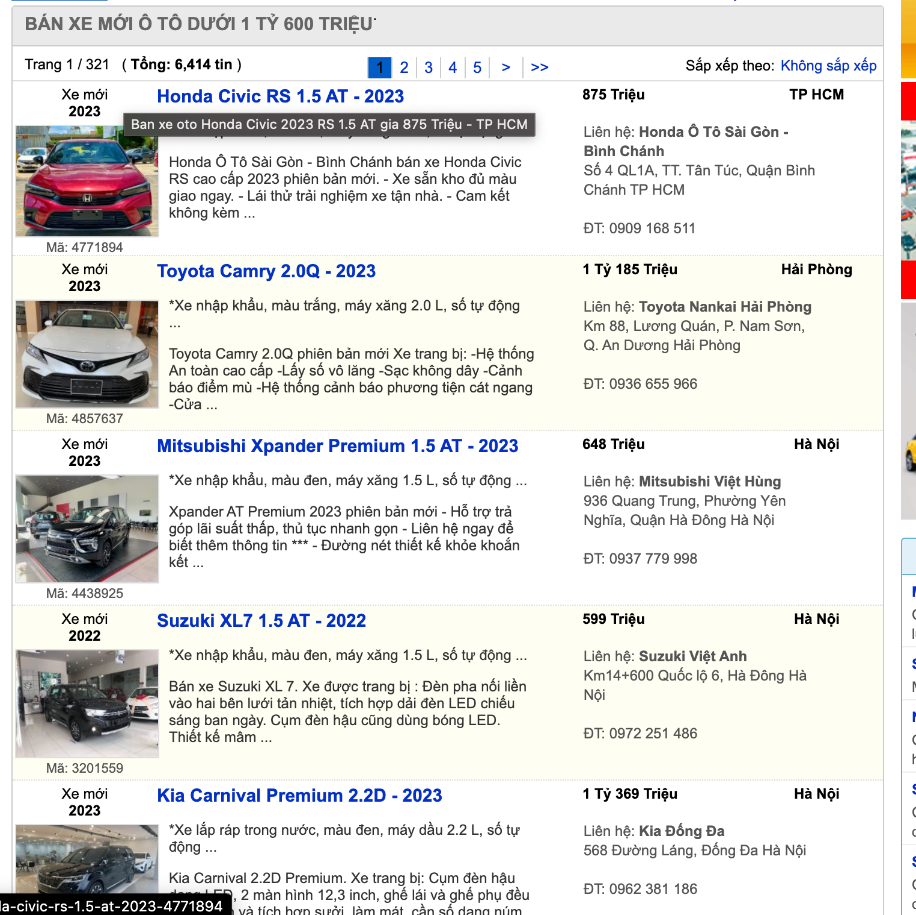
* Duyệt qua 300 trang danh sách các sản phẩm
* Lặp qua từng dữ liệu sản phẩm để lấy các thông tin tổng quan
* Lấy ra link chi tiết sản phẩm -> Tiến hành phân tích trang sản phẩm chi tiết để lấy các dữ liệu cần thiết
* Format làm sạch dữ liệu
* Kết quả: File chứa dữ liệu car-data.csv

**Bước 1**: Lấy ra tất cả link của các sản phẩm



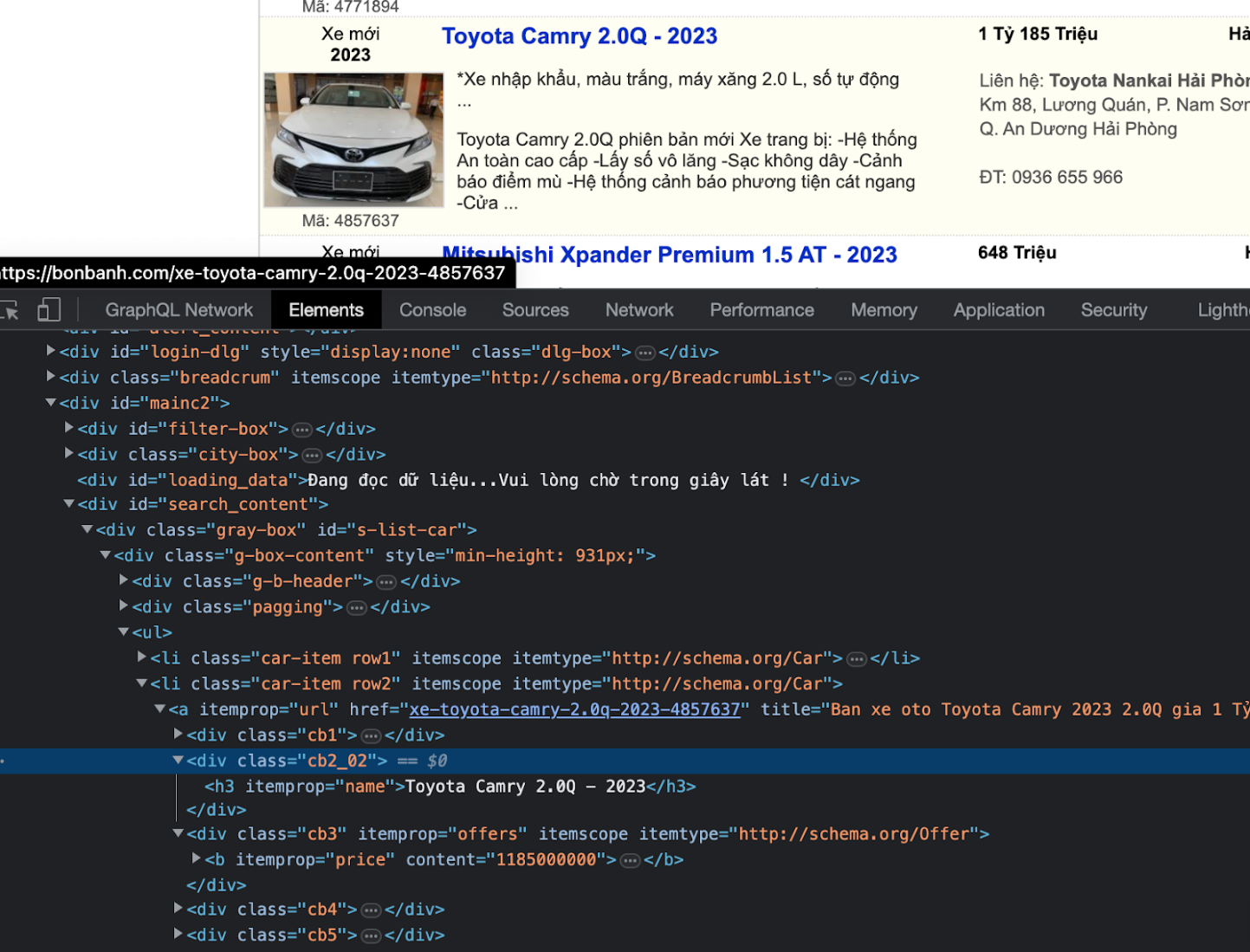
Hình 2. Đường dẫn tới các trang chứa danh sách xe ô tô (300 trang)

Sau khi kiểm tra thấy được trang web gọi đến một api.



