BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: PHÂN TÍCH DỮ LIỆU LỚN

ĐỀ TÀI

PHÂN TÍCH DỮ LIỆU VÀ DỰ ĐOÁN GIÁ XE BẰNG PHƯƠNG PHÁP HỒI QUY TUYẾN TÍNH

Giáo viên hướng dẫn :TS. Nguyễn Mạnh Cường

Nhóm - Lớp :10 - 20241IT6077002

Thành viên :Trần Xuân Vũ - 2021601795

Trần Văn Nam - 2021605962

Nguyễn Đình Quảng - 2022606022

Hà Nội, 2024

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn quý thầy, cô trường Đại Học Công Nghiệp Hà Nội đã tận tình dạy dỗ chúng em, trong đó phải kể đến quý thầy cô trong Khoa Công nghệ thông tin đã tạo điều kiện để chúng em thực hiện đề tài bài tập lớn.

Đặc biệt, chúng em xin chân thành cảm ơn giảng viên hướng dẫn – TS. Nguyễn Mạnh Cường đã tận tình giúp đỡ, hỗ trợ chúng em trong quá trình thực hiện đề tài. Cung cấp cho chúng em những kiến thức quý báu cũng như những lời khuyên hữu ích. Tạo động lực cho chúng em hoàn thành tốt nhiệm vụ của mình. Bên cạnh đó, chúng em cũng xin cảm ơn các bạn sinh viên trong Khoa Công nghệ thông tin đã đóng góp ý kiến giúp chúng em thực hiện đề tài đạt hiệu quả hơn.

Báo cáo bài tập lớn này đã giúp chúng em rèn luyện kỹ năng tư duy phân tích, xử lý dữ liệu và trình bày thông tin một cách có logic và rõ ràng. Chúng em hi vọng rằng những kiến thức và kinh nghiệm thu thập từ đề tài này sẽ tiếp tục hỗ trợ chúng em trong tương lai, không chỉ trong học tập mà còn trong sự nghiệp và cuộc sống.

Một lần nữa, chúng em xin chân thành cảm ơn sự hướng dẫn và định hướng của quý thầy cô và các bạn sinh viên khoa Công nghệ thông tin. Chúng em rất mong nhận được những ý kiến đóng góp để đề tài được hoàn thiện hơn.

Nhóm 10

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
RSE	Residual Standard Error: Sai số chuẩn của phần dư, đo lường mức độ khớp của mô hình.
SSR	*Sum of Squared Residuals*: Tổng bình phương sai số dư.
RMSE	*Root Mean Squared Error*: Căn bậc hai của sai số trung bình bình phương, đo lường độ chính xác dự đoán của mô hình.
R^2	*R-Squared*: Hệ số xác định, đo lường phần trăm biến thiên của biến phụ thuộc được giải thích bởi mô hình.

LỜI NÓI ĐẦU

Trong bối cảnh giao thông hiện đại, ô tô đóng vai trò quan trọng, phản ánh tính hiệu quả, phong cách cá nhân và địa vị kinh tế xã hội. Thị trường ô tô toàn cầu ngày càng phát triển với sự đa dạng về thương hiệu, mẫu mã và yếu tố ảnh hưởng đến giá xe. Do đó, việc hiểu và dự báo giá ô tô trở nên vô cùng cần thiết và phức tạp. Báo cáo này tập trung vào việc phân tích và khám phá các yếu tố ảnh hưởng đến giá trị ô tô, đặc biệt là ứng dụng kỹ thuật hồi quy tuyến tính trong dự đoán giá. Việc dự đoán chính xác giá ô tô có ý nghĩa quan trọng đối với cả ngành công nghiệp ô tô và người tiêu dùng.

Đối với các nhà sản xuất và đại lý, hiểu rõ các biến động giá giúp định hình chiến lược thị trường và đưa ra các quyết định kinh doanh hiệu quả. Đối với người tiêu dùng, việc nắm bắt thông tin về giá giúp đưa ra lựa chọn mua hàng sáng suốt.Giá ô tô chịu ảnh hưởng bởi nhiều yếu tố đa dạng, bao gồm thương hiệu, mẫu mã, năm sản xuất, loại nhiên liệu, v.v. Sự phức tạp của mạng lưới các yếu tố này tạo nên thách thức trong việc đánh giá giá trị ô tô.

Báo cáo của chúng em trình bày trong ba chương:

Chương 1. Tổng quan và phát biểu bài toán: Chương này giới thiệu tổng quan về phân tích dữ liệu và bài toán dự báo. Đặt nền móng, phát biểu bài toán, đi sâu vào lý do đằng sau việc lựa chọn chủ đề này, đặt ra các câu hỏi nghiên cứu cụ thể, làm nền tảng cho nghiên cứu của chúng em. Chương này nhấn mạnh tầm quan trọng của việc dự đoán giá ô tô trong thị trường ô tô phức tạp ngày nay, đặt nền tảng hiểu biết cho các phương pháp và cách tiếp cận tiếp theo.

