|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN HỌC**

**XÂY DỰNG MÔ HÌNH DỰA TRÊN PCA VÀ**

**HỒI QUY ĐA BIẾN**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **THỐNG KÊ MÁY TÍNH VÀ ỨNG DỤNG**

Giảng viên hướng dẫn : Hứa Thị Phượng Vân

Sinh viên thực hiện :

2286400022 - Hồ Nguyễn Hoàng Phát

2286400042 - Trần Lê Vân

2286400481 - Nguyễn Thị Thu Ngân

2286400001 – Phạm Quốc An

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2025

|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN HỌC**

**XÂY DỰNG MÔ HÌNH DỰA TRÊN PCA VÀ**

**HỒI QUY ĐA BIẾN**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **THỐNG KÊ MÁY TÍNH VÀ ỨNG DỤNG**

Giảng viên hướng dẫn : Hứa Thị Phượng Vân

Sinh viên thực hiện :

2286400022 - Hồ Nguyễn Hoàng Phát

2286400042 - Trần Lê Vân

2286400481 - Nguyễn Thị Thu Ngân

2286400001 – Phạm Quốc An

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

TPHCM, ngày……tháng……năm 2024

Giáo viên hướng dẫn

(Ký tên, đóng dấu)

LỜI CAM ĐOAN

Chúng tôi, Hồ Nguyễn Hoàng Phát, Trần Lê Vân, Phạm Quốc An, Nguyễn Thị Thu Ngân xin cam đoan rằng:

Tất cả thông tin và phân tích trình bày trong báo cáo này được thực hiện một cách chính xác và trung thực.

Mọi dữ liệu, nhận định hoặc ý kiến được trích dẫn từ các nguồn khác đều đã được nêu rõ nguồn gốc và trích dẫn đúng quy định. Tôi cam đoan rằng không có bất kỳ hành vi sao chép hoặc sử dụng thông tin không hợp pháp nào từ các nguồn khác.

Bài báo cáo này là kết quả của công trình nghiên cứu độc lập của chúng tôi và chưa từng được công bố tại bất kỳ nơi nào khác. Tôi cam đoan đã tuân thủ nghiêm ngặt các quy tắc và quy định của môn học, bao gồm việc tham khảo và áp dụng các công cụ nghiên cứu một cách hợp lệ. Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình.

Tôi hy vọng rằng bài báo cáo này sẽ cung cấp những thông tin hữu ích cho các nhà nghiên cứu , doanh nghiệp. Góp phần vào việc hiểu rõ hơn về mạng xã hội ngày nay.

TPHCM, ngày……tháng 10 năm 2024

**Sinh viên**

MỤC LỤC

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 3](#_Toc187183155)

[LỜI CAM ĐOAN 5](#_Toc187183156)

[MỤC LỤC 6](#_Toc187183157)

[CHƯƠNG I DỮ LIỆU TỰ THU THẬP 9](#_Toc187183158)

[1 Giới thiệu dữ liệu 9](#_Toc187183159)

[2 Tiền xử lý dữ liệu 10](#_Toc187183160)

[2.1 Phân bố của từng biến 12](#_Toc187183161)

[2.2 Outliers 13](#_Toc187183162)

[2.3 Giảm chiều dữ liệu và xử lý missing value 16](#_Toc187183163)

[3 Linear Resgression 20](#_Toc187183164)

[4 Random Forest 22](#_Toc187183165)

[5 LSTM 24](#_Toc187183166)

[CHƯƠNG II DỰ ĐOÁN TỶ LỆ RỜI ĐI CỦA KHÁCH HÀNG 26](#_Toc187183167)

[1 Giới thiệu 26](#_Toc187183168)

[1.1 Giới thiệu dữ liệu: 26](#_Toc187183169)

[1.2 Thông tin chung 26](#_Toc187183170)

[1.3 Chi tiết dữ liệu: 27](#_Toc187183171)

[2 Tiền xử lý dữ liệu 28](#_Toc187183172)

[2.1 Missing Values 29](#_Toc187183173)

[2.2 Outliers 30](#_Toc187183174)

[2.3 Phân phối dữ liệu 31](#_Toc187183175)

[2.4 Chuyển dữ liệu về dạng số - Mã hoá dữ liệu sử dụng One Hot Encoding 33](#_Toc187183176)

[2.5 Đa cộng tuyến VIF 35](#_Toc187183177)

[2.6 Scale Data – Chuẩn hoá dữ liệu 38](#_Toc187183178)

[2.7 Splitting train/test 39](#_Toc187183179)

[3 Logistic Regression 39](#_Toc187183180)

[4 RandomForest 41](#_Toc187183181)

[5 Kết luận 42](#_Toc187183182)

[CHƯƠNG III XÂY DỰNG MÔ HÌNH DỰA TRÊN PCA 43](#_Toc187183183)

[1 Giới thiệu bộ dữ liệu 43](#_Toc187183184)

[2 Tiền xử lý dữ liệu 44](#_Toc187183185)

[2.1 Dữ liệu bị trùng lặp 44](#_Toc187183186)

[2.2 Phân phối của các biến 45](#_Toc187183187)

[2.3 Xử lý giá trị thiếu (missing value) 49](#_Toc187183188)

[2.4 Phát hiện giá trị ngoại lai (outlier) 51](#_Toc187183189)

[2.5 Xử lý dữ liệu phân loại bằng phương pháp One-Hot Encoding 53](#_Toc187183190)

[2.6 Chuyển đổi dữ liệu phân loại về thành dạng số (Binary Encoding) 54](#_Toc187183191)

[2.7 Chuẩn Hóa Dữ Liệu bằng StandardScaler 55](#_Toc187183192)

[2.8 Xây dựng mô hình PCA 56](#_Toc187183193)

[3 Random Forest Regressor: 59](#_Toc187183194)

[4 Mô hình XGBoost Regressor: 60](#_Toc187183195)

[5 Mô hình LightGBM Regressor: 61](#_Toc187183196)

[CHƯƠNG IV DỰ ĐOÁN CHI PHÍ BẢO HIỂM HẰNG NĂM CHO XE 63](#_Toc187183197)

[1 Giới thiệu dữ liệu 63](#_Toc187183198)

[2 Tiền xử lý dữ liệu 64](#_Toc187183199)

[2.1 Kiểm tra dữ liệu 64](#_Toc187183200)

[2.2 Xử lý dữ liệu thiếu 64](#_Toc187183201)

[2.3 Kiểm tra và xử lý outliers 66](#_Toc187183202)

[2.4 Đa cộng tuyến 68](#_Toc187183203)

[2.5 Hồi quy tuyến tính 68](#_Toc187183204)

[2.6 Biểu đồ biến liên tục 71](#_Toc187183205)

[2.7 Biểu đồ phân phối của các cột số 72](#_Toc187183206)

[2.8 Ma trận tương quan 73](#_Toc187183207)

# DỮ LIỆU TỰ THU THẬP

* 1. Giới thiệu dữ liệu

Trong những năm gần đây Bitcoin đã nổi lên mạnh mẻ như một biểu tượng của sự đổi mới trong lĩnh vực tài chính và công nghệ. Sự biến động mạnh mẽ của giá trị Bitcoin không chỉ thu hút sự chú ý của các nhà đầu tư mà còn là chủ đề nghiên cứu của nhiều chuyên gia kinh tế và khoa học dữ liệu. Việc hiểu rõ những yếu tố ảnh hưởng đến giá Bitcoin là chìa khóa để đưa ra các quyết định giao dịch sáng suốt và tối ưu hóa lợi nhuận. Để làm được điều này, dữ liệu lịch sử về Bitcoin đóng vai trò vô cùng quan trọng. Nó cung cấp một cái nhìn toàn diện về xu hướng biến động giá, khối lượng giao dịch và các chỉ số liên quan qua từng ngày. Bộ dữ liệu mà chúng tôi đang sử dụng chính là nền tảng để xây dựng các mô hình phân tích và dự đoán trong tương lai.

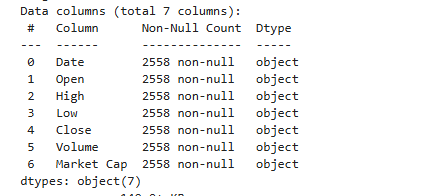
Bộ dữ liệu Bitcoin đã được thu thập từ trang web (<https://coinmarketcap.com/currencies/bitcoin/historical-data/>) ghi lại chi tiết về sự thay đổi giá cả, khối lượng giao dịch từ ngày 02/01/2018 đến 02/01/2015. Bộ dữ liệu có 2558 quan trắc, gồm 7 biến:

1. Date – Ngày giao dịch của Bitcoin
2. Open - Giá mở cửa của Bitcoin trong ngày
3. High - Giá cao nhất của Bitcoin trong ngày
4. Low - Giá thấp nhất của Bitcoin trong ngày
5. Close - Giá đóng cửa của Bitcoin trong ngày
6. Volume – Khối lượng giao dịch của Bitcoin trong ngày
7. Market Cap – Tổng vốn hóa thị trường của Bitcoin tại thời điểm giao dịch.

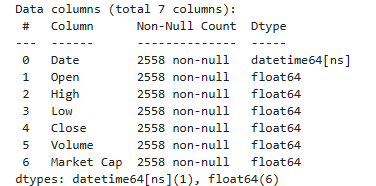


Bộ dữ liệu Bitcoin cung cấp cái nhìn chi tiết về sự biến động của giá và khối lượng giao dịch qua từng ngày. Với 2558 quan sát và 7 cột, bồ dữ liệu cho phép phân tích theo xu hướng dài hạn, giúp nhận diện sự thay đổi giá cả qua thời gian. Biến **Open**, **High**, **Low** và **Close** đại diện cho các mốc giá quan trọng trong một phiên giao dịch, phản ánh rõ ràng sự dao động giá của Bitcoin. Khối lượng giao dịch (**Volume**) là chỉ số thể hiện mức độ sôi động của thị trường, **Market Cap** cho biết tổng giá trị của Bitcoin đang lưu thông tại thời điểm đó. Thông qua việc phân tích các biến này, chúng ta có thể phát hiện các xu hướng, đánh giá rủi ro và dự đoán giá trong tương lai.

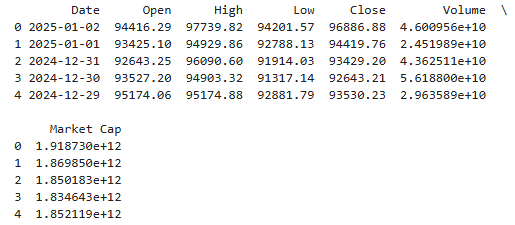
* 1. Tiền xử lý dữ liệu



Khi kiểm tra dữ liệu ban đầu, có thể nhận thấy tất cả các cột trong bộ dữ liệu đều thuộc kiểu ‘object’. Điều này chỉ ra rằng các giá trị trong các cột **Open, High, Low, Close, Volume** và **Market Cap** đang được lưu dưới dạng văn bản thay vì dạng số. Việc để dữ liệu ở dạng chuỗi sẽ gây ra khó khăn trong các phép tính toán và trực quan hóa. Nên em đã chuyển đổi sang kiểu số cũng như loại bỏ ký tự ‘$’ và dấu phẩy ‘ , ’, kết quả được thể hiện ở các hình dưới đây.

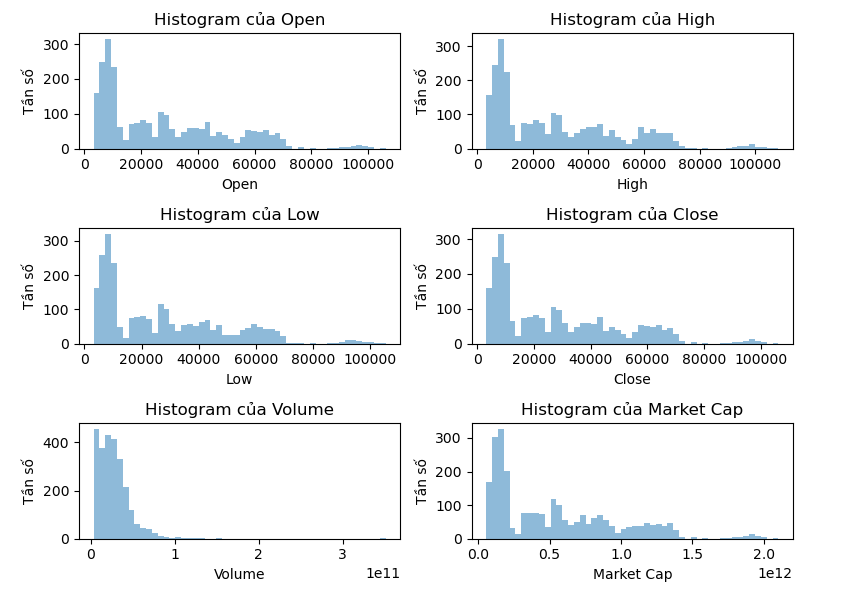


Sau khi chuyển đổi dạng chuỗi về dạng số



Dữ liệu sau khi loại bỏ ký tự và dấu phẩy

* + 1. Phân bố của từng biến

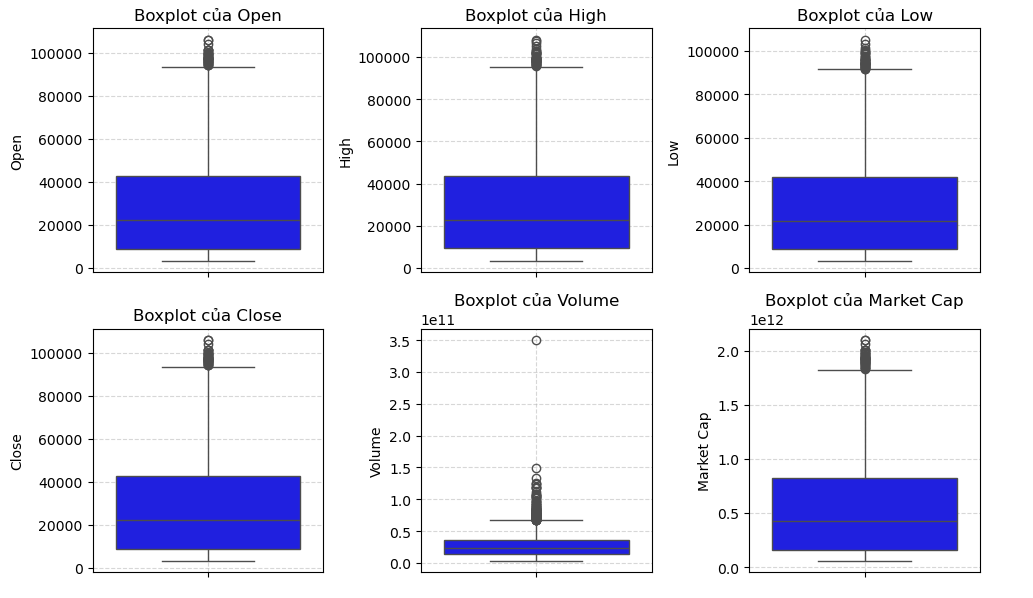


Open, High, Low, Close: các biến liên quan đến giá đều có phân phối lệch phải. Điều này cho thấy phần lớn giá Bitcoin giao dịch ở mức thấp, nhưng có những thời điểm giá tăng cao đột biến, thể hiện qua phần đuôi kéo dài của biểu đồ. Các mức giá vượt quá 60,000 USD có tần suất thấp, chứng tỏ các phiên giao dịch đạt đỉnh không phổ biến.

Volume: phân phối khối lượng giao dịch cũng có đuôi dài, phản ánh một số giao dịch có khối lượng lớn hơn nhiều so với phần còn lại. Phần lớn khối lượng giao dịch nằm ở mức thấp dưới 50 tỷ USD, nhưng có những điểm vượt quá 100 tỷ USD.

Market Cap: phần lớn tập trung dưới 500 tỷ USD, một số điểm dữ liệu cho thấy vốn hóa (Market Cap) vượt quá 1 nghìn tỷ USD, phản ánh các giai đoạn Bitcoin bùng nổ và chiếm lĩnh thị trường tiền điện tử.

* + 1. Outliers



Open, High, Low, Close: phần lớn các giá trị tập trung trong khoảng nhất định, nhưng có nhiều điểm ngoại lai phía trên, phản ánh các giai đoạn tăng giá đột biến của Bitcoin

Volume, Market Cap: cũng có sự phân tán lớn, với nhiều điểm vượt xa phần còn lại, cho thấy sự xuất hiện của các phiên giao dịch bất thường hoặc các đợt tăng trưởng mạnh

Kết luận:

Open, High, Low, Close: có xu hướng tập trung ở mức thấp trong phần lớn thời gian, nhưng cả hai biểu đồ đều cho thấy sự tăng giá đột biến trong một số giai đoạn, được thể hiện qua các điểm ngoại lai. Điều này phản ánh đặc điểm của thị trường Bitcoin thường xuyên biến động mạnh với những thời kỳ tăng trưởng bùng nổ.