Hình 3. Dang sách sản phẩm theo từng trang

**Bước 2**: Phân tích cấu trúc trang web của từng sản phẩm



Hình 4. Phân tích cấu trúc của website

Tìm chứa thông tin về thông số kỹ thuật cơ bản của xe

Tìm kiếm class và id của các thẻ

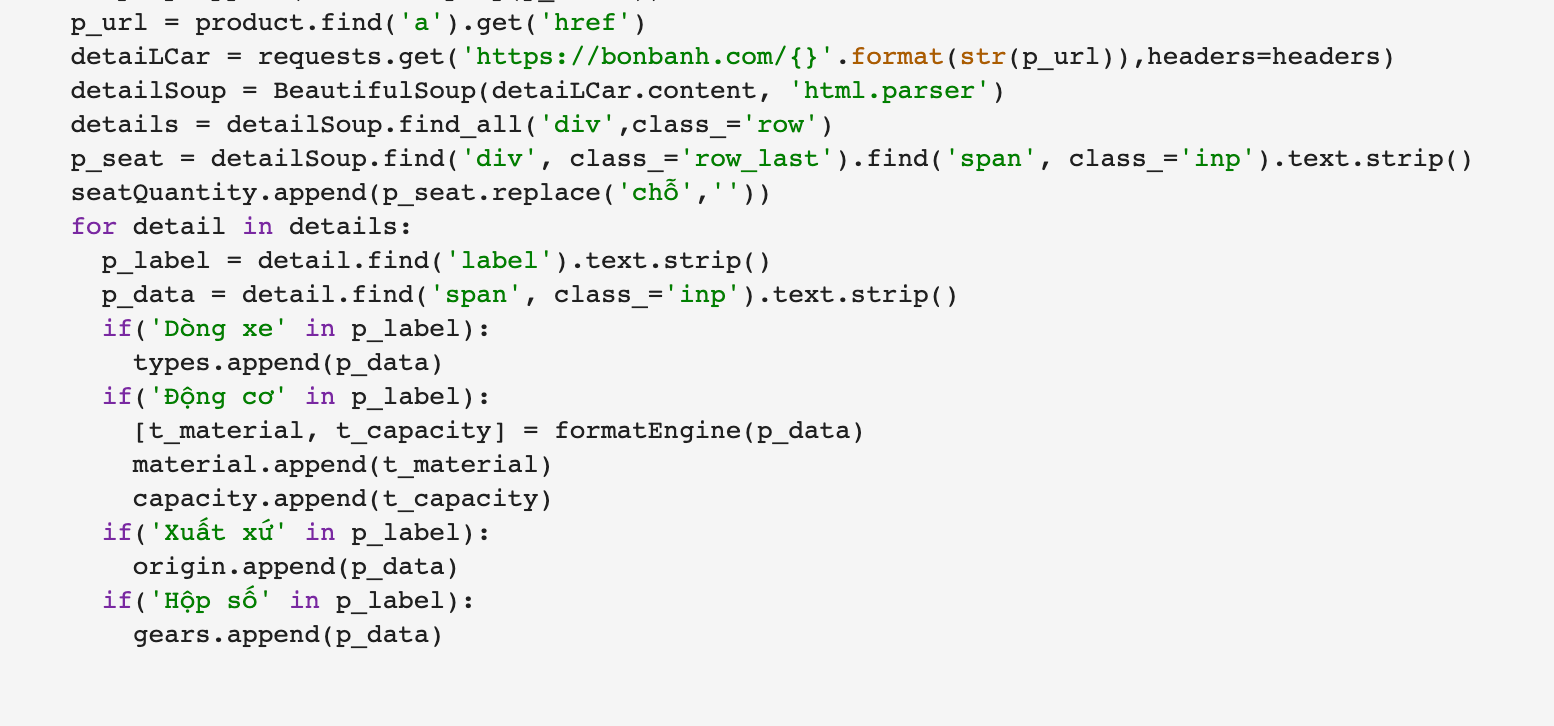
Sau khi đã lấy được các thông số kỹ thuật. Tìm thẻ chứa link chuyển sang trang chi tiết sản phẩm.

A screen shot of a computer

Description automatically generated with medium confidence

Hình 5. Tìm đường link chứa url dẫn tới trang chi tiết

Lấy dữ liệu các thông tin chi tiết của sản phẩm

****

Hình 6. Lấy dữ liệu thông tin chi tiết của sản phẩm

**A screenshot of a computer

Description automatically generated with medium confidence**

Hình 7. Lấy các class

**Bước 3**: Trích xuất thông tin và lưu vào file csv

A screenshot of a computer program

Description automatically generated with low confidence

Hình 8. Trích xuất thông tin lưu vào file .csv

**Source code:**

**A screenshot of a computer program

Description automatically generated with low confidence**

Hình 9. Source code

### Format & làm sạch dữ liệu

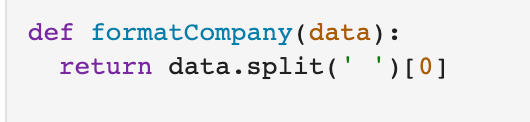
* Format thông tin động cơ: ví dụ Xăng 2.0L -> Xăng vào 2.0

A screen shot of a computer code

Description automatically generated with low confidence

Hình 10. Format dữ liệu động cơ

* Format thông tin tên hãng:



Hình 11. Format thông tin tên hãng

* Format tên xe:

A picture containing text, font, screenshot, white

Description automatically generated

Hình 12. Format tên xe

* Format đơn vị tiền:

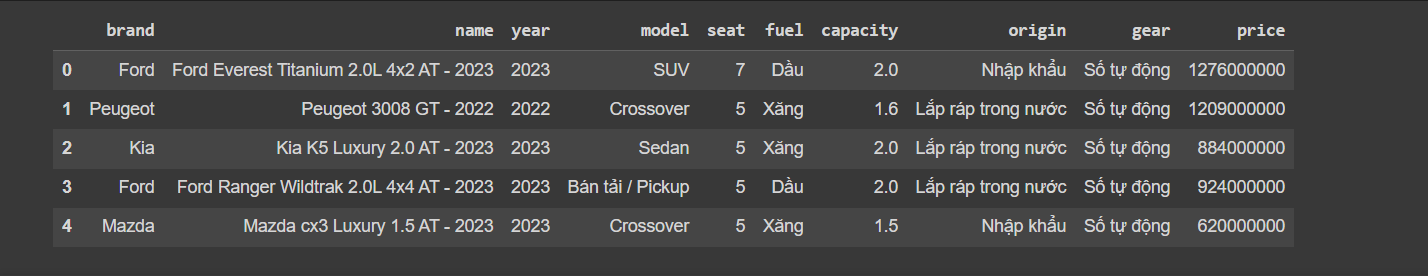
A picture containing text, screenshot, font

Description automatically generated

Hình 13. Format đơn vị tiền

### Mô tả dữ liệu

Dữ liệu được crawl từ website bao gồm 6000 hàng và 10 cột bao gồm:

* Hãng xe: hãng sản xuất (ví dụ: Ford, Toyota, Hyundai,…)
* Tên xe: tên đầy đủ xe (ví dụ: Ford Everest Titanium 2.0L 4x2AT,..)
* Năm sản xuất: năm sản xuất của xe
* Dòng xe (ví dụ: SUV, Crossover,…)
* Số chỗ ngồi
* Nhiên liệu
* Dung tích xi lanh
* Xuất sứ
* Hộp số

Hình 14. Bảng mô tả các cột dữ liệu

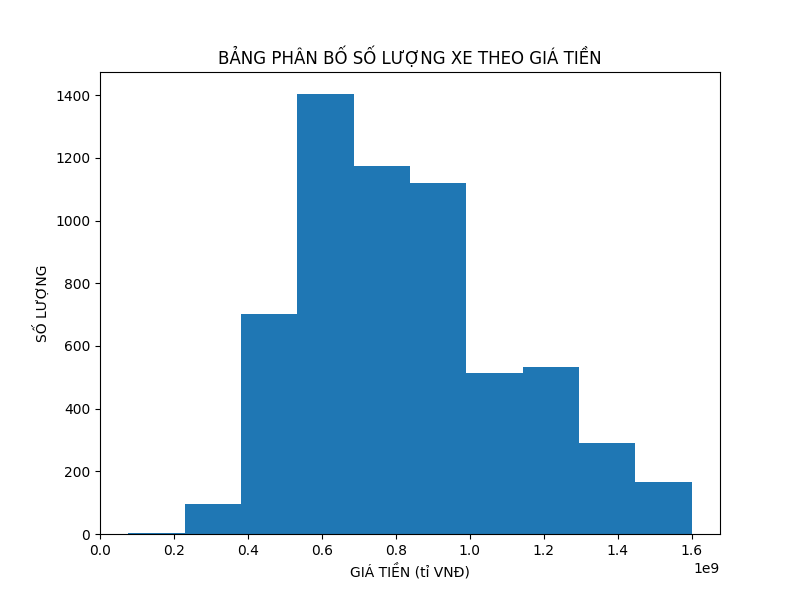
**A screenshot of a computer

Description automatically generatedĐánh giá về số các ô giá trị trống trong dữ liệu:**

Hình 15. Thông tin về các cột giá trị trong dữ liệu

* Các cột dữ liệu đều có đủ dữ liệu (ở cột non-null thì có đủ 6000 giá trị) => crawl dữ liệu rất tốt
* Có nhiều cột dữ liệu kiểu Object, điều đó gây khó khăn ở việc chuẩn hóa dữ liệu

**Phân bố dữ liệu của cột dự đoán “price”**



Hình 16. Bảng phân bố số lượng xe theo giá tiền

Đa phần giá xe tập trung ở phân khúc giá từ 400 triệu - 1 tỉ 200 triệu.

Có một số giá trị khá nhỏ (<200 triệu) nhưng không đáng kể.

# Trích xuất đặc trưng

### Loại bỏ các cột dữ liệu không cần thiết, thêm cột dùng cho dự đoán

Toàn bộ dữ liệu thô bao gồm tổng cộng **6000 hàng** với **10 cột**. Trong đó, có các cột không cần thiết cho mô hình dự đoán giá, ta sẽ loại bỏ các cột này. Đó là cột “Unnamed: 0” – đây là cột được tạo ra tự động khi chúng ta load file “.csv”.

Do các thông số tổng quát của xe hơi đều khá tương đồng với mỗi chiếc xe, nhưng cùng một thông số thì giá tiền giữa các hãng xe lại có sự chênh lệch. Do đó chúng ta tách từ cột “Tên xe” thành một cột mới là “Hãng xe”.

A screenshot of a computer

Description automatically generated with low confidenceCuối cùng ta đổi tên các cột dữ liệu để thuận tiện trong quá trình chuẩn hóa dữ liệu.

Hình 17. Dữ liệu sau khi thêm bớt các cột

### Làm sạch dữ liệu với đúng kiểu dữ liệu

Bộ dữ liệu bao gồm 10 cột với 7 dữ liệu dạng đối tượng và 3 dữ liệu dạng số.

1. **Dữ liệu đối tượng**

Các dữ liệu dạng đối tượng:

* Tên xe
* Dòng xe
* Nhiên liệu
* Xuất sứ
* Hộp số
* Giá xe

Vì các thông số bao quát của xe chưa phản ánh hết giá của xe nên chúng ta tách từ cột “Tên xe” thêm một cột “Hãng xe”.

Cột “Dung tích” ta loại bỏ ký tự đơn vị “L”.