Volume và Market Cap: có sự phân bố không đồng đều. Các giá trị thấp chiếm đa số, trong khi các điểm cao vượt trội cho thấy có những phiên giao dịch lớn, có thể do sự tham gia của các tổ chức hoặc các sự kiện tác động mạnh đến thị trường.

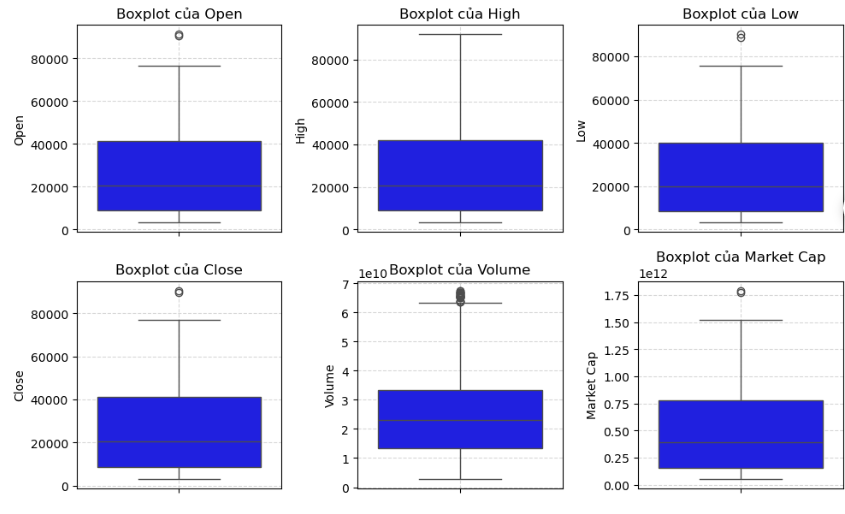
Sự tồn tại của nhiều outliers trong cả hai biểu đồ nhấn mạnh tính không ổn định và biến động cao của thị trường Bitcoin. Điều này vừa là rủi ro nhưng cũng là cơ hội lớn cho các nhà đầu tư trong các giai đoạn tăng giá mạnh.

Để đảm bảo dữ liệu ổn định và giảm thiểu sự ảnh hưởng của các giá trị bất thường, phương pháp IQR (Interquartile Range) đã được sử dụng để loại bỏ outliers. Các giá trị được xem là ngoại lai và bị loại bỏ khỏi dữ liệu khi nằm ngoài khoảng:

[Q1 – 1.5 \* iQR, Q3 + 1.5 \* IQR]

Trong đó:

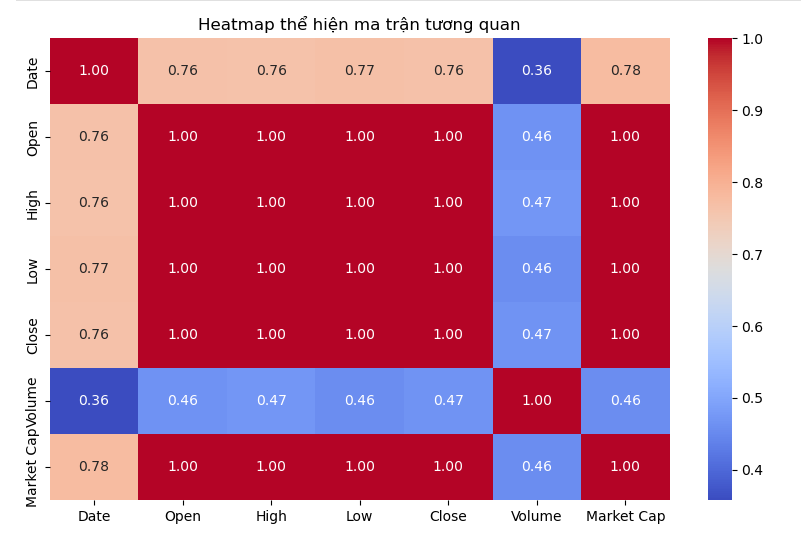
* Q1 (phân vị 25%) và Q3 (phân vị 75%) là các điểm chia dữ liệu thành các phần tư.
* IQR (Interquartile Range) là sự chênh lệch giữa Q3 và Q1, thể hiện độ phân tán của 50% dữ liệu giữa hai phân vị này.



Sau khi xử lý ngoại lai (outliers), các biểu đồ Boxplot (hình 2.4) cho thấy dữ liệu của các biến Open, High, Close và Market Cap đã trở nên ổn định hơn, ít điểm ngoại lai hơn.

Volume có sự phân tán mạnh hơn với một vài điểm ngoại lai, cho thấy sự biến động lớn về khối lượng giao dịch trong một số giai đoạn.

* + 1. Giảm chiều dữ liệu và xử lý missing value



Ma trận tương quan cho thấy:

Các biến Open, High, Low, Close có mức độ tương quan rất cao, cho thấy sự liên quan chặt chẽ giữa các biến giá trong mỗi phiên giao dịch. Khi giá mở cửa (Open) tăng, giá cao (High) tăng, giá thấp (Low) và giá đóng cửa (Close) cũng có xu hướng tăng theo, cho thấy có hiện tượng đa cộng tuyến

Volume có tương quan thấp so với các biến còn lại, đặc biệt là với Date, cho thấy khối lượng giao dịch không phải lúc nào cũng diễn ra đồng bộ với biến động giá hoặc thời gian.

Market Cap: có mức độ tương quan manh với tất cả các biến giá, điều này phản ánh sự ảnh hưởng trực tiếp của giá Bitcoin đến vốn hóa thị trường. Khi giá Bitcoin tăng hoặc giảm, vốn hóa thị trường thay đổi tương ứng.

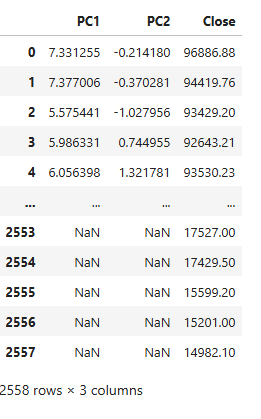
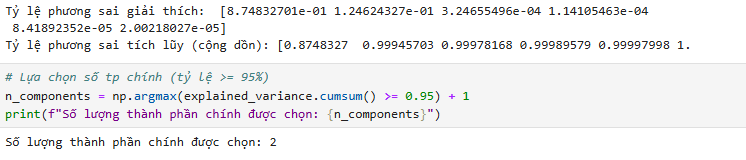
Để giải quyết vấn đề đa cộng tuyến và giảm bớt sự dư thừa thông tin giữa các biến có tương quan cao, sử dụng PCA (Principal Component Analysis), phương pháp này không chỉ giảm chiều dữ liệu mà còn giữ lại được thông tin quan trọng, giúp mô hình trở nên ổn định và hiểu quả hơn trong việc dự đoán.



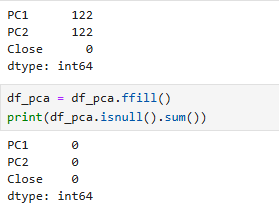
Các giá trị tỷ lệ phương sai giải thích cho thấy mức độ quan trọng của từng thành phần chính. Cụ thể, các thành phần chính đầu tiên giữ lại phần lớn sự biên thiên trong dữ liệu, trong khi các thành tiếp theo đóng góp ít hơn. Tỷ lệ phương sai giải thích của các thành phần đầu tiên là rất cao, cho thấy hiệu quả của PCA trong việc rút gọn dữ liệu mà vẫn duy trì được tính đại diện của dữ liệu gốc.

Tỷ lệ phương sai tích lũy cung cấp tổng cộng tỷ lệ phương sai giải thích của các thành phần chính, giúp đánh giá sự đóng góp của các thành phần vào việc giữ lại thông tin của dữ liệu.

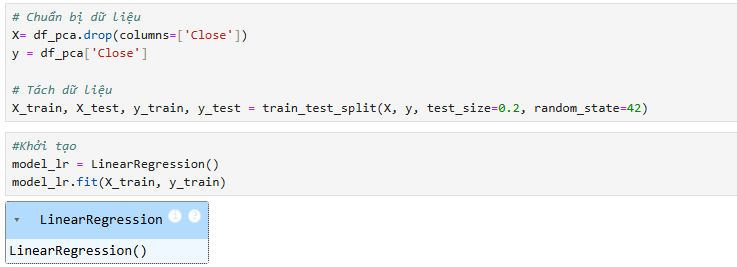
Tiếp theo, mã lệnh kiểm tra tổng phương sai tích lũy và lựa chọn số lượng thành phần chính sao cho tổng phương sai tích lũy đạt ít nhất 95% (Hình 2.7). Điều này đảm bảo rằng 95% thông tin quan trọng của dữ liệu được giữ lại trong mô hình, nhưng chỉ với một số ít các thành phần chính. Số thành phần chính được chọn có kết quả là 2



Sau khi giảm chiều dữ liệu, kết quả của PCA được kết hợp với biến Close từ dữ liệu gốc. Tạo thành một DataFrame mới (df\_pca) với ba cột: PC1, PC2, và Close (hình 2.8). Khi kiểm tra dữ liệu thiếu (hình 2.9), ta nhận thấy giá trị trong cột PC1 và PC2 có 122 giá trị thiếu. Để xử lý vấn đề này, phương pháp *ffill()* (forward fill) được sử dụng để thay thế các giá trị NaN bằng giá trị không bị thiếu từ dòng trước đó. Sau khi áp dụng *ffill()*, dữ liệu không còn thiếu giá trị.



* 1. Linear Resgression

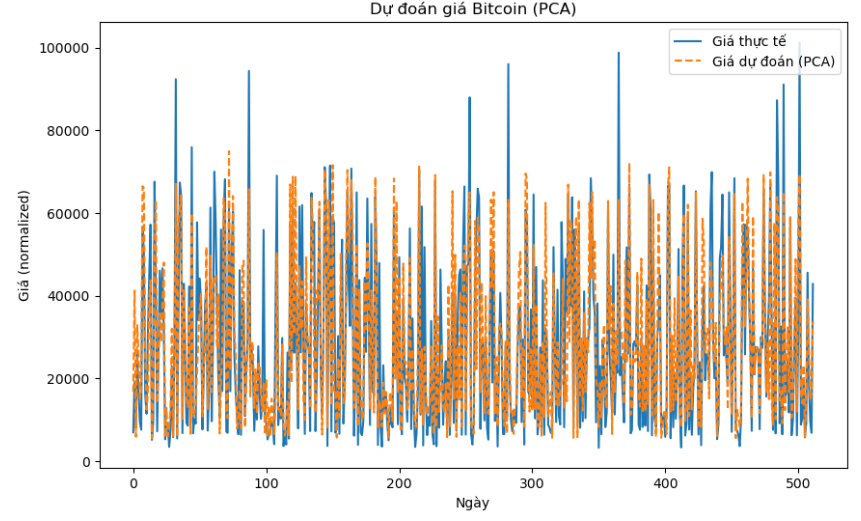


Dữ liệu được chuẩn bị bằng cách loại bỏ biến Close từ DataFrame df\_pca để tạo ra X (các đặc trưng) và y (biến mục tiêu là Close). Biến X sẽ bao gồm các thành phần chính (PC1, PC2) từ PCA. Dữ liệu được chia thành hai phần: tập huấn luyện (80% dữ liệu) và tập kiểm tra (20% dữ liệu) thông qua hàm *train\_test\_split()*. Điều này giúp mô hình học từ tập huấn luyện và được đánh giá trên tập kiểm tra. Một mô hình hồi quy tuyến tính (LinearRegression()) được khởi tạo và huấn luyện bằng cách sử dụng tập huấn luyện (X\_train, y\_train).



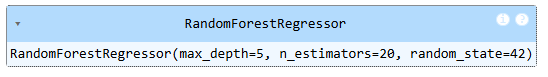
Sau khi chạy mô hình ta có các kết quả như sau:

* MAE (Mean Absolute Error): trung bình độ sai số tuyệt đối, cho thấy mức độ sai lệch trung bình giữa các dự đoán và giá tri thực tế là khoảng 7895.89
* RMSE (Root Mean Squared Error): Căn bậc hai của sai số bình phương trung bình, dùng để đo mức độ chính xác của mô hình với độ nhạy cảm với sai số lớn, có mức sai số căn bậc hai là 11077.21, cho thấy mô hình có độ sai lệch khá lớn, nhưng vẫn có thể chấp nhận được.
* R² (R-squared): Chỉ số xác định độ phù hợp của mô hình, cho biết tỷ lệ biến động của dữ liệu được giải thích bởi mô hình khoảng 73% biến động trong giá Close, đây là một mức độ tương đối tốt.



Biểu đồ so sánh giữa giá trị thực tế và giá dự đoán có thể thấy:

* Giá thực tế (màu xanh dương) có sự biến động lớn, phản ánh các thay đổi mạnh mẽ trong giá trị của Bitcoin trong suốt thời gian.
* Giá dự đoán (đường nét đứt màu cam) theo xu hướng cũng biến động khá gần với giá thực tế, tuy nhiên, có sự chệch lệch nhất định trong một số khoảng thời gian, đặc biệt là vào những đỉnh cao hoặc đáy sâu của giá.
  1. Random Forest

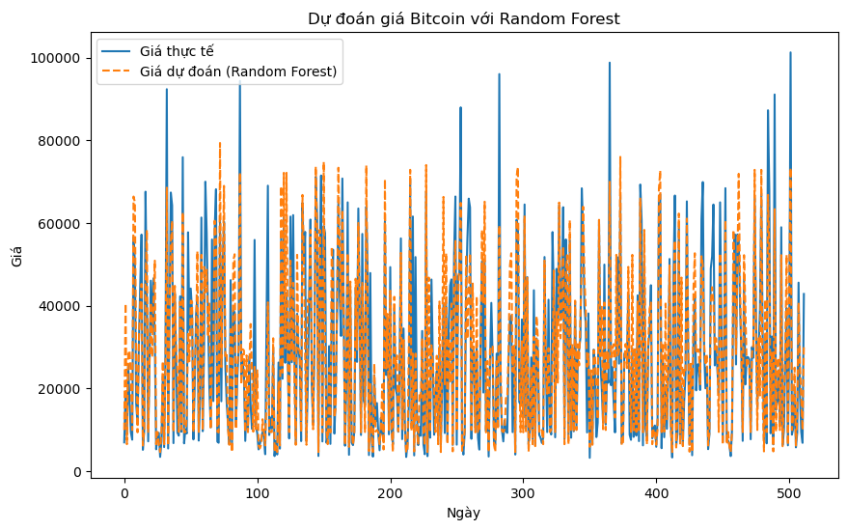


Sau khi thực hiện Hồi quy tuyến tính, vì muốn cho ra kết quả dự đoán tốt hơn, nên tiếp tục thực hiện thêm mô hình Random Forest Regressor để cải thiện độ chính xác trong việc dự đoán giá Bitcoin. Mô hình Random Forest được khởi tạo với 20 cây quyết định (n\_estimators=20) và độ sâu của mỗi cây là 5 (max\_depth=5). Các tham số này giúp hạn chế việc quá khớp (overfitting) và đảm bảo mô hình không quá phức tạp.



Mô hình Random Forest có kết quả như sau:

* MAE (Mean Absolute Error): trung bình độ sai số tuyệt đối, cho thấy mức độ sai lệch trung bình giữa các dự đoán và giá tri thực tế là khoảng 6450,71
* RMSE (Root Mean Squared Error): Căn bậc hai của sai số bình phương trung bình, dùng để đo mức độ chính xác của mô hình với độ nhạy cảm với sai số lớn, có mức sai số căn bậc hai là 9638.96, cho thấy mô hình có độ sai lệch đáng kể, nhưng vẫn có thể chấp nhận được.
* R² (R-squared): Chỉ số xác định độ phù hợp của mô hình, cho biết tỷ lệ biến động của dữ liệu được giải thích bởi mô hình khoảng 80% biến động trong giá Close, đây là một mức độ tương tốt.