Với cột “Giá xe” định dạng ban đầu ở web là “X tỉ Y triệu”, do đó trong quá trình clean chúng ta cần xử lý bằng cách viết hàm để format thành dạng XXX,XXX,XXX,XXX.

1. **Dữ liệu dạng số**

Đối với các dữ liệu dạng số, sau khi xử lý cột “Giá tiền” ở phần crawl thì chúng ta nhận được cột với kiểu dữ liệu Object, để thuận tiện cho quá trình xử lý khi training data thì chúng ta cần chuyển nó sang kiểu số.



Hình 18. Chuyển kiểu dữ liệu của cột “Giá tiền”

A screenshot of a computer

Description automatically generated with low confidenceDữ liệu sau khi làm sạch:

Hình 19. Dữ liệu sau khi làm sạch

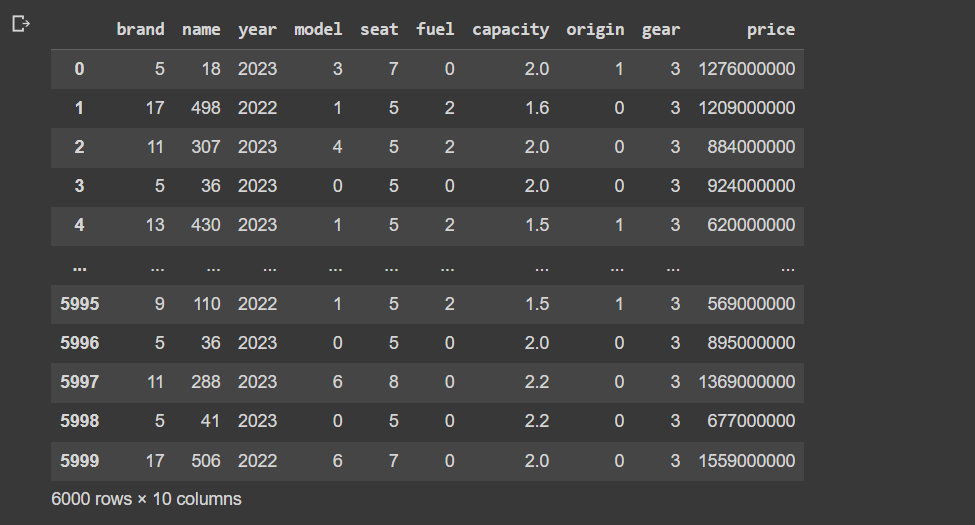
### Xử lý dữ liệu trống

Như đã trình bày ở phần 2.2, do dữ liệu khi crawl về không có giá trị trống nên chúng ta cũng sẽ bỏ qua bước xử lý này.

### Chuyển các dữ liệu phân loại thành dữ liệu dạng số (Encode Categorical)

Sử dụng *LabelEncoder* của thư viện *sklearn* để chuyển các kiểu dữ liệu Object thành dữ liệu số cho mô hình máy học có thể hiểu.

Chuyển các cột dữ liệu ‘brand’*, ‘name’, ‘model’, ‘fuel’, ‘origin’, ‘gear’* thành dạng số.

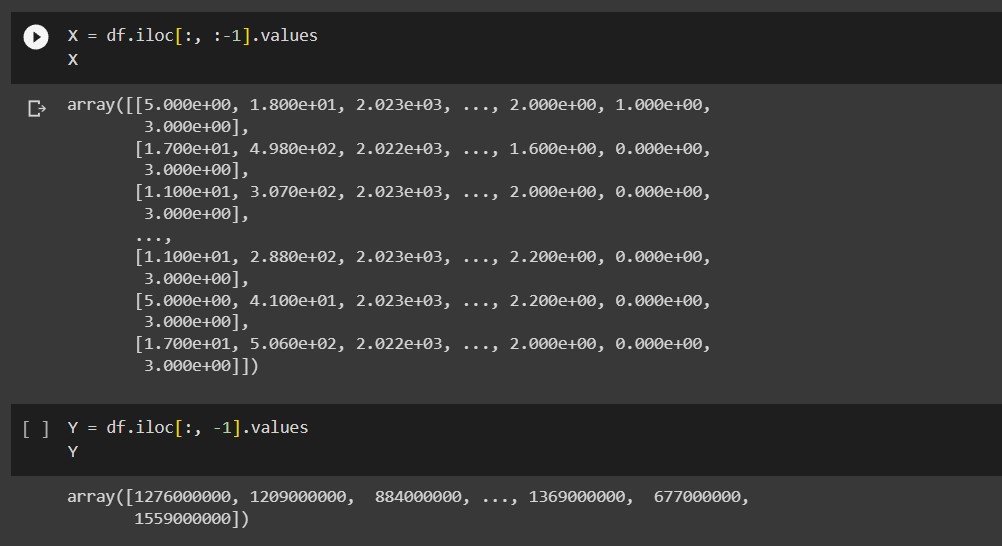
Dữ liệu sau khi chuyển thành dạng số:

Hình 20. Dữ liệu sau khi chuyển các cột Object thành số bằng LabelEncoder của thư viện sk-learn

### Chia dữ liệu thành tập input X, nhãn Y

Khi thực hiện xử lý với mô hình máy học, thông thường chúng ta cần chuyển các dữ liệu sau khi đã Label – Encode thành dạng numpy-array

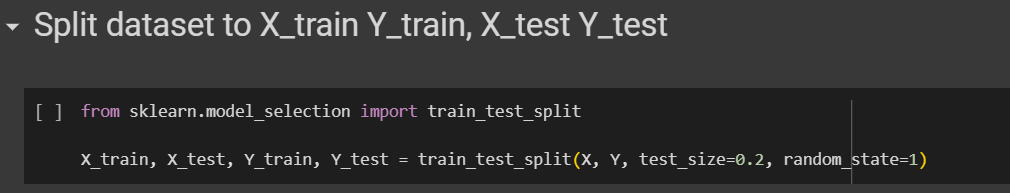
Chúng ta lấy các cột đầu tiên làm tập input X và tập dữ liệu giá làm output Y



Hình 21. Khởi tạo tập dữ liệu X và nhãn Y

Sau khi có dữ liệu X và nhãn Y, chúng ta tiếp tục chia 2 tập dữ liệu thành dữ liệu train và test để đưa vào mô hình học máy. Sở dĩ chúng ta cần chia thành tập train và test để tránh tình trạng overfitting model và giúp chúng ta xây dựng được các tham số dùng cho đánh giá mô hình.

Thực hiện chia X và Y thành tập train và test với tỉ lệ 4:1 (80% train và 20% test)



Hình 22. Chia dữ liệu thành X\_train, Y\_train, X\_test, Y\_test

### Chuẩn hóa dữ liệu

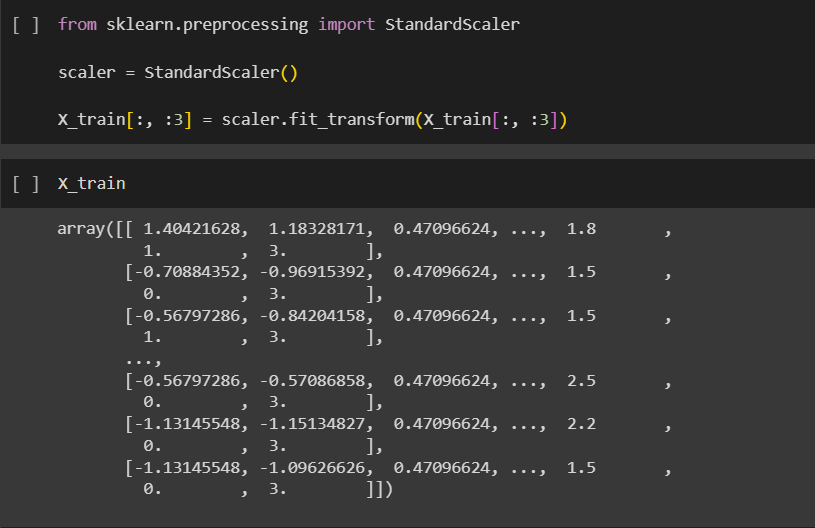
Tiếp theo trong quá trình chuẩn bị dữ liệu để đưa vào mô hình máy học là công đoạn chuẩn hóa dữ liệu. Quan sát các giá trị số sau khi Label-Encode, ta dễ dàng nhận thấy sự chênh lệch các giá trị giữa các cột giá trị, điều này dễ dẫn đến sai khác trong quá trình train data, làm giảm hiệu suất của mô hình, do đó ta cần chuẩn hóa các giá trị này về cùng một miền dữ liệu.

Sử dụng lớp StandardScaler của lớp scikit – learn để đưa các giá trị về phân phối chuẩn, phương pháp này được thực hiện theo công thức:

X\_scaler = (X – mean) / standard\_deviation

Trong đó: X là giá trị ban đầu

Mean là giá trị trung bình của tập dữ liệu

 Standard\_deviation là độ lệch chuẩn của tập giá trị

Hình 23. Chuẩn hóa dữ liệu tập X\_train

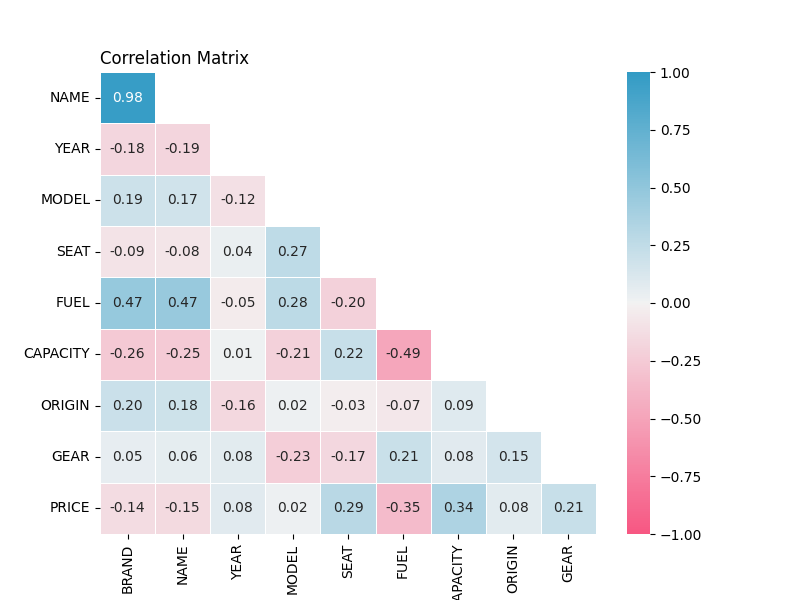
Thực hiện tương tự với tập X\_test.

### 3.7. Tương quan giữa các cột thuộc tính của dữ liệu

Việc xem xét sự tương quan giữa các thuộc tính dự đoán giá xe hơi giúp ta có thể chọn lựa các thuộc tính có độ tương quan cao với giá để đưa vào dự đoán.

Ở đây do có khá ít cột dữ liệu (10 cột) và các dữ liệu đa phần đều chỉ mang tính khái quát về một chiếc xe nên không có sự tương quan cao với giá xe.

Trong một số trường hợp, độ tương quan cũng không hoàn toàn thể hiện tính phù hợp của một thuộc tính trong bài toán dự đoán. Trong trường hợp này, mặc dù độ tương quan thấp nhưng chúng ta nhưng có thể mang lại vài thông tin hữu ích, ví dụ như giá xe sẽ phụ thuộc phần lớn vào hãng sản xuất.



Hình 24. Ma trận tương quan giữa các thuộc tính nhận dạng

# Mô hình hóa dữ liệu

Để đáp ứng cho yêu cầu bài toán Dự đoán giá xe hơi, nhóm đã chọn ra 2 mô hình cho để giải quyết bài toán này: Hồi quy tuyến tính (Random Forest Regression) và Hồi quy của thư viện XGBoost (XGBoost Regression).

### Mô hình Random Forest Regression

**Cơ sở lý thuyết**: Random Forest là một thuật toán học máy dựa trên việc xây dựng một tập hợp các cây quyết định độc lập. Mỗi cây quyết định được xây dựng dựa trên một tập con dữ liệu ngẫu nhiên và không tương tác với nhau trong quá trình dự đoán. Kết quả dự đoán cuối cùng được tính toán bằng cách kết hợp kết quả của tất cả các cây quyết định. Random Forest có khả năng xử lý dữ liệu rộng, chống overfitting, xử lý dữ liệu thiếu và xếp hạng đặc trưng, là một trong những thuật toán học máy phổ biến và mạnh mẽ.