Biểu đồ trên so sánh giá thực tế và giá dự đoán sử dụng mô hình Random Forest:

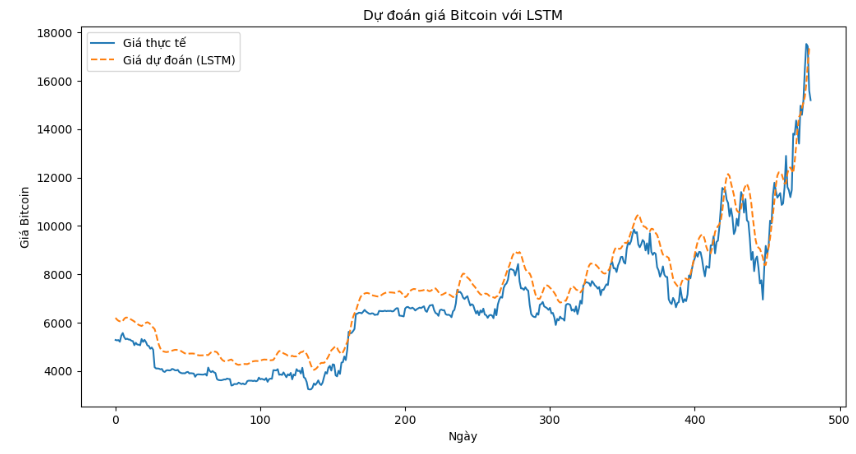
* Giá thực tế (màu xanh dương) cho thấy sự biến động mạnh mẽ, phản ánh các dao động không thể đoán trước của giá Bitcoin trong thời gian. Các đỉnh và đáy của giá thường xuyên thay đổi.
* Giá dự đoán (đường nét đứt màu cam) từ mô hình Random Forest có xu hướng bám sát khá gần giá thực tế, nhưng vẫn có một số sai lệch ở các đỉnh và đáy. Mô hình dự đoán có thể không bắt kịp chính xác những biến động lớn trong một số khoảng thời gian, nhưng nhìn chung vẫn thể hiện xu hướng chung của giá Bitcoin.
  1. LSTM

Mô hình Random Forest đã có cải thiện hơn so với mô hình hồi quy tuyến tính, nhưng để tiếp tục nâng cao độ chính xác và khả năng dự đoán với dữ liệu chuỗi thời gian như giá Bitcoin, em đã thử áp dụng mô hình LSTM (Long Short-Term Memory). Đây là một mô hình học sâu, thuộc nhóm mạng nơ-ron hồi quy (RNN), chuyên dụng cho dữ liệu có tính tuần tự và phụ thuộc vào quá khứ, giúp mô hình học và dự đoán các xu hướng trong chuỗi thời gian một cách hiệu quả hơn. Kết quả đã có cải thiện so với hai mô hình trước đó.



Mô hình LSTM có kết quả như sau:

* MAE (Mean Absolute Error): trung bình độ sai số tuyệt đối, cho thấy mức độ sai lệch trung bình giữa các dự đoán và giá tri thực tế là khoảng 768.45, đây là một sai số chấp nhận được.
* RMSE (Root Mean Squared Error): Căn bậc hai của sai số bình phương trung bình, dùng để đo mức độ chính xác của mô hình với độ nhạy cảm với sai số lớn, có mức sai số căn bậc hai là 836.66, cho thấy mô hình có sai lệch nhỏ nhưng vẫn có thể cải thiện hơn nữa.
* R² (R-squared): Chỉ số xác định độ phù hợp của mô hình, cho biết tỷ lệ biến động của dữ liệu được giải thích bởi mô hình khoảng 89% biến động trong giá Close, cho thấy mô hình rất tốt trong việc dự đoán giá Bitcoin.



Biểu đồ trên so sánh giá thực tế và giá dự đoán sử dụng mô hình LSTM:

* Giá thực tế (màu xanh dương) có sự biến động mạnh mẽ, đặc biệt là những đợt tăng giá đột ngột về cuối biểu đồ, phản ánh các biến động khó lường của thị trường Bitcoin.
* Giá dự đoán (đường nét đứt màu cam) từ mô hình LSTM theo xu hướng bám sát giá thực tế khá chặt chẽ, nhưng vẫn có một số sai lệch ở những điểm cực trị và sự biến động mạnh trong một số giai đoạn.

Qua ba mô hình trên thì mô hình LSTM là mô hình hiệu quả nhất với khả năng dự đoán cao nhất và sai số thấp nhất. Mô hình này nắm được xu hướng biến động của giá Close của Bitcoin rất tốt, đặc biệt là trong các giai đoạn biến động mạnh.

# DỰ ĐOÁN TỶ LỆ RỜI ĐI CỦA KHÁCH HÀNG

* 1. Giới thiệu
     1. Giới thiệu dữ liệu:

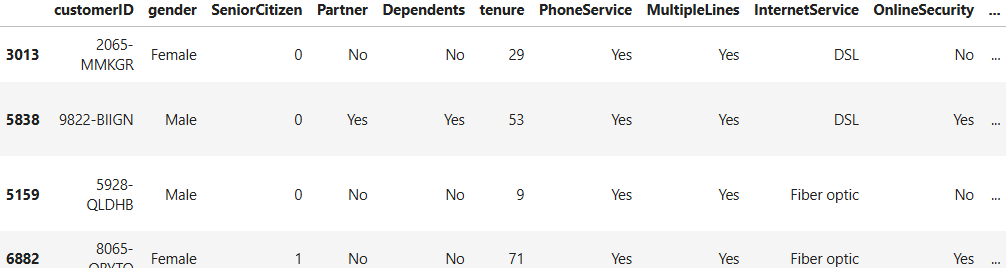
Bộ dữ liệu “Customer Churn” được thu thập trên Kaggle. Miêu tả thông tin khách hàng thuộc một công ty liên quan đến ngành viễn thông. Mục tiêu của bộ dữ liệu này là dự đoán tỷ lệ rời đi của khách hàng thông qua việc phân tích các yếu tố về gia đình lẫn hành vi sử dụng dịch vụ. Dẫn đến việc họ có tiếp tục sử dụng dịch vụ hay không. Nhờ phân tích nó, doanh nghiệp có thể đưa ra các chiến lược nhằm thu hút khách hàng mới lẫn giữ chân khách hàng cũ.

* + 1. Thông tin chung

Số lượng dòng: 7043 (7043 khách hàng).

Số lượng cột: 21 cột – Gồm thông tin cá nhân và dịch và mà họ sử dụng.

Biến mục tiêu: Churn – là cột thứ 21 thể hiện khách hàng có tiếp tục sử dụng dịch vụ không (Gồm 2 giá trị Yes hoặc No).



* + 1. Chi tiết dữ liệu:

Thông tin cá nhân:

customerID: Mã ID của khách hàng.

gender: giới tính.

Partner: Tình trạng hôn nhân.

Dependents: Có người phụ thuộc (con cái) không.

SeniorCitizen: Khách hàng là người lớn tuổi hay không (Int).

tenure: Số tháng khách hàng đã sử dụng dịch vụ (Int).

* Thông tin dịch vụ:

PhoneService: Dịch vụ điện thoại.

MultipleLines: Có nhiều đường dây điện thoại: Yes/No/No phone Service.

InternetService: Loại dịch vụ khách hàng sử dụng: DSL/Fiber optic/No.

OnlineSecurity: Khách hàng có sử dụng dịch vụ bảo mật online không Yes/No/No internet service.

OnlineBackup: Khách hàng có sử dụng dịch vụ sao lưu dữ liệu online không (dịch vụ hỗ trợ).

DeviceProtection: Dịch vụ bảo vệ thiết bị.

Techsupport: Dịch vụ hỗ trợ kỹ thuật.

StreamingTV: Dịch vụ truyền hình trực tuyến.

StreamingMovies: Xem phim trực tuyến.

* Thông tin hợp đồng và phương thức thanh toán:

Contract: Loại hợp đồng: Month-to-month, One year, Two year.

PaperlessBilling: Khách hàng có sử dụng hoá đơn điện tử không.

PaymentMethod: Phương thức thanh toán Electronic check/Mailed check/Bank transfer (automatic)/Credit card (automatic).

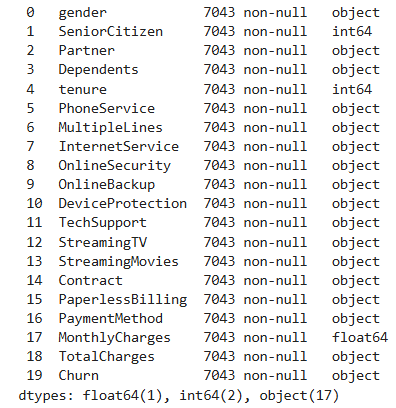
MonthlyCHarges: Cước phí hàng tháng (Float).

TotalCharges: Tổng số tiền khách hàng đã thanh toán (Object).

* Biến mục tiêu (kết quả dự đoán):

Churn: Khách hàng có sử dụng tiếp dịch vụ hay không (Yes: Rời bỏ, No: Tiếp tục sử dụng).

* 1. Tiền xử lý dữ liệu



* + 1. Missing Values

Các cột dữ liệu đều không bị thiếu. Vì vậy, kiểm tra xem có giá trị nào bị rỗng hay không:



Có 11 giá trị rỗng

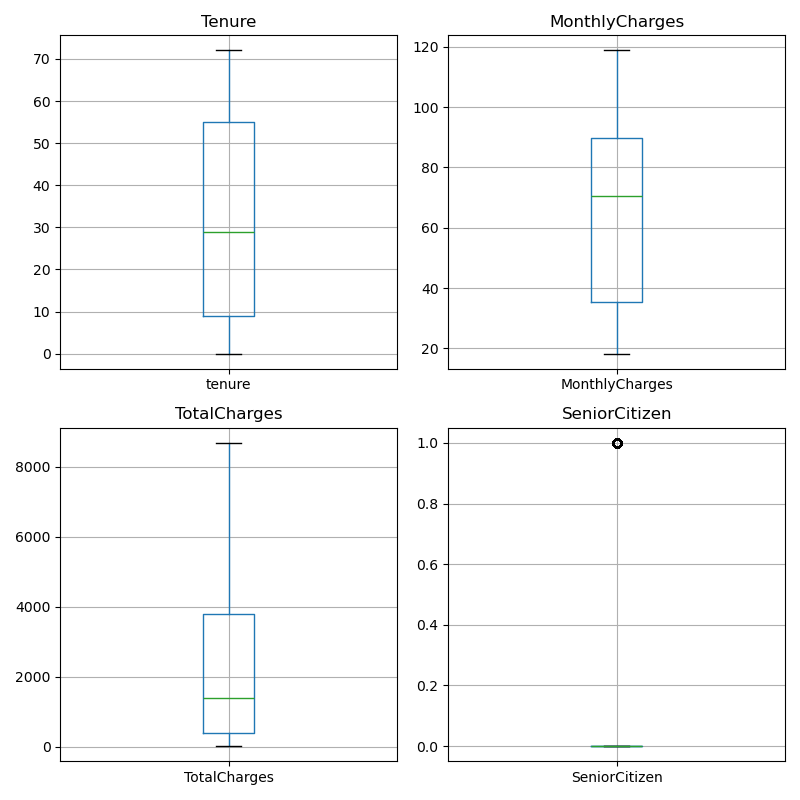
Để xử lý, đưa các giá trị từ rỗng về dạng NaN, từ đó sử dụng SimpleImputer (median) để xử lý dữ liệu thiếu.



Kiểm tra dữ liệu một lần nữa, các giá trị thiếu đã được xử lý.

* + 1. Outliers

Đầu tiên, kiểm tra Boxpot để kiểm tra xem có outlier không



Rất ít outlier xuất hiện, tuy nhiên không quá nghiêm trọng, sẽ giữ nguyên vì để làm tránh mất mát dữ liệu

Tenure: Đa số khách hàng đều ở trong khoảng mức từ 10-60 tháng. Tuy nhiên, có một số khách chỉ sử dụng 1 tháng, hoặc trên mức 70 tháng. Không có outlier rõ ràng nào vượt qua mức thông thường. Cho thấy thời gian sử dụng dịch vụ này hoàn toàn hợp lý -> Không xử lý

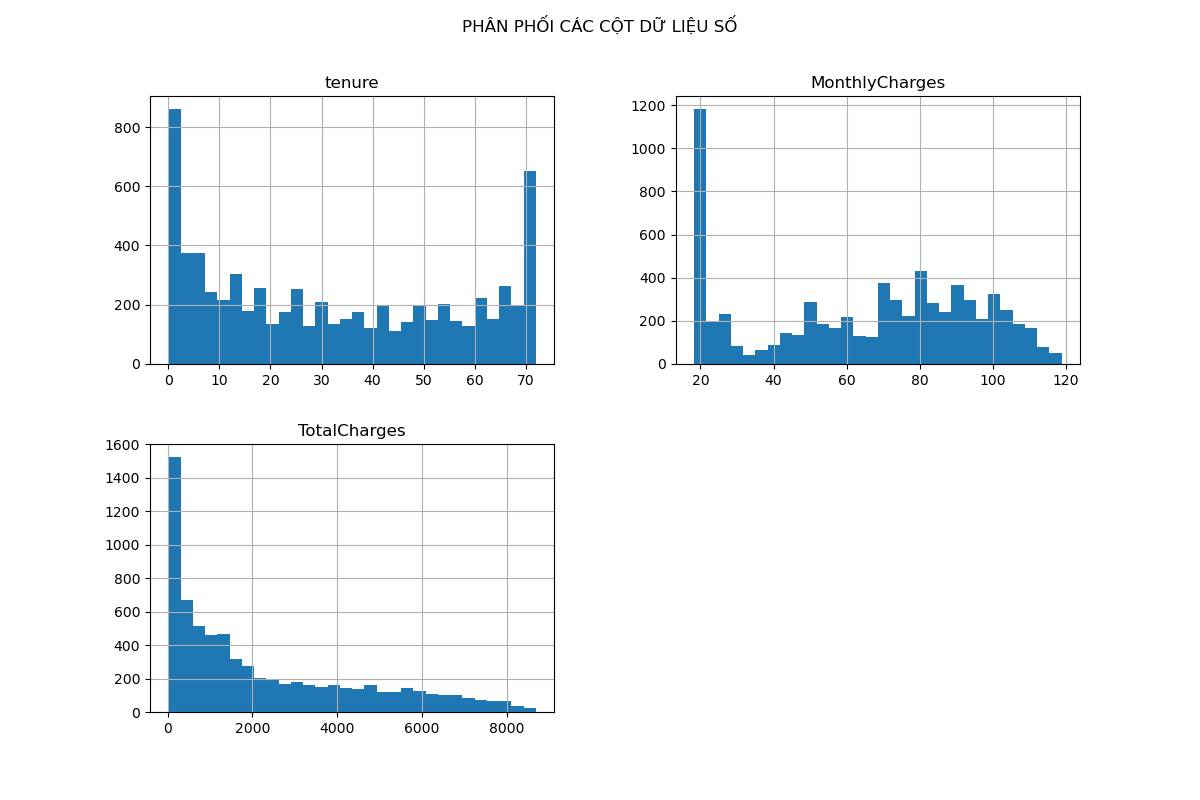
MonthlyCharges: Có một số khách hàng sử dụng cước phí khá cao, lên đến gần 120, lệch khá xa so với dữ liệu trung bình vì khách hàng này sử dụng nhiều dịch vụ. Đây cũng không phải là điều xa lạ nên không cần xử lý. Vì điều này phản ánh mức độ sử dụng dịch vụ của khách hàng có chi tiêu nhiều.

TotalCharges: Ở Boxplot Total, dữ liệu đang chệch về một hướng khá cao, cụ thể có khách lên tới hơn 8000, nhưng điều này phản ánh KHÁCH HÀNG SỬ DỤNG DỊCH VỤ TRONG MỘT KHOẢNG THỜI GIAN LÂU DÀI. Nên không loại bỏ

SeniorCitizen: Không có outlier, chỉ có giá trị 1 và 0

* Kết luận: Không loại bỏ outliers
  + 1. Phân phối dữ liệu

Vẽ biểu đồ 3 cột là dữ liệu số để xem phân phối dữ liệu



Cột 1 ‘tenure’: Phân phối lệch về phía hai bên khá nhiều, bên trái lệnh hơn rõ. Bởi vì đây là những khách hàng mới, sẽ có xu hướng rời bỏ dịch vụ sớm. Vì có thể do đang đi tìm dịch vụ viễn thông tốt hơn.

Tuy nhiên: Càng về lâu về dài, cụ thể là ở hướng bên phải, vẫn có rất nhiều khách hàng sử dụng dịch vụ rất lâu (trên 70 tháng)

-> Ý nghĩa: Cần phát triển những chương trình để giữ chân khách hàng mới, có thể là tạo ưu đãi cho những khách hàng sử dụng lâu dài.

Cột 2 ‘MonthlyCharge’: Hơi lệch nhẹ, vì có một phần khách hàng có chi phí sử dụng dịch vụ cao hơn hẳn (100-120)

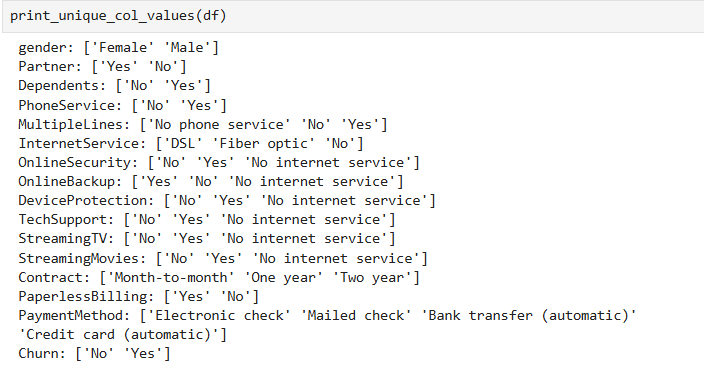
-> Ý nghĩa: Điều này thể hiện khách hàng đang sử dụng nhiều gói dịch vụ cao cấp, hoặc các gói hỗ trợ đi kèm. Cần tập trung phát triển các gói dịch vụ. Sử dụng chiến lược bán chéo (cross-sell) và bán thêm (upsell)

Cột 3 ‘TotalCharges’: Phân phối lệch phải, thể hiện sự tích luỹ chi phí qua thời gian. Phần lớn khách hàng có tổng chi phí ở mức thấp, nhưng khá nhiều khách hàng lại cao hơn nhất nhiều.

Khá dốc ở phần đầu, vì khách hàng mới có tỷ lệ rời đi hơi cao nên tổng doanh thu còn thấp.

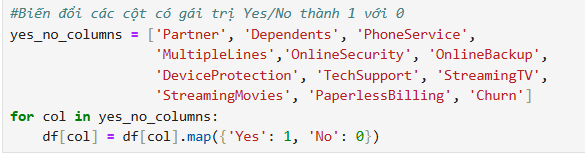
-> Ý nghĩa: Cần xác định nguyên nhân khách hàng rời bỏ dịch vụ sớm, để phục vụ cho việc cải thiện chất lượng dịch vụ, hoặc tạo nhiều chiến dịch dữ chân khách hàng.

* + 1. Chuyển dữ liệu về dạng số - Mã hoá dữ liệu sử dụng One Hot Encoding

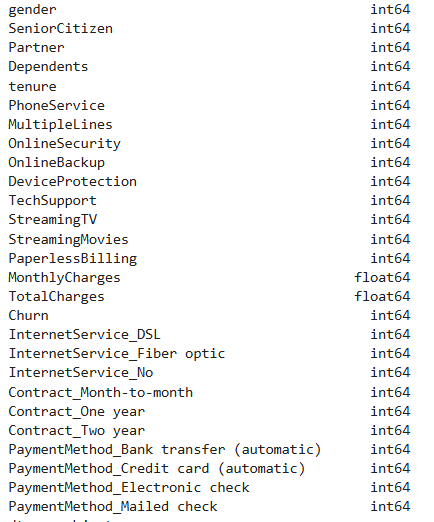


Ở các cột có No phone service, No internet service. Ta biến nó thành No luôn, vì đều thể hiện giá trị giống nhau.

Tiếp theo, biến đổi các cột có giá trị là Yes/No thành 1/0



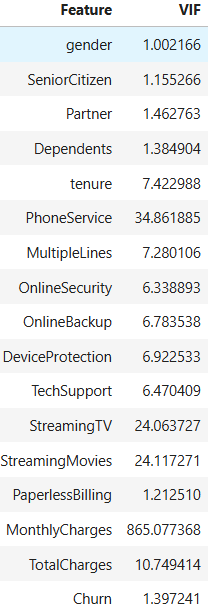
Và cuối cùng là sử dụng One Hot Encoding bằng hàm get\_dummies ở cột InternetService, Contract, PaymentMethod.



Tất cả dữ liệu đã được đưa về dạng số, hoàn tất.

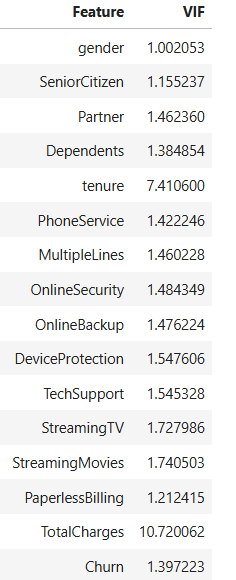
* + 1. Đa cộng tuyến VIF

Kiểm tra đa cộng tuyến trước xem có những cột nào cần được loại bỏ



Cột MonthlyCharges cần được loại bỏ vì có đa cộng tuyến rất cao

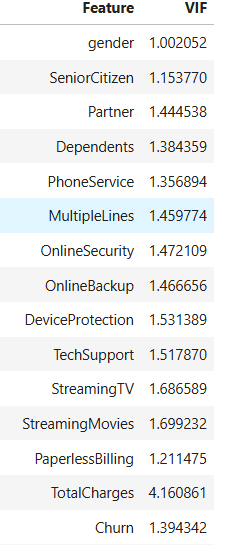
Tiếp tục kiểm tra VIF



Cột ‘tenure’ có VIF = 7.41, cần được loại bỏ

Cột TotalCharges tuy có VIF = 10, cao hơn. Tuy nhiên cần được dữ lại để xem xét tiếp vì đây là cột rất quan trọng. Nói lên mức chi tiêu của khách hàng.

Kiểm tra lại kết quả sau khi loại bỏ ‘tenure’

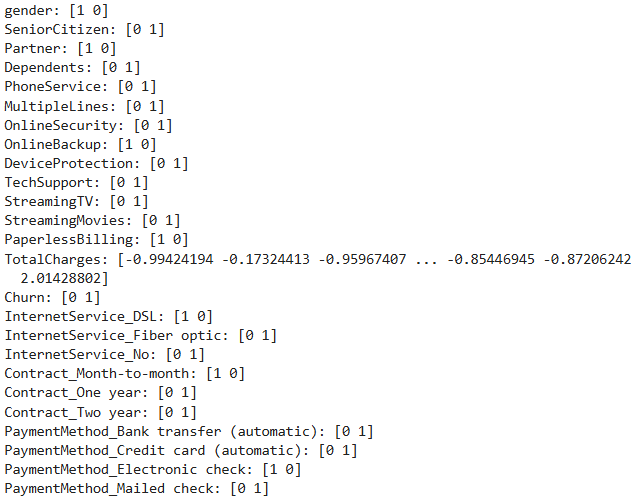


Tất cả các cột đã có VIF dưới mức 5, để tránh loại bỏ đi cột quan trọng. Kết thúc quá trình loại bỏ đa cộng tuyến.

* + 1. Scale Data – Chuẩn hoá dữ liệu

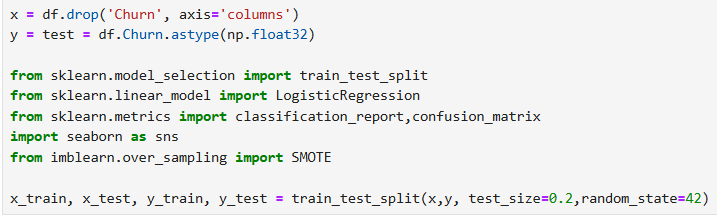
Chỉ còn cột TotalCharges là cần được chuẩn hoá, vì các cột khác đã bị loại bỏ vì đa cộng tuyến.

Kết quả sau khi chuẩn hoá:

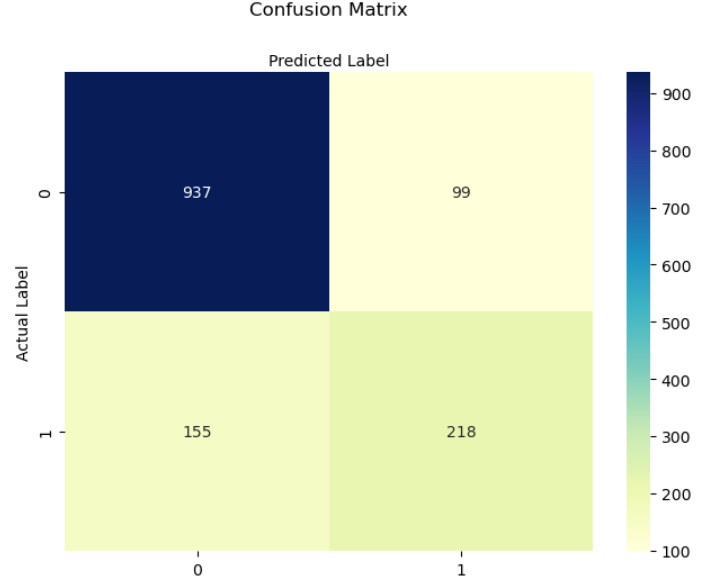
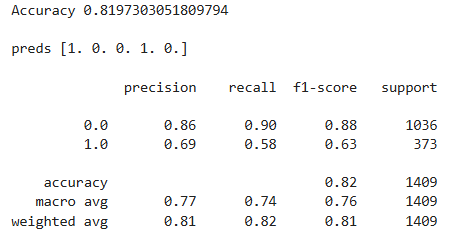
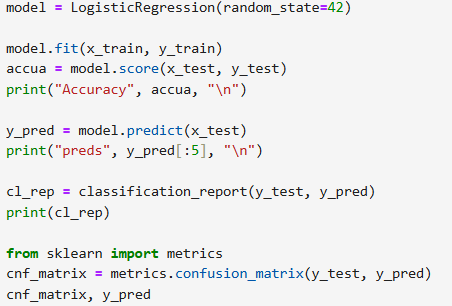


Tất cả dữ liệu đã sẵn sàng để được đưa vào huấn luyện

* + 1. Splitting train/test



* 1. Logistic Regression



Đánh giá kết quả huấn luyện:

Độ chính xác: 81.97%

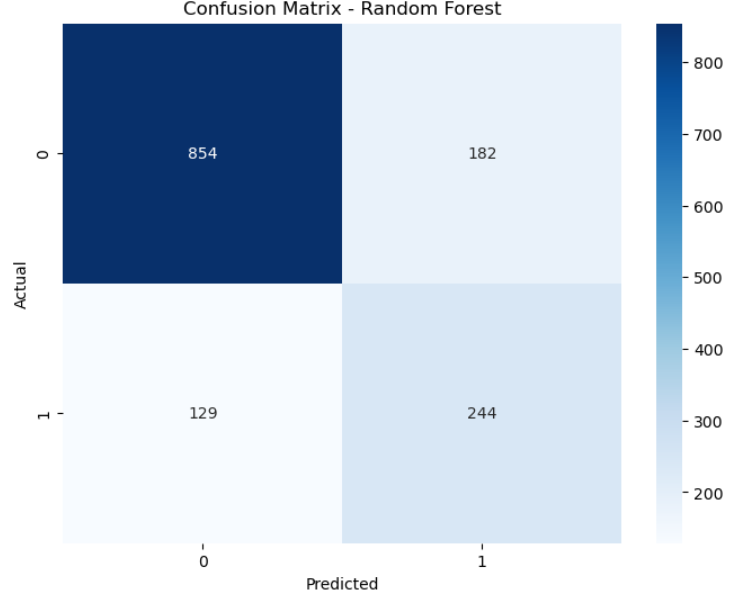
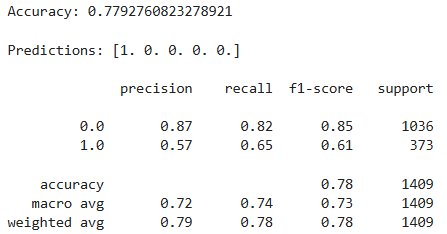
Báo cáo phân loại:

Lớp 0: 86%

Lớp 1: 69%

Precision cao của lớp 0 cho thấy mô hình đã dự đoán rất tốt khách hàng không rời bỏ dịch vụ, nhưng độ chính xác ở lớp 1 còn hơi thấp cho thấy dữ liệu của lớp 1 còn thiếu thông tin đánh giá, hoặc số lượng mẫu còn ít nên không thể cho ra kết quả cao hơn.

* 1. RandomForest



**Độ chính xác (Accuracy): Accuracy = 77,93%.**

**Độ chính xác giảm xuống còn 78% so với 82% ở mô hình Logistic.**

Precision:

Lớp 0: 87%

Lớp 1: 57%

Mô hình vẫn dự đoán tốt khách hàng ở lại (lớp 0). Mô hình đã có sự tiến bộ khi phát hiện được nhiều khách hàng rời bỏ dịch vụ hơn (ở biểu đồ Confusion Matrix). Vì vậy nên độ chính xác tổng thể đã giảm, đây vẫn là tín hiệu khả quan đối với bộ dữ liệu.

* 1. Kết luận

-Khách hàng khi mới vừa sử dụng dịch vụ thường dễ rời bỏ

Phân phối dựa vào thời gian sử dụng dịch vụ ‘tenure’ cho thấy nhiều khách hàng sẽ rời bỏ sau tháng đầu tiên

Giải pháp: Tạo ra các chương trình khuyến mãi, ưu đãi, ví dụ như mua 1 năm giảm giá theo phần trăm,…

-Khách hàng có hợp đồng theo tháng (Month-to-month) có tỷ lệ rời bỏ cao nhất

Giải pháp: Khuyến khích ký hợp đồng dài hạn, tăng cường quảng bá gói combo

**Kết luận:** Việc giữ chân khách hàng là một điều rất quan trọng, doanh nghiệp cần triển khai các hình thức marketing. Tăng cường chăm sóc khách hàng để làm hài lòng các khách hàng cũ.

# XÂY DỰNG MÔ HÌNH DỰA TRÊN PCA

* 1. Giới thiệu bộ dữ liệu

Bộ dữ liệu được thu thập từ trang Kaggle – Một nền tảng thi đấu về khoa học dữ liệu và cộng đồng trực tuyến lớn nhất thế giới, thuộc sở hữu của Google LLC, cung cấp các công cụ và tài nguyên mạnh mẽ để giúp người dùng đạt được mục tiêu trong lĩnh vực khoa học dữ liệu và học máy. Bộ dữ liệu này có tên là “Diamond Online Marketplace” , chứa thông tin chi tiết về hơn 6485 viên kim cương, bao gồm các thông số vật lý (như chiều dài, chiều rộng, chiều cao và trọng lượng carat), đặc điểm chất lượng (như cắt, màu sắc, độ tinh khiết và phát quang) và giá cả. Nó cũng bao gồm các thuộc tính như loại chứng nhận, tỷ lệ và tính đối xứng, làm cho nó trở thành một tài nguyên tuyệt vời để phân tích cách các yếu tố khác nhau ảnh hưởng đến giá kim cương.

Bộ dữ liệu bao gồm 6485 quan trắc và gồm 18 biến sau:

Shape: Hình dạng hình học của viên kim cương.

Cut: Đánh giá chất lượng của cách cắt kim cương.

Color: Đánh giá màu sắc của kim cương từ D (không màu) đến Z.

Clarity: Đánh giá độ trong dựa trên các khuyết điểm.

Carat Weight: Trọng lượng của viên kim cương tính bằng cara.

Length/Width Ratio: Tỷ lệ giữa chiều dài và chiều rộng.

Depth %: Độ sâu của kim cương tính theo phần trăm chiều rộng của nó.

Table %: Chiều rộng của mặt trên tính theo phần trăm

Polish: Chất lượng hoàn thiện bề mặt của kim cương.

Symmetry: Độ chính xác của hình dạng viên kim cương.

Girdle: Độ dày của cạnh viên kim cương.

Culet: Kích thước của mặt đáy.

Length: Chiều dài của viên kim cương tính bằng milimét.

Width: Chiều rộng của viên kim cương tính bằng milimét

Height: Chiều cao của viên kim cương tính bằng milimét.

Price: Giá của viên kim cương tính bằng đô la Mỹ ($).

Type: Loại chứng nhận hoặc nguồn gốc của viên kim cương.

Fluorescence: Mức độ huỳnh quang UV của kim cương



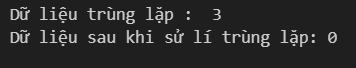
* 1. Tiền xử lý dữ liệu
     1. Dữ liệu bị trùng lặp

Trong quá trình làm sạch dữ liệu của tập diamond, một bước quan trọng là xác định và loại bỏ các dòng dữ liệu trùng lặp để đảm bảo sự chính xác và nhất quán trong quá trình phân tích.