Trong thuật toán Decision Tree (cây quyết định), khi xây dựng cây quyết định nếu để độ sâu tùy ý thì cây sẽ phân loại đúng hết các dữ liệu trong tập training dẫn đến mô hình có thể dự đoán tệ trên tập validation/test, khi đó mô hình bị overfitting, hay nói cách khác là mô hình có [high variance](https://viblo.asia/p/the-bias-variance-decomposition-eW65Gm3YZDO) (thể hiện tốt với dữ liệu có sẵn nhưng kém hiệu quả với dữ liệu mới hoàn toàn).

Thuật toán Random Forest gồm nhiều cây quyết định, mỗi cây quyết định đều có những yếu tố ngẫu nhiên:

* Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định
* Lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định

**Bộ tham số của mô hình**: RandomForestRegressor(n\_estimators, random\_state)

Trong đó:

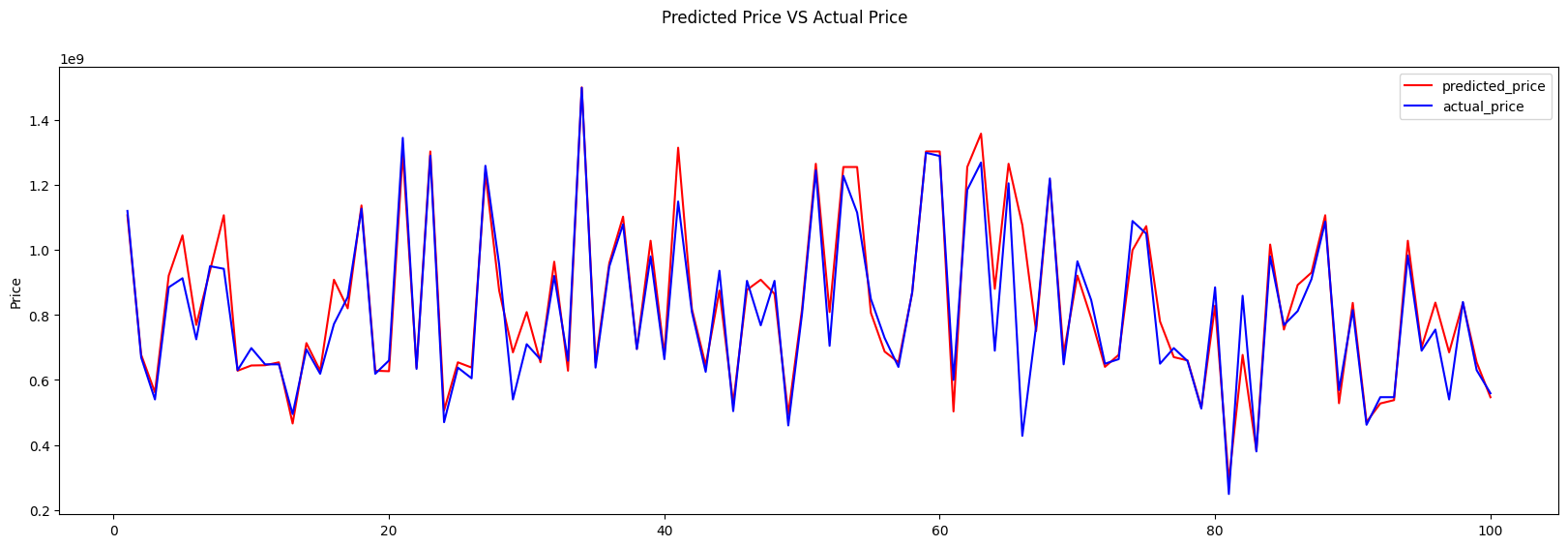
* Estimators: số lượng cây quyết định được xây dựng trong random forest, với estimators càng lớn càng tăng hiệu suất của mô hình, nhưng cũng tăng thời gian huấn luyện
* Random\_state: bằng cách đặt random\_state thành một giá trị cố định, chúng ta đảm bảo rằng mỗi lần chạy mô hình, chúng ta sẽ nhận được kết quả nhất quán.

A screenshot of a computer

Description automatically generated with medium confidence

Hình 25. Sử dụng thuật toán RandomForest để xây dựng mô hình dự đoán giá

**Biểu đồ tương quan giữa tập test và predict:**

****

Hình 26. Tương quan giữa tập test và predict khi sử dụng thuật toán RandomForest

Chú thích: Biểu đồ biểu diễn kết quả chênh lệch giữa giá dự đoán và giá thực tế của từng mẫu (lấy kết quả của 100 mẫu đầu tiên trong tập kiểm thử), đường màu đỏ là giá dự đoán, đường màu xanh là giá thực tế.

**Kết quả**: Nhìn trên đồ thị ta thấy đường màu đỏ và màu xanh chênh lệch không nhiều, dao động trong khoảng 20 đến 100 triệu. Kết quả dự đoán đánh giá trên hàm score có sẵn của mô hình của mô hình có độ chính xác theo tiêu chuẩn đánh giá R2(%) đạt **85.31%** trên tập dữ liệu kiểm thử.

**Đánh giá**: Độ hiệu quả dự đoán với các metrics, các metrics sử dụng để đánh giá bao gồm: *RMSE*, *MAE* và *R2*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RMSE**  *(nghìn đồng)* | **MAE**  *(nghìn đồng)* | **R2**  *(%)* |
| **Kết quả** | 105,769,763.10 | 57,789,856.07 | 85.31 |