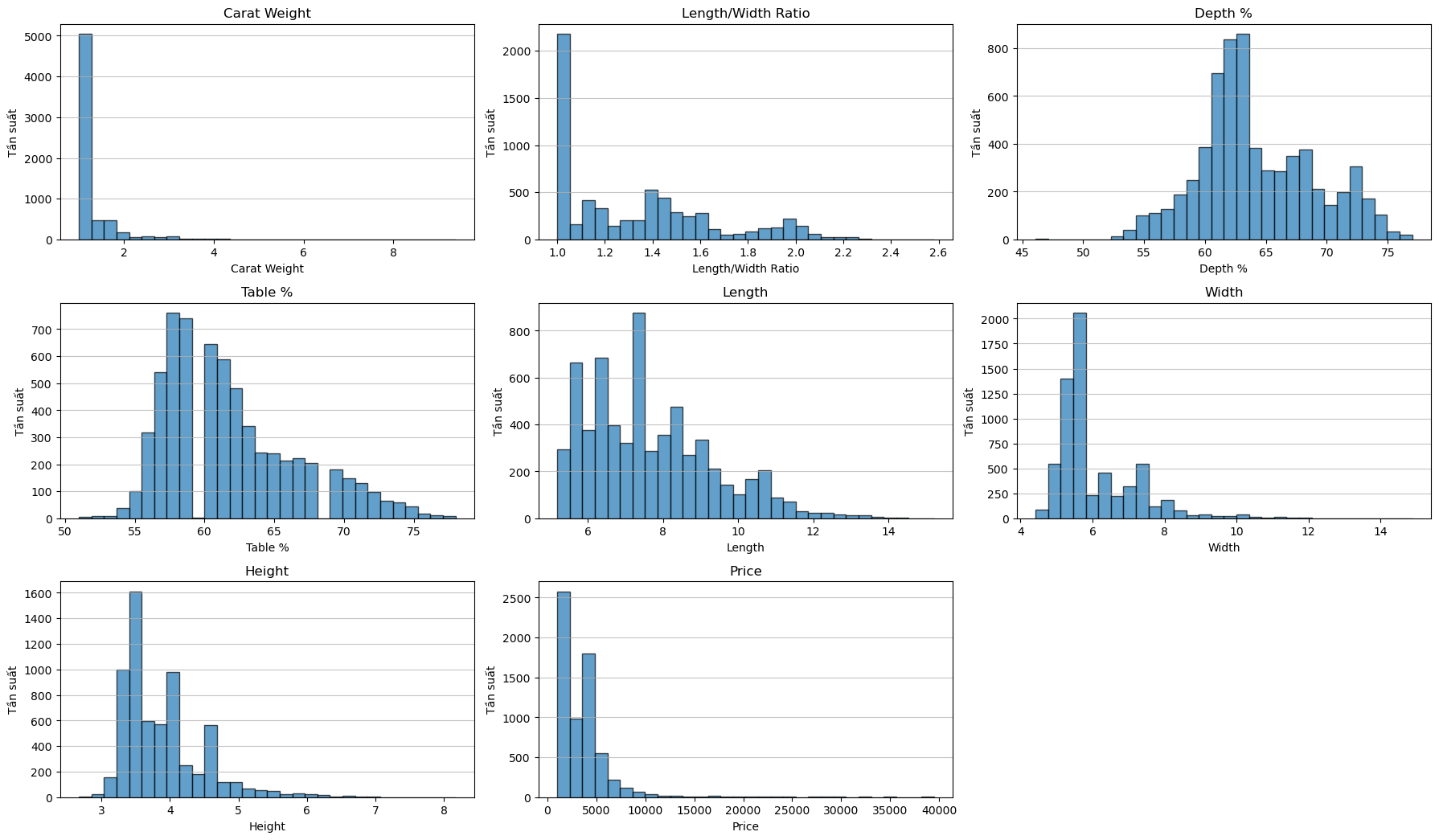
Đầu tiên, tập dữ liệu phát hiện có 3 dòng dữ liệu trùng lặp.

Tiếp theo, các dòng trùng lặp được loại bỏ bằng cách giữ nguyên dòng đầu tiên trong danh sách các dòng trùng lặp và loại trừ các dòng còn lại. Sau khi hoàn thành bước trên, kiểm tra lại cho thấy không còn bất cứ dòng dữ liệu trùng lặp nào trong tập dữ liệu. Kích thước cuối cùng của tập dữ liệu là 6482 dòng và 18 cột, phản ánh chính xác số lượng dữ liệu hợp lệ sau khi làm sạch.

Quá trình loại trừ dữ liệu trùng lặp là một bước quan trọng nhằm cải thiện chất lượng và độ tin cậy của dữ liệu đầu ra, qua đó cho phép mô hình học máy hoặc các phép phân tích kế tiếp đưa ra kết luận chuẩn xác và hợp lý hơn.



* + 1. Phân phối của các biến



Hình ảnh bao gồm 8 biểu đồ histogram thể hiện phân phối của các biến số quan trọng trong tập dữ liệu kim cương (diamonds dataset). Các biểu đồ giúp trực quan hóa tần suất xuất hiện của từng giá trị trong các cột dữ liệu dạng số, bao gồm: Carat Weight, Length/Width Ratio, Depth %, Table%, Length, Width, Height, Price.

Phân tích chi tiết từng biểu đồ:

Carat Weight: Phân phối lệch phải (Right-skewed): Phần lớn kim cương có trọng lượng nhỏ, tập trung quanh mức 1 - 2 carat. Các viên kim cương có trọng lượng trên 3 carat chiếm tần suất rất thấp, thể hiện rõ sự chênh lệch giữa kim cương phổ thông và cao cấp.

Length/Width Ratio: Tập trung ở mức 1.0 - 1.5: Điều này cho thấy phần lớn kim cương có hình dạng tương đối cân đối, gần với hình vuông hoặc chữ nhật. Tỷ lệ lớn (> 2.0) ít xuất hiện: Điều này cho thấy kim cương có hình dạng thon dài hoặc bất thường là hiếm

Depth %: Độ sâu tập trung quanh mức 60 - 65%. Đây là tỷ lệ phổ biến, đảm bảo kim cương có độ sáng tối ưu.

Table % : Phân phối lệch phải nhẹ (Slight Right-skewed): Phần lớn kim cương có tỷ lệ bề mặt bàn trong khoảng 55 - 65%.

Length :Tập trung quanh 6 - 9 mm: Đây là kích thước phổ biến của kim cương trong tập dữ liệu. Chiều dài lớn (> 10 mm) hiếm gặp: Điều này cho thấy kim cương dài là ngoại lệ, có thể thuộc dòng sản phẩm cao cấp hoặc thiết kế đặc biệt.

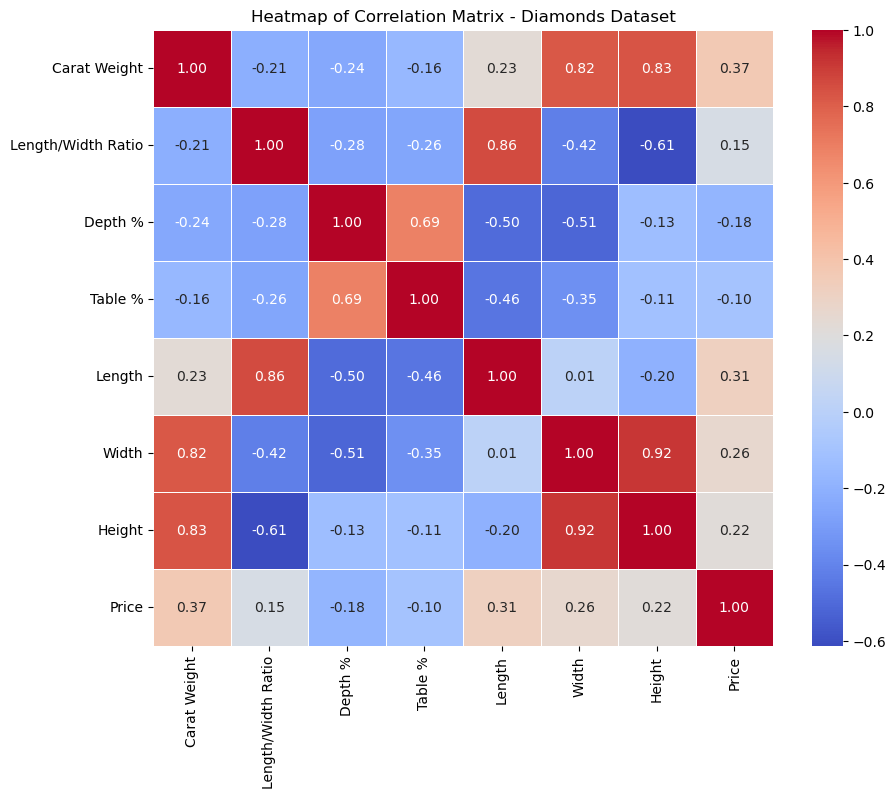
Width: Tập trung ở mức 5 - 7 mm, phần lớn kim cương có chiều rộng trong khoảng này, phản ánh sự đồng đều về kích thước. Các viên kim cương có chiều rộng lớn hơn 10 mm xuất hiện rất ít.

Height: Tập trung trong khoảng 3 - 4.5 mm: Đây là chiều cao phổ biến của kim cương. Chiều cao lớn (> 5 mm) là ngoại lệ, thể hiện các viên kim cương đặc biệt có độ dày lớn.

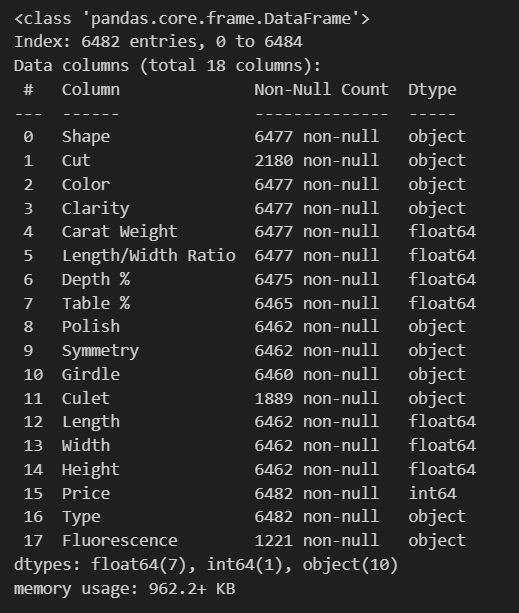
Price: Phân phối rất lệch phải (Highly Right-skewed): Hầu hết kim cương có giá dưới 5000 USD, với số ít viên có giá trị lên đến 40,000 USD.

Kết luận:

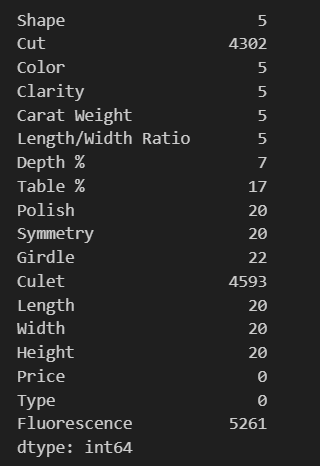
Việc phân tích histogram cho thấy sự phân bố của các biến trong tập dữ liệu không đồng đều, đặc biệt ở các biến như giá và trọng lượng carat. Điều này là cơ sở quan trọng cho các bước phân tích tiếp theo, giúp xây dựng mô hình dự đoán giá trị kim cương và đưa ra các quyết định phân loại dựa trên đặc điểm vật lý.



* Kiểu dữ liệu của các biến :



* + 1. Xử lý giá trị thiếu (missing value)



Một số cột có lượng lớn giá trị thiếu như:

+ Cut (4302 giá trị thiếu), Culet (4593), Fluorescence (5261).

+ Hai cột Price và Type hoàn toàn không có dữ liệu (0 giá trị).

+ Các cột khác như Shape, Color, Clarity, Carat Weight chỉ thiếu 5 giá trị, mức độ ảnh hưởng nhỏ.

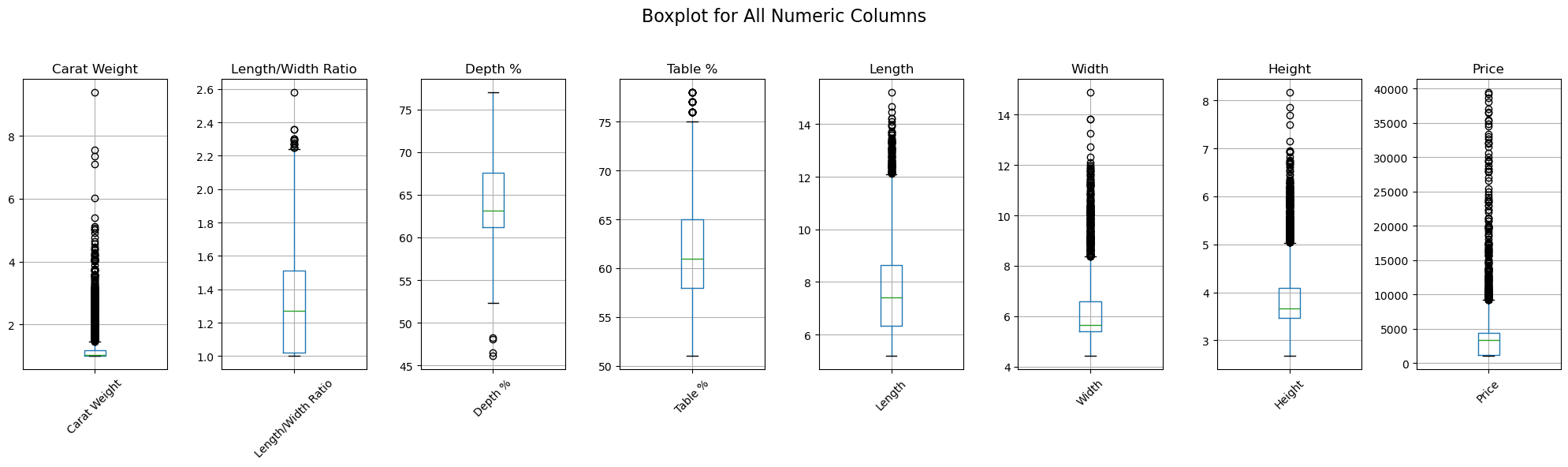
Để xử lý giá trị thiếu trong tập dữ liệu, phương pháp điền (imputation) được áp dụng nhằm đảm bảo tính đầy đủ và nhất quán trước khi tiến hành phân tích hoặc xây dựng mô hình. Cách tiếp cận được chia thành hai hướng phù hợp với từng loại dữ liệu:

+ Đối với các cột dữ liệu số (numerical):Các giá trị bị thiếu được điền bằng giá trị trung vị (median). Việc sử dụng trung vị giúp giảm thiểu tác động của các giá trị ngoại lai và đảm bảo sự phân phối dữ liệu không bị lệch quá nhiều. Phương pháp này đặc biệt hữu ích khi dữ liệu có nhiều sự chênh lệch hoặc không phân phối chuẩn.

+ Đối với các cột dữ liệu phân loại (categorical): Các cột thuộc loại phân loại được xử lý bằng cách điền vào các giá trị bị thiếu bằng giá trị xuất hiện nhiều nhất (mode). Điều này giúp giữ cho tập dữ liệu nhất quán với xu hướng chung của phần lớn dữ liệu hiện có, từ đó hạn chế sai lệch và bảo toàn tính chính xác của các đặc điểm phân loại.

Dữ liệu sau khi được tách thành hai nhóm – cột số và cột phân loại – sẽ được xử lý theo từng chiến lược phù hợp. Việc áp dụng các phương pháp điền giá trị thiếu này giúp đảm bảo tập dữ liệu đầy đủ, hạn chế mất mát thông tin và tăng cường chất lượng cho các bước phân tích tiếp theo. Đây là bước tiền xử lý quan trọng trong việc xây dựng các mô hình dự đoán và phân tích dữ liệu một cách hiệu quả và chính xác.

* + 1. Phát hiện giá trị ngoại lai (outlier)



* Để đảm bảo dữ liệu ổn định và giảm thiểu sự ảnh hưởng của các giá trị bất thường, phương pháp IQR (Interquartile Range) đã được sử dụng để loại bỏ outliers. Các giá trị được xem là ngoại lai và bị loại khỏi dữ liệu khi nằm ngoài khoảng:

[Q1 – 1.5 \* IQR; Q3 + 1.5 \* IQR]

* Kết quả phát hiện outliers:
* Carat Weight: 1336 outliers
* Length/Width Ratio: 15 outliers
* Depth %: 4 outliers
* Table %: 38 outliers
* Length: 85 outliers
* Width: 250 outliers
* Height: 302 outliers
* Price: 216 outliers
* Xử lý outlier:
* Để xử lý các giá trị ngoại lai, phương pháp Winsorization được áp dụng. Đây là kỹ thuật thay thế các giá trị outliers bằng các giá trị giới hạn (lower bound hoặc upper bound) của khoảng IQR.
* Nguyên tắc:

+ Nếu một giá trị nhỏ hơn ngưỡng dưới (Q1 - 1.5 \* IQR), giá trị đó được thay thế bằng ngưỡng dưới.

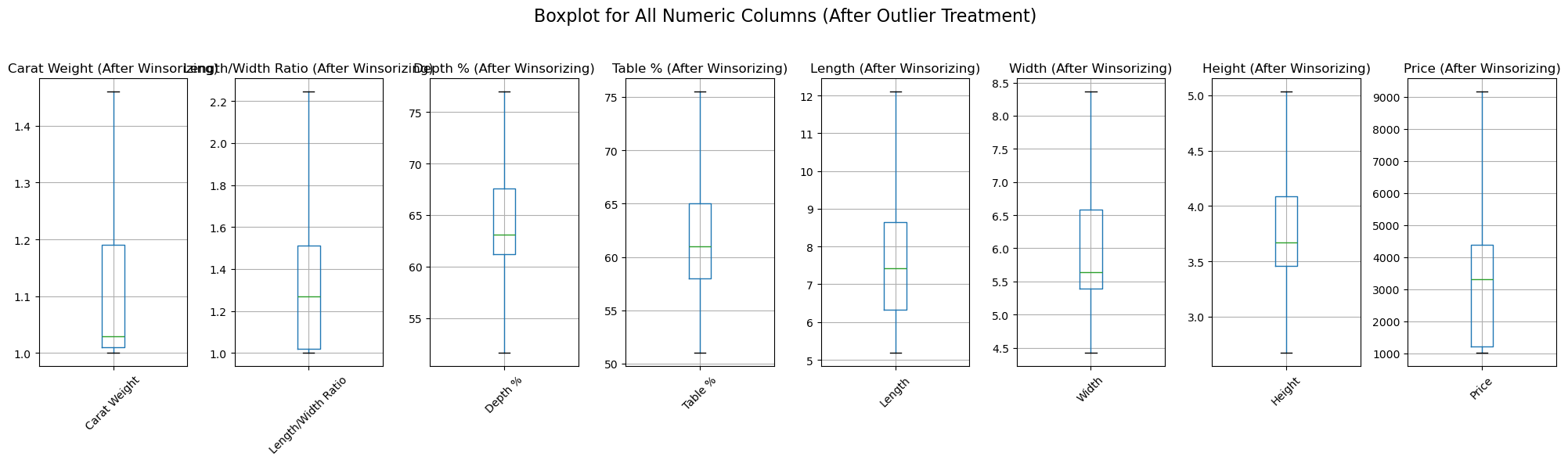
+ Nếu một giá trị lớn hơn ngưỡng trên (Q3 + 1.5 \* IQR), giá trị đó được thay thế bằng ngưỡng trên.

* Kết quả:

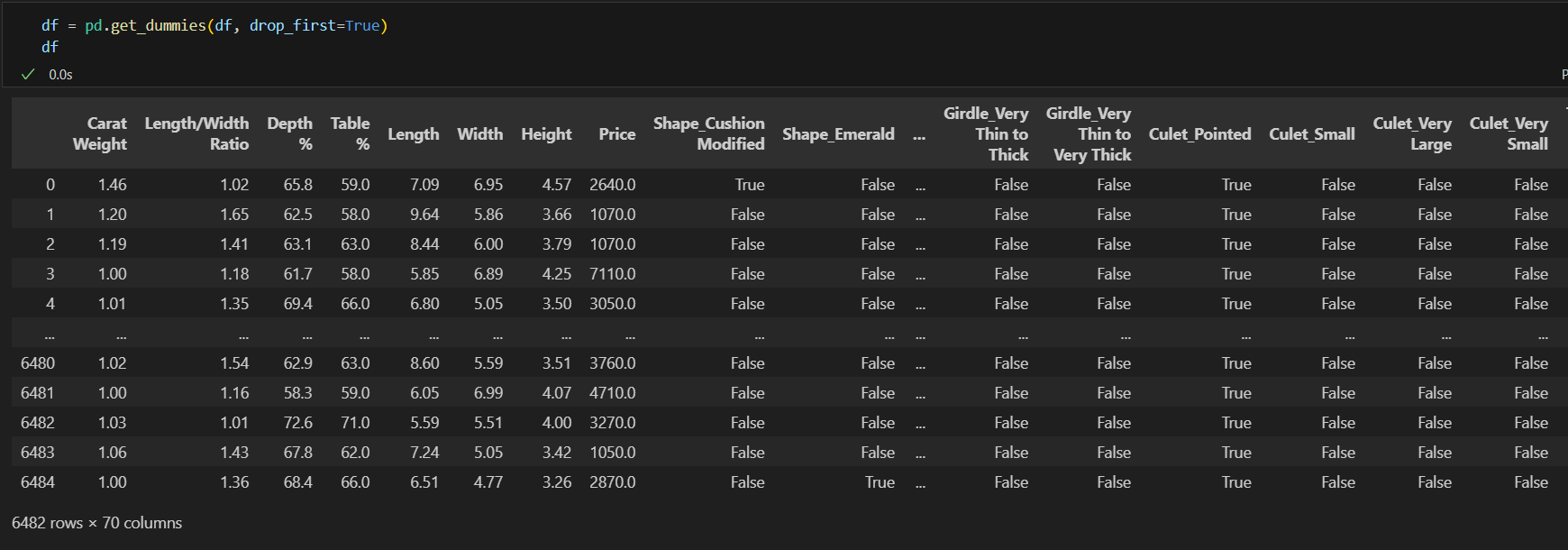
Các outliers được xử lý mà không loại bỏ dữ liệu, đảm bảo giữ lại lượng thông tin tối đa cho mô hình.

* Kết luận :

Quá trình xác định và xử lý outliers là một bước quan trọng đối với việc xử lý dữ liệu. Việc loại trừ hoặc thay thế các dữ liệu ngoại lai giúp mô hình học máy vận hành tốt hơn, giảm thiểu tình trạng overfitting hoặc làm sai lệch kết quả dự đoán. Phương pháp Winsorization được sử dụng vì khả năng xử lý outliers nhanh chóng mà không làm mất mát dữ liệu quan trọng, nhờ vậy nâng cao hiệu suất và độ tin cậy của mô hình dự đoán.



* + 1. Xử lý dữ liệu phân loại bằng phương pháp One-Hot Encoding



Các biến phân loại như Shape, Cut, Color, Clarity, Polish, Symmetry, Girdle, Culet, Type, Fluorescence không thể trực tiếp đưa vào mô hình học máy, do chúng là dữ liệu dạng văn bản hoặc danh mục.

One-Hot Encoding tạo ra các cột nhị phân (binary), đại diện cho từng giá trị duy nhất trong biến phân loại, giúp mô hình hiểu rõ hơn về sự khác biệt giữa các hạng mục

* Thực hiện One-Hot Encoding
* Quá trình này tạo ra các cột mới tương ứng với mỗi giá trị trong biến phân loại
* Drop first=True được sử dụng để giảm thiểu hiện tượng đa cộng tuyến (multicollinearity) bằng cách loại bỏ một cột đại diện cho mỗi biến. Điều này giúp giảm thiểu tổng số cột, cải thiện hiệu suất mô hình và loại bỏ thông tin dư thừa trong dữ liệu.
* Kết quả:
* Sau khi mã hóa , tập dữ liệu có 70 cột với 6482 dòng.
* Các cột phân loại được mã hóa thành các cột mới:

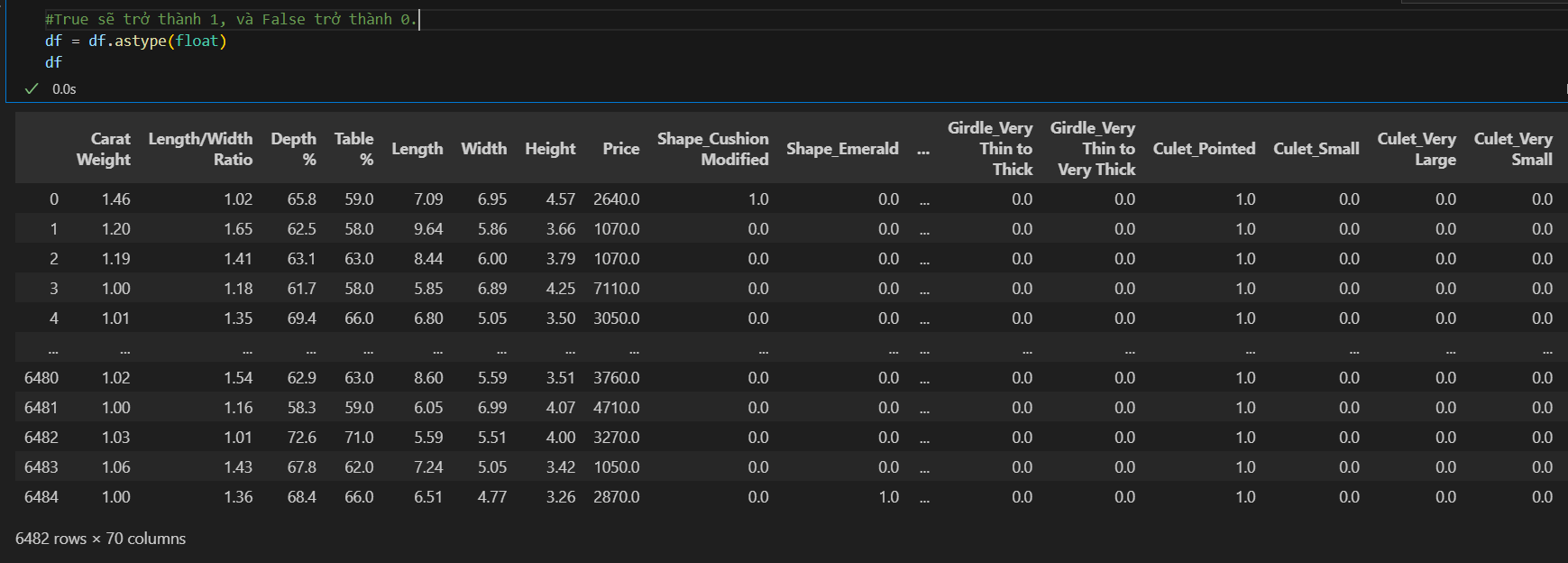
+ Shape\_Cushion Modified, Shape\_Emerald đại diện cho Shape

+ Culet\_Pointed, Culet\_Small, Culet\_Very Large đại diện cho Culet

+ Type\_GIA Lab-Grown , Type\_IGI Lab-Grown đại diện cho Type

+ Fluorescence\_Medium, Fluorescence\_Strong đại diện cho Fluorescence,…

* + 1. Chuyển đổi dữ liệu phân loại về thành dạng số (Binary Encoding)



Quy trình và cách hoạt động:

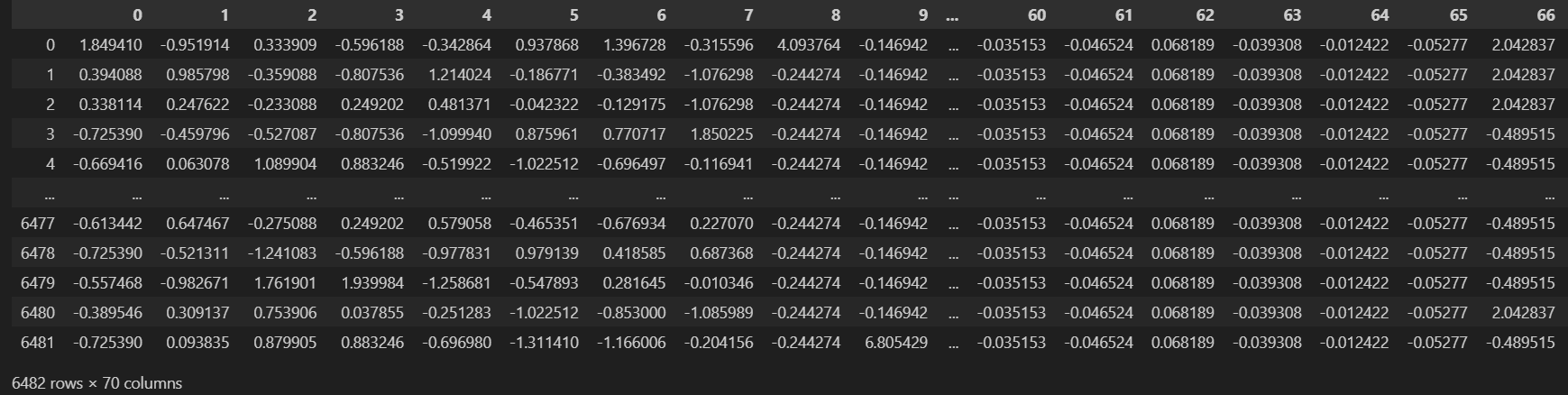
Bước chuyển đổi được thực hiện bằng phương pháp astype(float), trong đó:

+ True được chuyển thành 1.0

+ False được chuyển thành 0.0

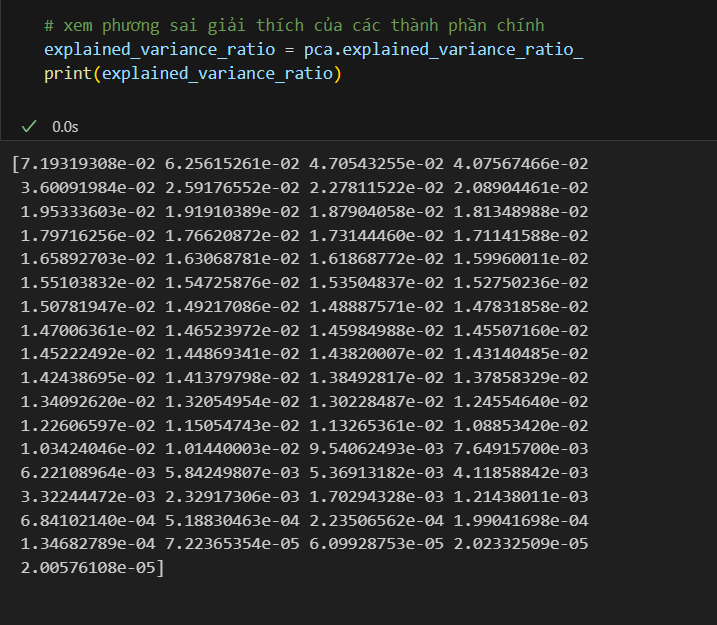
Quá trình này được áp dụng cho toàn bộ các cột dữ liệu phân loại đã được One-Hot Encoding trước đó.

* + 1. Chuẩn Hóa Dữ Liệu bằng StandardScaler

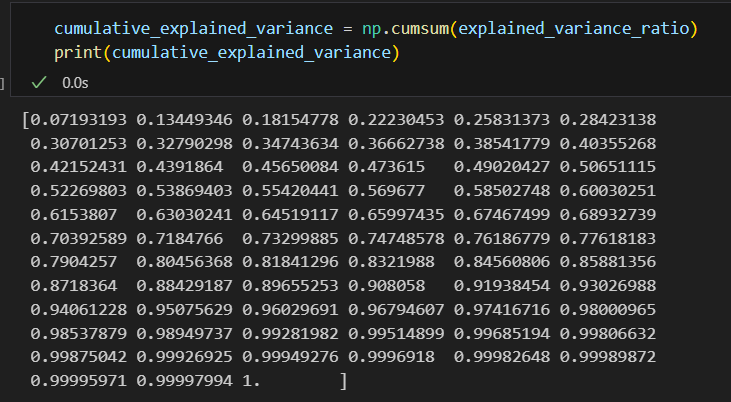


Chuẩn hóa giúp đưa tất cả các đặc trưng (features) của dữ liệu về cùng một quy mô, tránh sự chênh lệch lớn giữa các cột. Điều này giúp mô hình học máy hội tụ nhanh hơn và cải thiện độ chính xác trong quá trình huấn luyện

* + 1. Xây dựng mô hình PCA



Tỷ lệ phương sai giải thích (explained variance ratio) của các thành phần chính sau khi áp dụng PCA (Phân tích thành phần chính). Mỗi giá trị đại diện cho mức độ đóng góp của một thành phần vào tổng phương sai của dữ liệu. Các thành phần đầu tiên giải thích phần lớn sự biến thiên, trong khi các thành phần sau có tỷ lệ phương sai nhỏ dần. Điều này cho phép lựa chọn giữ lại các thành phần chính có ý nghĩa, giúp giảm chiều dữ liệu mà vẫn bảo toàn được phần lớn thông tin, tối ưu hóa mô hình và loại bỏ nhiễu không cần thiết.



Tích lũy phương sai giải thích (cumulative explained variance) sau khi áp dụng PCA (Phân tích thành phần chính). Mỗi giá trị trong danh sách biểu thị tỷ lệ phương sai cộng dồn của các thành phần chính, cho thấy tổng mức độ thông tin mà các thành phần đầu tiên giải thích được.

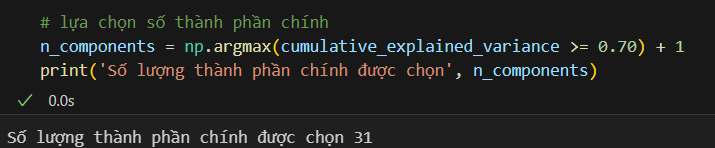
Kết quả cho thấy:

+ 5 thành phần đầu tiên giải thích khoảng 30% phương sai của dữ liệu.

+ Đến thành phần thứ 20, tỷ lệ phương sai giải thích đạt khoảng 90%.

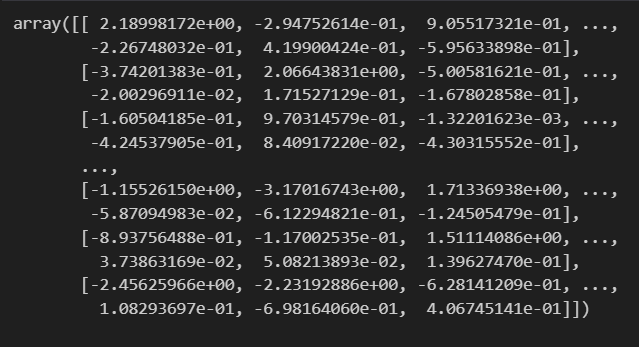
+ Từ thành phần thứ 30 trở đi, phương sai cộng dồn gần như đạt 100%, cho thấy hầu hết thông tin đã được giữ lại.

Việc tính toán phương sai tích lũy giúp xác định số lượng thành phần chính cần thiết để giữ lại phần lớn thông tin của dữ liệu, từ đó có thể giảm chiều dữ liệu mà không làm mất quá nhiều thông tin quan trọng. Điều này hỗ trợ trong việc tăng hiệu suất mô hình và giảm độ phức tạp của dữ liệu.



Lựa chọn số lượng thành phần chính (principal components) trong PCA dựa trên phương sai tích lũy. Là giữ lại các thành phần chính sao cho tổng phương sai giải thích đạt ít nhất 70%.