*Chú thích:*

* RMSE: Root Mean Square Error là phép đo dùng để đo lường độ lớn sai số dự đoán giữa giá trị dự đoán của mô hình và giá trị thực tế trong dữ liệu.  
  RMSE đo lường độ lớn trung bình của sai số dự đoán theo đơn vị của biến phụ thuộc. Điều này có nghĩa là giá trị RMSE càng nhỏ, mô hình càng có khả năng dự đoán chính xác và gần với giá trị thực tế.
* MAE: Mean Absolute Error là phép đo dùng để đo lường trung bình giá trị tuyệt đối của sai số dự đoán giữa giá trị dự đoán của mô hình và giá trị thực tế trong dữ liệu.  
  MAE đo lường độ lớn trung bình của sai số dự đoán dựa trên đơn vị tuyệt đối của biến phụ thuộc. Giá trị MAE càng nhỏ, mô hình càng có khả năng dự đoán chính xác và gần với giá trị thực tế.  
  MAE thường được sử dụng trong các bài toán hồi quy và có ưu điểm là nhạy cảm với các giá trị ngoại lệ (outliers), vì nó không bình phương các sai số như RMSE. Tuy nhiên, MAE không nhận biết được hướng của sai số (dương hay âm), chỉ xem xét độ lớn tuyệt đối của sai số.
* R2: R-squared là phép đo dùng để đo lường tỉ lệ phương sai của biến phụ thuộc mà mô hình có thể giải thích được so với tổng phương sai của biến thụ thuộc

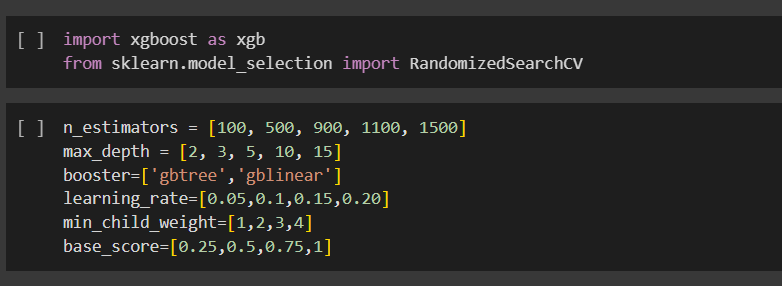
R2 càng gần 1 thì mô hình càng tốt trong việc dự đoán biến phụ thuộc  
R2 càng gần 0 thì mô hình càng không tốt trong việc dự đoán biến phụ thuộc  
Tuy nhiên R2 không đánh giá toàn diện được mọi khía cạnh của mô hình, do đó cần kết hợp với các phép đo khác để thực hiện đánh giá

### Mô hình Hồi quy của thư viện XGBoost - XGBoost Regression

**Cơ sở lý thuyết:** XGBoost (Extreme Gradient Boosting) là một giải thuật được base trên gradient boosting, tuy nhiên kèm theo đó là những cải tiến to lớn về mặt tối ưu thuật toán, về sự kết hợp hoàn hảo giữa sức mạnh phần mềm và phần cứng, giúp đạt được những kết quả vượt trội cả về thời gian training cũng như bộ nhớ sử dụng. Có nhiều các mô hình cho việc Hồi quy (Regression), Phân lớp (Classification),...

**Bộ tham số của mô hình:** Sử dụng RandomizedSearchCV để tìm siêu tham số cho mô

|  |  |  |
| --- | --- | --- |
| Tên tham số | Mảng giá trị | Ý nghĩa |
| n\_estimators | [100, 500, 900, 1100, 1500] | Số estimators trong mô hình tìm kiếm |
| max\_depth | [2, 3, 5, 10, 15] | Độ sâu tìm kiếm tối đa |
| booster | [‘gbtree’, ‘gblinear’] | Mô hình để chạy |
| learning\_rate | [0.05, 0.1, 0.15, 0.20] | Tốc độ học |
| min\_child\_weight | [1,2,3,4] | Trọng lượng tối thiểu của giá trị con |
| base\_scrore | [0.25, 0.5, 0.75, 1] | Điểm khởi tạo tìm tham số |

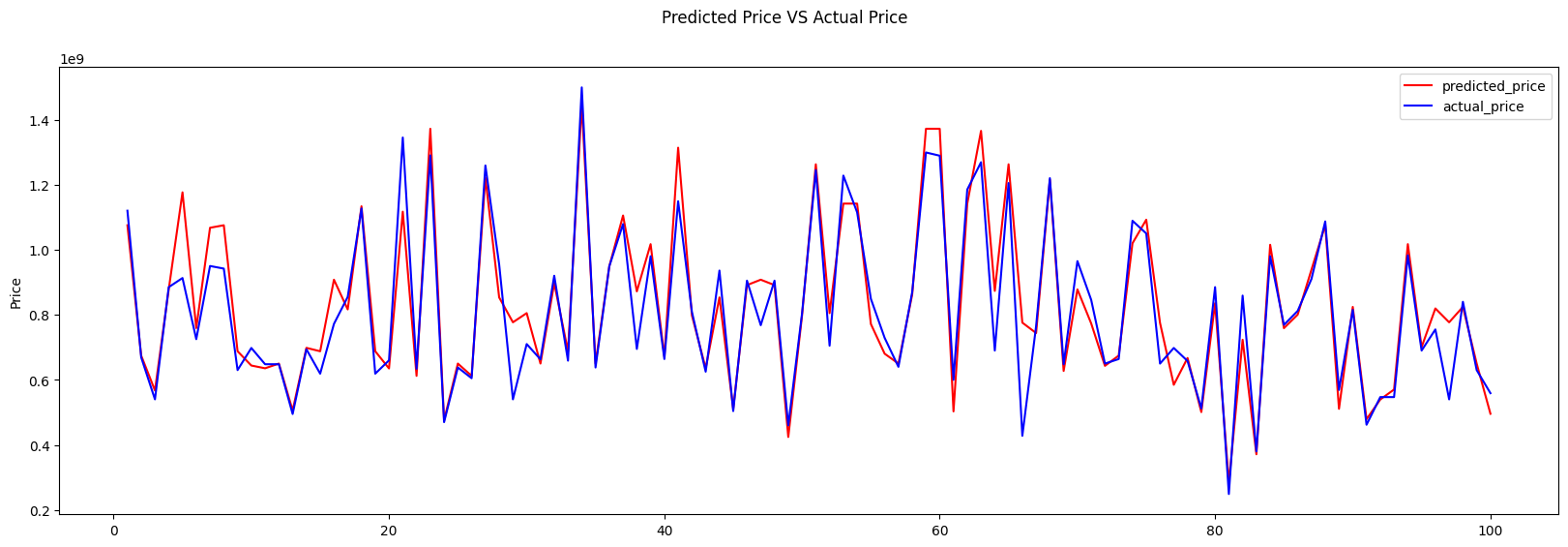


Hình 27. Khởi tạo bộ tham số cho mô hình XGBoost

Các tham số sử dụng trong quá trình huấn luyện mô hình: Sau khi sử dụng RandomizedSearchCV để tìm siêu tham số cho mô hình, các tham số tìm được như sau:

|  |  |  |
| --- | --- | --- |
| **Tên tham số** | **Giá trị** | **Ý nghĩa** |
| n\_estimators | 1100 | Số estimators trong tìm kiếm |
| max\_depth | 5 | Độ sâu tìm kiếm tối đa |
| booster | gbtree | Mô hình để chạy |
| learning\_rate | 0.1 | Tốc độ học |
| min\_child\_weight | 2 | Trọng lượng tối thiểu của giá trị con |
| base\_scrore | 0.75 | Mức điểm làm mốc tìm tham số |