Kết quả cho thấy 31 thành phần chính được chọn để đảm bảo dữ liệu giữ lại đủ thông tin cần thiết, đồng thời giảm chiều dữ liệu mà không làm mất quá nhiều phương sai. Việc lựa chọn số lượng thành phần theo ngưỡng này giúp tối ưu hóa mô hình, tăng hiệu suất và giảm độ phức tạp.

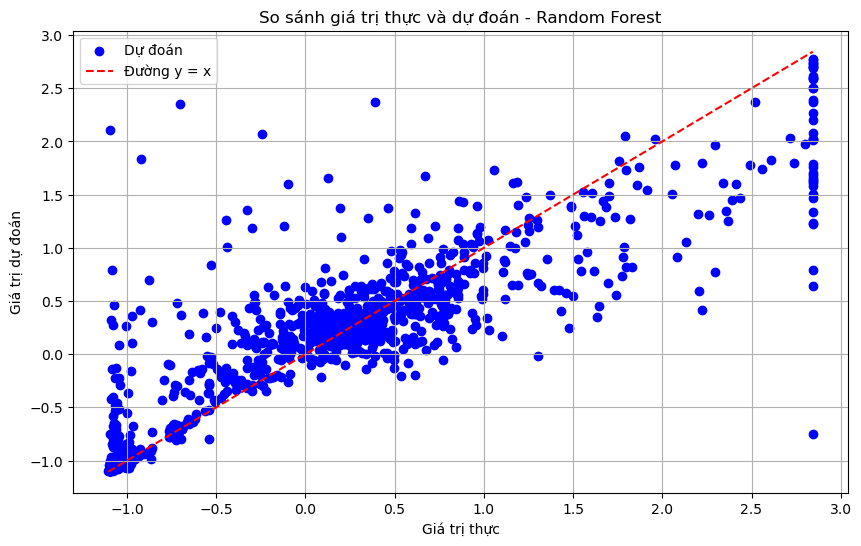
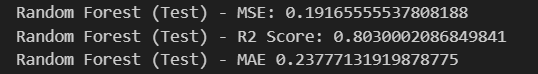


* 1. Random Forest Regressor:

Random Forest là một thuật toán ensemble learning (học tập tập hợp) dựa trên việc kết hợp nhiều cây quyết định (Decision Trees).

Random Forest Regressor là phiên bản áp dụng cho các bài toán hồi quy (regression), được sử dụng để dự đoán các giá trị liên tục.

Mô hình hoạt động bằng cách xây dựng nhiều cây quyết định độc lập và lấy trung bình các dự đoán từ tất cả các cây. Điều này giúp tăng độ chính xác và giảm overfitting.

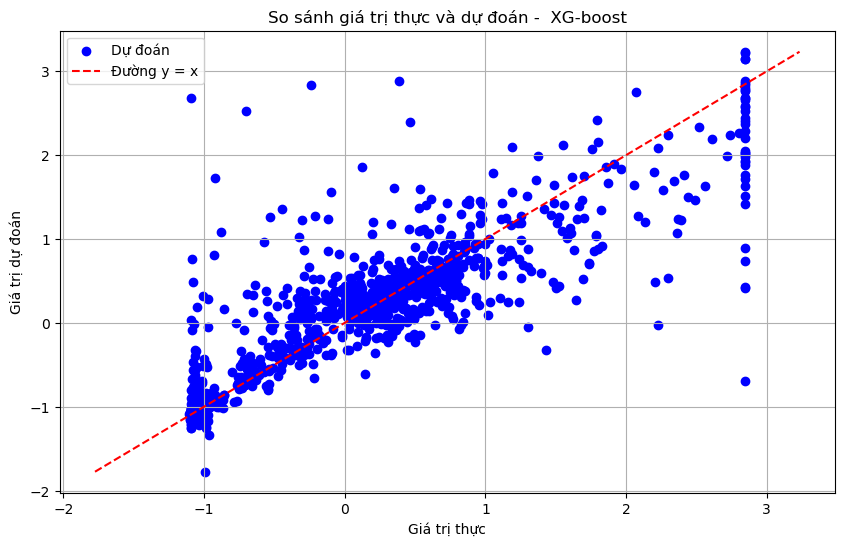
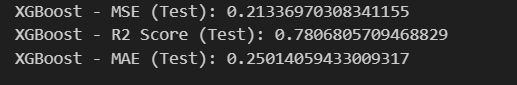


* 1. Mô hình XGBoost Regressor:

XGBoost (Extreme Gradient Boosting) là một thuật toán học máy dựa trên gradient boosting.

Được thiết kế để tối ưu hóa cả về tốc độ và hiệu suất, XGBoost là một trong những mô hình phổ biến nhất trong các cuộc thi Machine Learning và Kaggle.

XGBoost Regressor là phiên bản dành cho các bài toán hồi quy (regression), giúp dự đoán giá trị liên tục, chẳng hạn như dự đoán giá nhà, doanh thu hoặc giá kim cương.

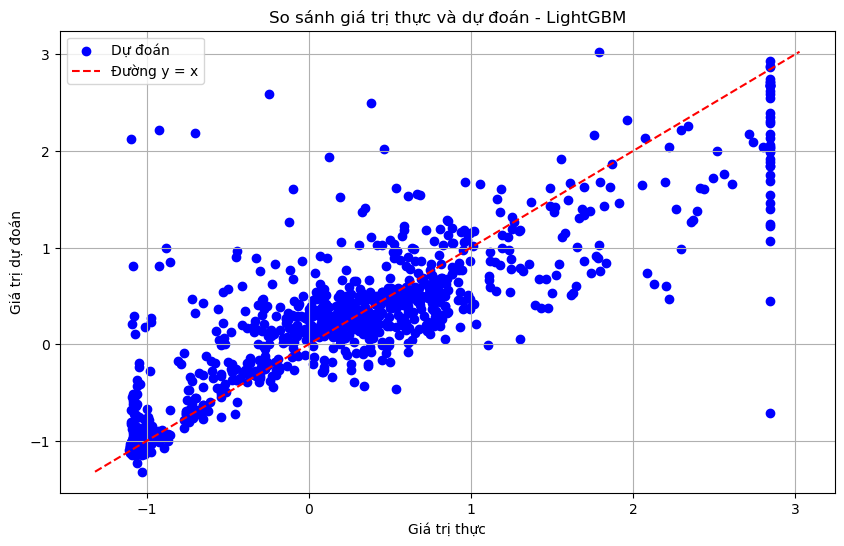
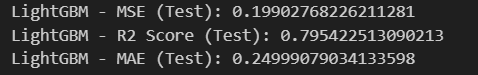


* 1. Mô hình LightGBM Regressor:

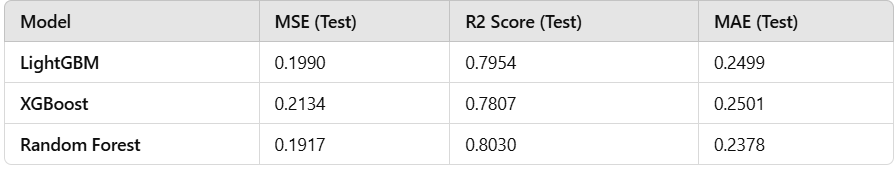
LightGBM (Light Gradient Boosting Machine) là một thuật toán học máy thuộc nhóm ensemble learning, dựa trên phương pháp gradient boosting.

LightGBM được phát triển bởi Microsoft và nổi tiếng với khả năng huấn luyện nhanh, hiệu suất cao, và khả năng mở rộng tốt trên các tập dữ liệu lớn.

LightGBM Regressor là phiên bản của LightGBM áp dụng cho các bài toán hồi quy (regression), dự đoán các giá trị liên tục.



* So sánh 3 model :



Nhận xét:

* Random Forest có MSE thấp nhất (0.1917) và R2 cao nhất (0.8030), cho thấy mô hình này dự đoán tốt hơn các mô hình khác.
* LightGBM có kết quả gần tương đương với Random Forest nhưng vẫn kém hơn một chút về MSE và R2.
* XGBoost có hiệu suất thấp hơn so với hai mô hình còn lại, đặc biệt về MSE và R2.

Kết luận : Random Forest là lựa chọn phù hợp nhất cho bài toán này, nhờ vào hiệu suất cao và khả năng tổng quát tốt.

# DỰ ĐOÁN CHI PHÍ BẢO HIỂM HẰNG NĂM CHO XE

* 1. Giới thiệu dữ liệu

Bộ dữ liệu này được thu thập từ trang Kaggle – một nền tảng cộng đồng dành cho khoa học dữ liệu và học máy. Đây là bộ dữ liệu về **chi phí bảo hiểm ô tô** (Automobile Insurance Costs), trích xuất từ các nguồn thống kê thực tế về xe hơi và các yếu tố liên quan đến chi phí bảo hiểm. Bộ dữ liệu giúp phân tích các yếu tố ảnh hưởng đến mức phí bảo hiểm của từng loại xe.

Bộ dữ liệu bao gồm thông tin chi tiết về các đặc điểm của xe và chủ sở hữu, với tổng cộng khoảng 200 quan sát, bao gồm các biến chính:

make: Hãng sản xuất xe.

fuel-type: Loại nhiên liệu (xăng/diesel).

body-style: Kiểu dáng xe (sedan, hatchback, SUV, v.v.).

horsepower: Công suất của xe (mã lực).

price: Giá bán của xe.

city-mpg: Mức tiêu thụ nhiên liệu trong thành phố (dặm/gallon).

highway-mpg: Mức tiêu thụ nhiên liệu trên đường cao tốc.

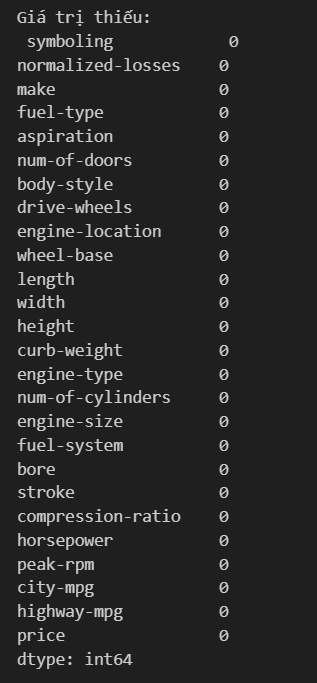
insurance-cost: Chi phí bảo hiểm hàng năm cho xe.

* 1. Tiền xử lý dữ liệu
     1. Kiểm tra dữ liệu



Bảng dữ liệu ô tô cung cấp thông tin về thông số kỹ thuật và giá cả, với các biến như hãng sản xuất, loại nhiên liệu, dung tích động cơ và giá xe. Ký tự ? trong cột như tổn thất chuẩn hóa (normalized-losses) cho thấy dữ liệu thiếu cần xử lý. Giá xe dao động từ 13,495 USD đến 26,625 USD, phản ánh sự khác biệt về cấu hình và hiệu suất giữa các mẫu xe. Sự chênh lệch lớn ở mã lực (horsepower) và tỷ số nén (compression-ratio) không chỉ giữa các hãng mà còn trong cùng thương hiệu như Volvo, cho thấy sự đa dạng về thiết kế và trang bị động cơ. Xử lý dữ liệu thiếu và phân tích các yếu tố như dung tích động cơ, loại xe là bước quan trọng để xây dựng mô hình dự đoán giá chính xác và ổn định.

* + 1. Xử lý dữ liệu thiếu

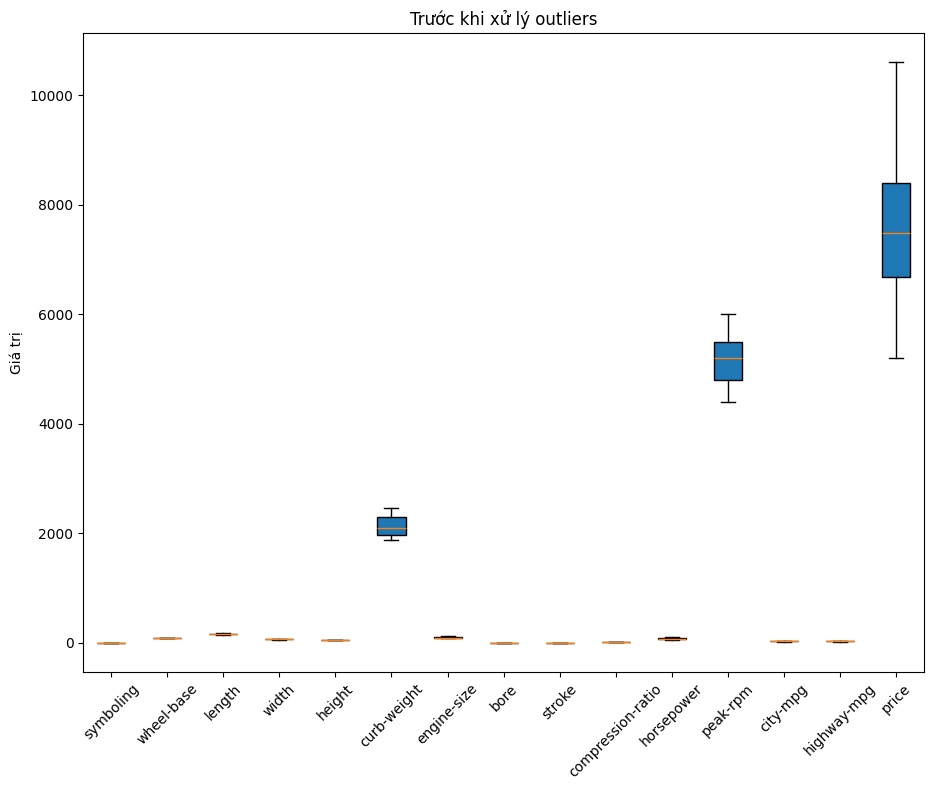
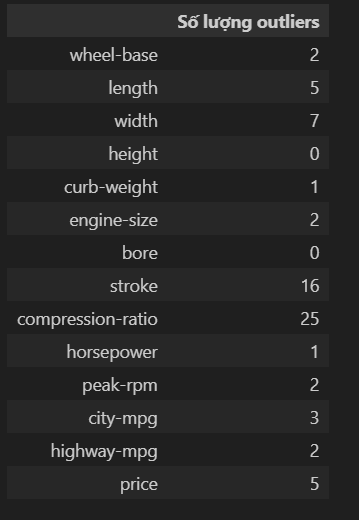


Không cần xử lý giá trị thiếu, giúp tiết kiệm thời gian và tránh các phương pháp ước tính hoặc điền dữ liệu có thể làm sai lệch phân tích.

Đảm bảo độ chính xác cao hơn cho mô hình vì dữ liệu đầy đủ, không có nguy cơ ảnh hưởng từ việc ước lượng giá trị bị thiếu.

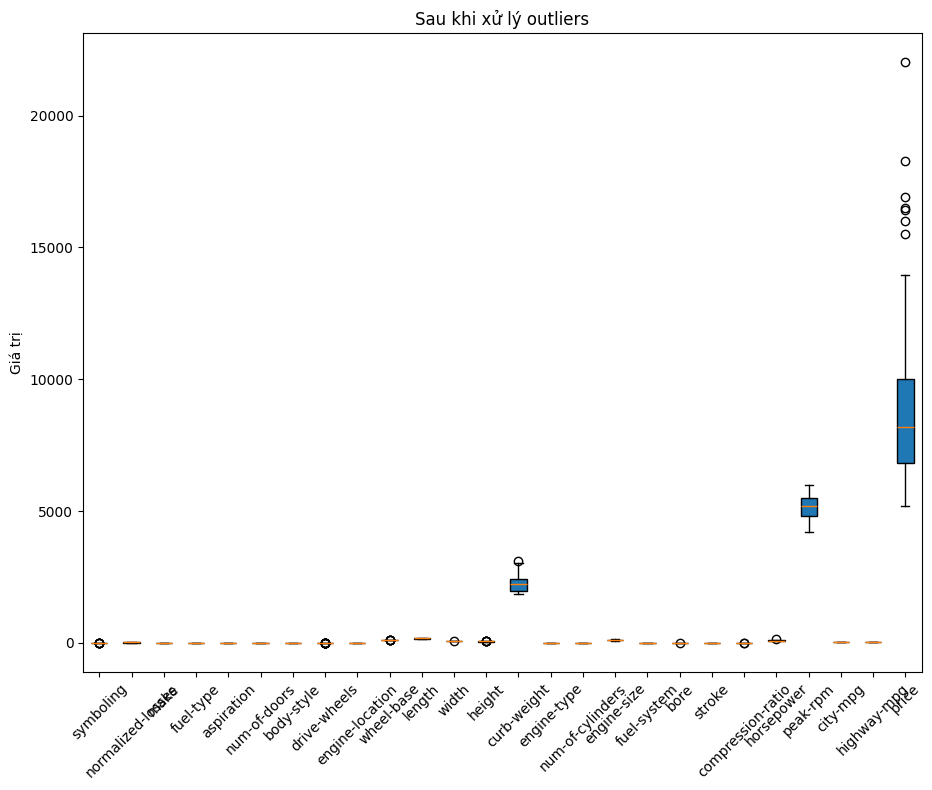
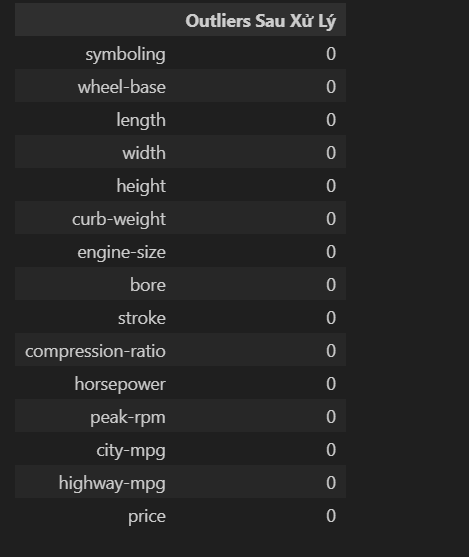
Xử lý sạch dữ liệu: Đã loại bỏ toàn bộ giá trị thiếu, đảm bảo dữ liệu không còn lỗi.