**Biểu đồ tương quan giữa tập test và predict**

****

Hình 28. Tương quan giữa tập test và predict khi sử dụng mô hình XGBoost

**Đánh giá:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RMSE**  *(nghìn đồng)* | **MAE**  *(nghìn đồng)* | **R2**  *(%)* |
| **Kết quả** | 98,721,948.65 | 61,073,155.8 | 87.2 |

### So sánh hiệu quả của các mô hình

|  |  |  |  |
| --- | --- | --- | --- |
|  | **RMSE** *(nghìn đồng)* | **MAE** *(nghìn đồng)* | **R2***(%)* |
| **RandomForest Regression** | 105,769,763.10 | 57,789,856.07 | 85.31 |
| **XGBoost Regression** | 98,721,948.65 | 61,073,155.8 | 87.2 |

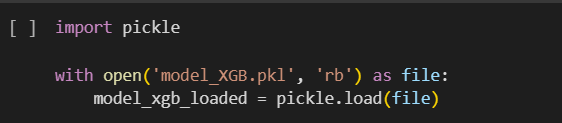
**Đánh giá:**

* Thông số MAE của thuật toán RandomForest Regression nhỏ hơn XGBooots Regression, điều đó cho thấy độ sai lệch các giá trị của thuật toán RandomForest Regression ít hơn
* Hiệu suất R2 của XGBoost lại cao hơn một ít và RMSE thấp hơn cho thấy mô hình XGBoost dự đoán tốt hơn.
* Điều này xảy ra có thể do chọn các thông số của XGBoost chưa đạt hiệu quả cao nhất.
* Trong thực tế tùy vào dữ liệu và mục đích sử dụng có thể chọn một trong 2 mô hình trên.

# Dự đoán và kết luận

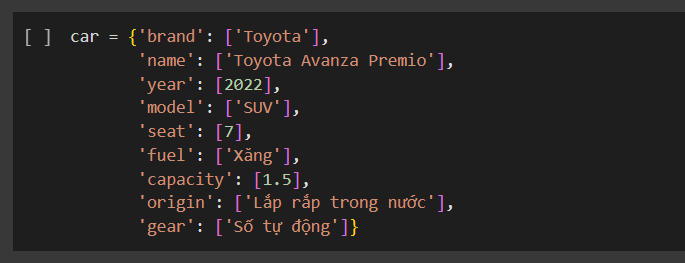
### Dự đoán giá một chiếc xe hơi bất kì

Ở phần này, chúng ta sẽ sử dụng mô hình sau khi lưu dưới dạng file .pkl để tiến hành áp dụng dự đoán giá trong thực tế

**Load file pkl:**

Hình 29. Load file pkl

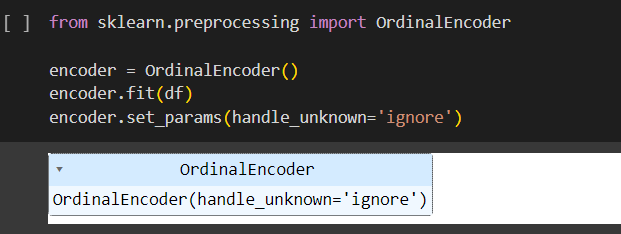
**Tạo đối tượng mới:** ở đây chúng ta sẽ giả định tạo một đối tượng mới để dự đoán giá của nó theo mô hình đã train được



Hình 30. Tạo một đối tượng xe hơi mới

**Fit các thuộc tính của dữ liệu vào 2 lớp OrdinalEncoder và StandardScaler:**

* OrdinalEncoder: Fit các đối tượng có dữ liệu Object thành dạng số
* StandardScaler: Chuẩn hóa các dữ liệu dạng số sau khi đã encode



Hình 31. Sử dụng lớp OrdinalEncoder

A screenshot of a computer program

Description automatically generated with medium confidence

Hình 32. Sử dụng lớp StandardScaler

**Fit các thuộc tính của đối tượng mới tạo vào 2 lớp trên:**

A screen shot of a computer program

Description automatically generated with low confidence

Hình 33. Hàm dự đoán giá

**Kết quả:**

**A screenshot of a computer

Description automatically generated with low confidence**

Hình 34. Kết quả dự đoán

### Mô hình hóa dữ liệu

* Đã xây dựng được 2 mô hình dự đoán giá xe hơi dựa trên 2 thuật toán là Random Forest Regression và XGBoost Regression với độ chính xác được đánh giá khá cao. Random Forest Regression cho R2 là 85.31%. Với mô hình XGBoost Regression cho số liệu đáng tin cậy hơn Linear Regression với kết quả R2 là 87.2%.
* Có thể tăng độ chính xác của mô hình XGBoost Regression bằng việc thay đổi các bộ tham số đầu vào của mô hình để chọn ra bộ tham số tối ưu hơn
* Nhìn chung, việc sử dụng thuật toán nào trong các bài toán khác phụ thuộc vào dữ liệu đầu vào và kết quả mong muốn của bài toán đó.

# Tài liệu tham khảo

1. [Beautiful Soup](https://beautiful-soup-4.readthedocs.io/en/latest/)
2. [Random Forest Regression](https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html)
3. [XGBoost For Regression](https://xgboost.readthedocs.io/en/stable/)
4. [MAE, MSE and RMSE](https://viblo.asia/p/danh-gia-model-trong-machine-learing-RnB5pAq7KPG)
5. [Visualize Data](https://www.youtube.com/watch?v=MhVXgurOwIw&t=17s)