* + 1. Kiểm tra và xử lý outliers



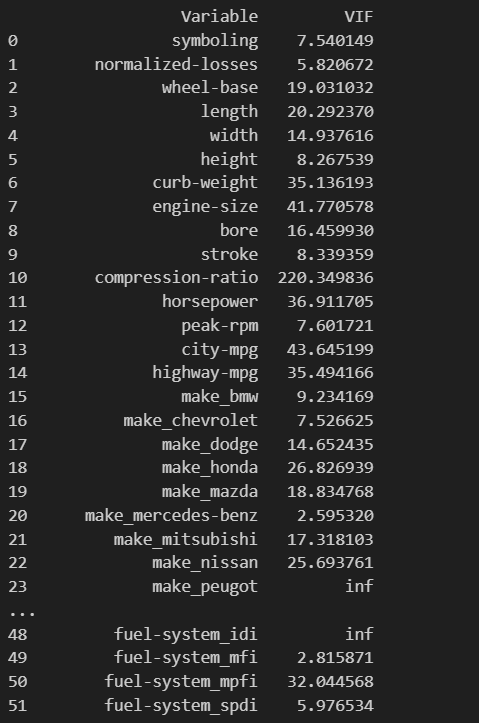
Bảng trên cho thấy compression-radio (25) và stoke (16) có nhiều outliers, phản ánh sự khác biệt giữa các mẫu xe do thiế kế động cơ đa dạng.

Height và bore không có outliers, cho thấy dữ liệu đồng nhất. Price có 5 outliers, cho thấy sự chênh lệch lớn về giá các mẫu xe. Việc xử lý là cần thiết để tránh ảnh hưởng đến mô hình.



Sau khi xử lý outliers của skole giảm từ 16 xuống 4 và price từ 5 xuống 2, giúp dữ liệu ổn định hơn. Tuy nhiên, compression-radio vẫn có 22 outliers, cho thấy sự phân tán lớn. Việc giảm outliers cải thiện dữ liệu nhưng loại bỏ quá nhiều có thể ảnh hưởng độ chính xác mô hình.

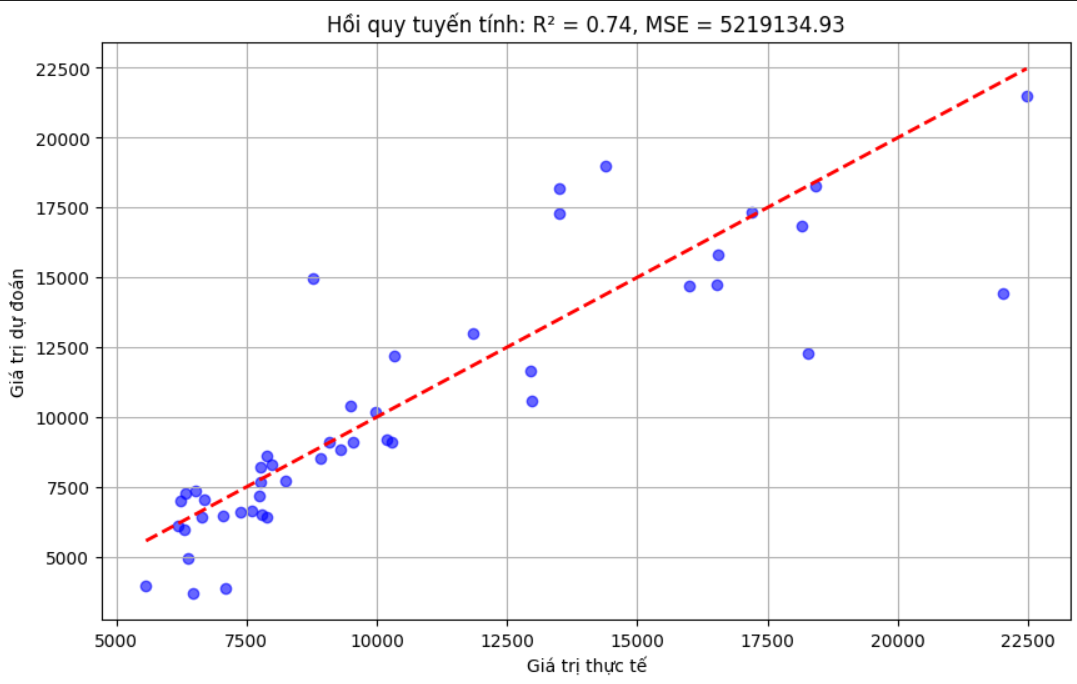
* + 1. Đa cộng tuyến



Phân tích VIF cho thấy compression-radio (220.35), city-mpg (43.65) và engine-size (41.77) có VIF cao, có thể làm giảm độ chính xác của mô hình.

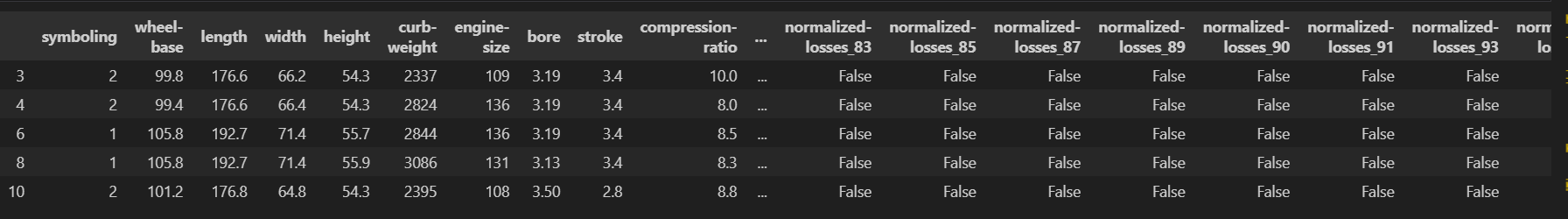
Ngược lại, symboling (7.54) và fuel-system\_mfi (2.82) có VIF thấp, cho thấy ít có đa cộng tuyến. Xử lý đa cộng tuyến bằng cách loại bỏ hoặc kết hợp các biến có VIF cao sẽ giúp cải thiện hiệu suất và độ chính xác của mô hình.

* + 1. Hồi quy tuyến tính

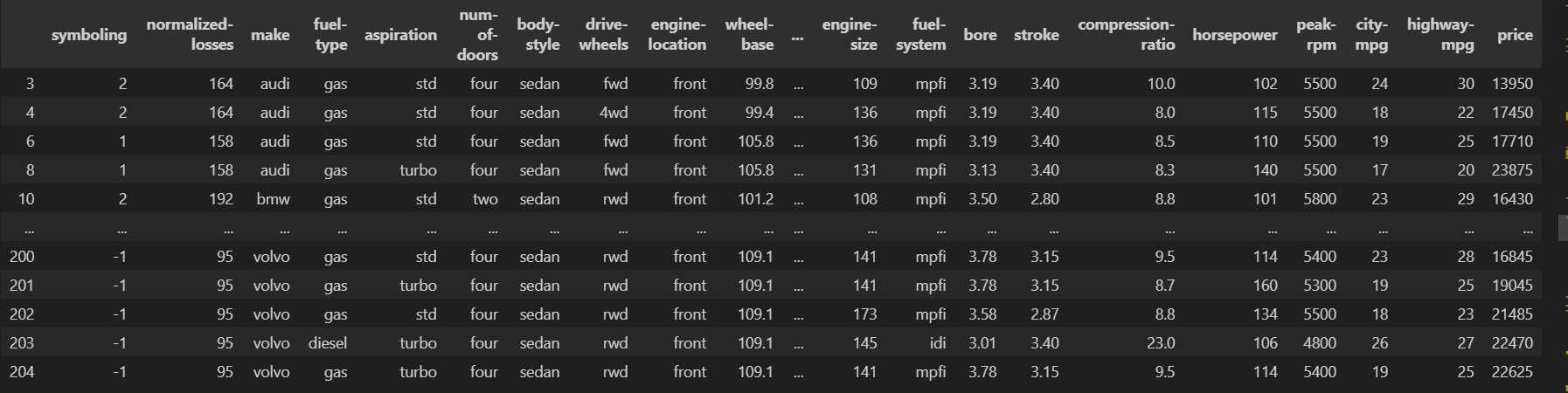


Mô hình hôi quy tuyến tính với R2 = 0,74 cho thấy mô hình giải thích được 74% phương sai, độ chính xác khá tốt. Tuy nhiên, MSE cao phản ánh sự chênh lệch giữa giá trị dự đoán và thực tế

Hầu hết điểm dữ liệu gần đường hồi quy nhưng vẫn có một số điểm lệch xa, cho thấy chưa hoàn toàn phù hợp. Loại bỏ outliers hoặc thêm biến mới có thể cải thiện độ chính xác.

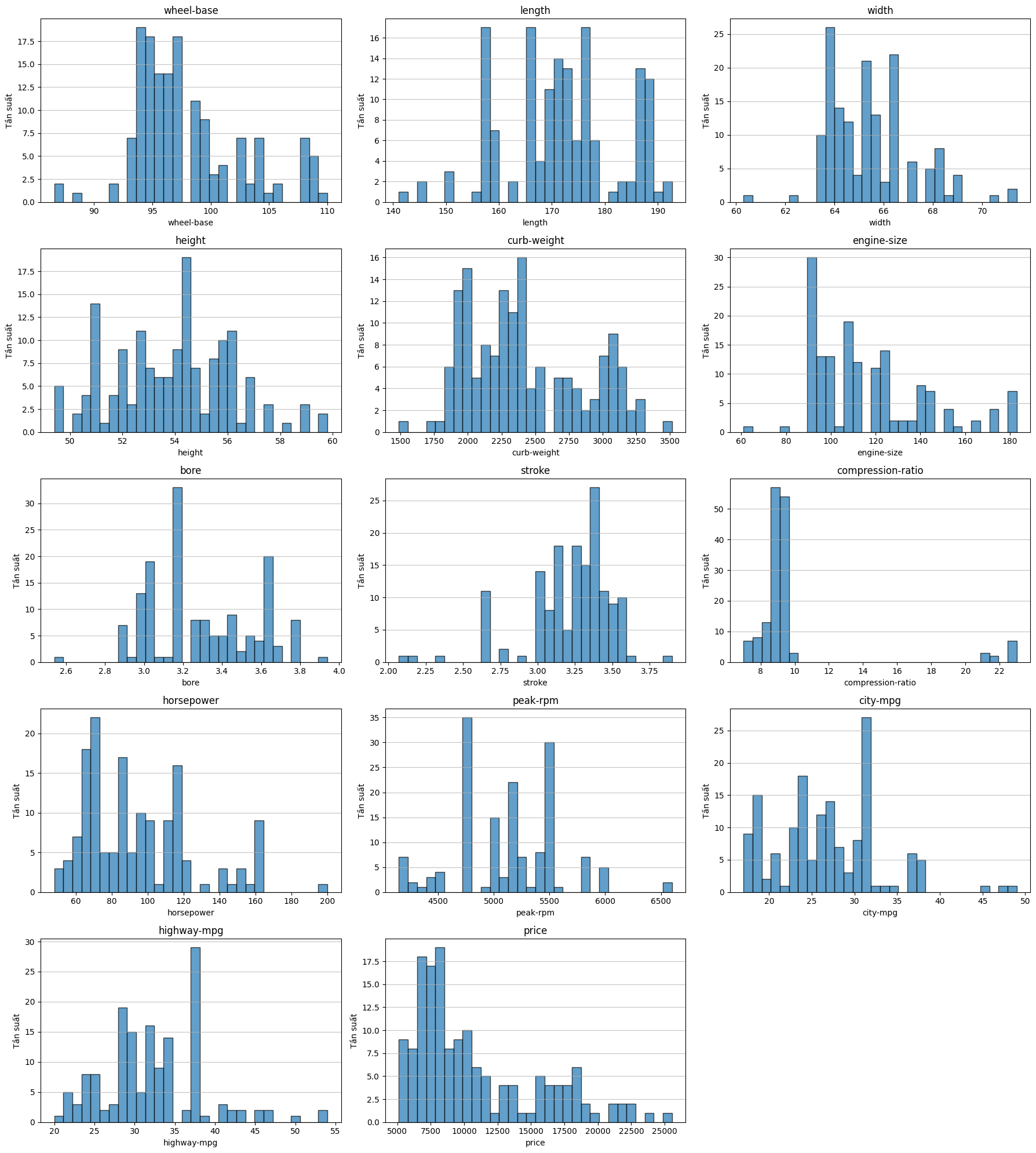


Bảng dữ liệu hiển thị thông tin kỹ thuật của xe như wheel-base, length, width, height, curb-weight, engine-size và compression-radio. Ngoài ra, còn có các cột đại diện cho normalized-losses dược biểu diễn dưới dạng các giá trị Boolean (True/False).



Dữ liệu trên cho thấy các cột đã được xử lý, trong đó các mục False. Do đó, không có thông tin mất cân bằng, giúp tăng độ chính xác của mô hình dự đoán. Quá trình này đóng vai trò quan trong trong việc giảm sai lệch và cải thiện hiệu suất của các thuật toán máy học.

* + 1. Biểu đồ biến liên tục



Biểu đồ cho thấy sự chênh lệch giữa các mẫu xe. Wheel-base, length, width phân bố cân đối, trong khi curb-weight và engine-size lệch phải, phản ánh phần lớn xe có trọng lượng và động cơ nhỏ.

Compression-ratio và price có nhiều ngoại lệ, thể hiện sự phân hóa giữa xe bình dân và cao cấp. Stroke, bore, height phân bố đều, còn peak-rpm và horsepower lệch phải, cho thấy một số xe hiệu suất cao.

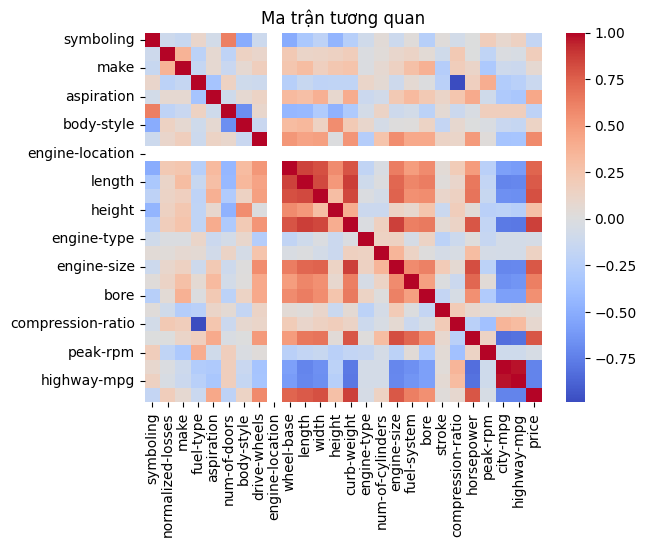
Engine-size tỉ lệ thuận với price, trong khi city-mpg và highway-mpg tập trung vào mức tiêu thụ trung bình.

* + 1. Biểu đồ phân phối của các cột số



Biểu đồ cho thấy sự phân bố đa dạng giữa các biến. Wheel-base, length, width phân bố lệch nhẹ, trong khi curb-weight và engine-size lệch phải, phản ánh chênh lệch giữa các xe nhẹ và nặng. Price và compression-radio có nhiều ngoại lệ, cho thấy sự phân hóa giữa xe bình dân và xe cao cấp. City-mpg và highway-mpg phân phối đều, phản ánh mức tiêu thụ nhiên liệu ổn định.

* + 1. Ma trận tương quan



Ma trận tương quan cho thấy engine-size, curb-weight, horsepower liên quan chặt chẽ đến price, trong kho city-mpg và highway-mpg có tương quan.

Compression-radio và peak-rpm ít ảnh hưởng đến giá. Wheel-base, length, width tỷ lệ thuận với curb-weight và engine-size. Ma trận giúp nhận diện yếu tố chính ảnh hưởng đến giá xe.