Chương 2. Các kỹ thuật giải quyết bài toán: Chương này trình bày một số phương pháp, kỹ thuật phổ biến để giải quyết bài toán cũng như ưu nhược điểm của từng phương pháp. Từ đó lựa chọn, đi sâu vào các kỹ thuật cốt lõi để giải quyết bài toán phân tích dự báo giá ô tô. Trình bày, làm sáng tỏ những điều cơ bản của mô hình hồi quy tuyến tính, từ việc chuẩn bị dữ liệu đến trình bày và đánh giá mô hình. Nó nhằm mục đích làm sáng tỏ quá trình phân

tích, trang bị cho người đọc cái nhìn tổng quan về lý thuyết và bước thực hiện cụ thể, hỗ trợ cho việc hiểu rõ quá trình phân tích dữ liệu và dự đoán giá xe ô tô.

Chương 3. Kết quả thực nghiệm: Chương này trình bày ứng dụng thực tế của các phương pháp của chúng em vào bài toán. Ở đây, chúng em trình bày các kết quả đã đạt được từ phân tích của mình, đưa ra bằng chứng thực nghiệm về tính hiệu quả của mô hình. Chương này không chỉ nêu bật những phát hiện của chúng em mà còn khuyến khích sự suy ngẫm về những bài học kinh nghiệm và ý nghĩa rộng hơn của nghiên cứu của chúng em.

Khi thực hiện việc khám phá phân tích giá ô tô thông qua hồi quy tuyến tính, chúng em không chỉ mở rộng tầm nhìn học thuật mà còn tìm ra những hiểu biết sâu sắc về giá trị hữu hình cho tất cả các bên liên quan trong lĩnh vực ô tô. Báo cáo của chúng em phấn đấu không chỉ là một bài tập học thuật, chúng em mong muốn khơi dậy sự tò mò và nhiệt tình trong cộng đồng nghiên cứu, thúc đẩy những tiến bộ hơn nữa trong ngành công nghiệp ô tô. Chúng em hình dung công việc của mình như một bước đệm hướng tới những khám phá mới và tiến bộ có ý nghĩa trong việc hiểu và dự báo động lực ngày càng phát triển của giá ô tô.

CHƯƠNG 1. TỔNG QUAN VÀ PHÁT BIỂU BÀI TOÁN

1.1 Tổng quan về phân tích dữ liệu

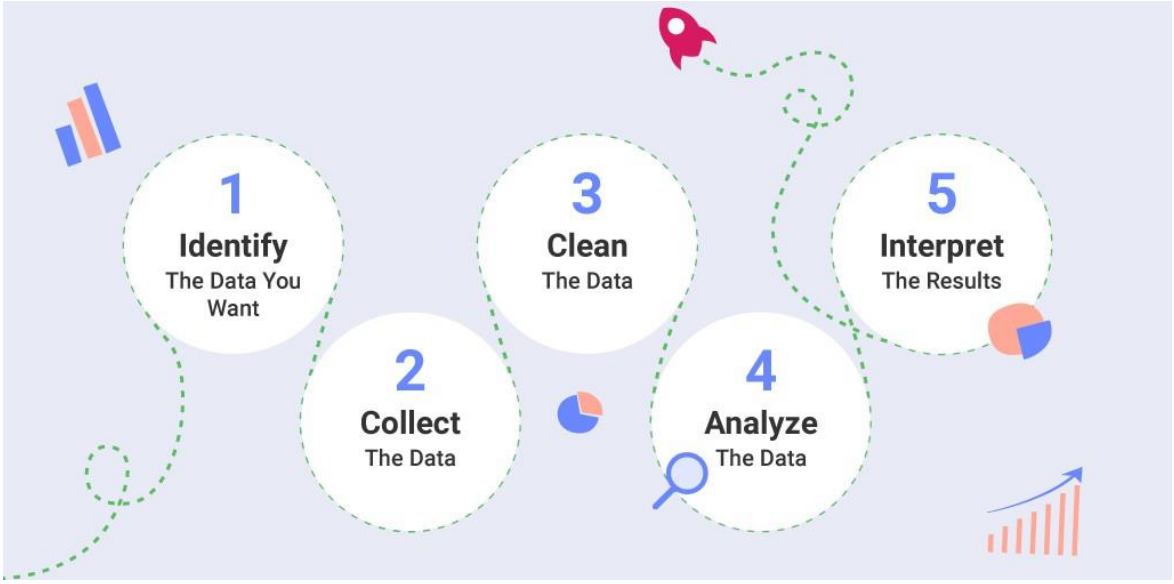
1.1.1 Khái niệm phân tích dữ liệu

Công nghệ thông tin đã đánh dấu một bước tiến quan trọng trong sự phát triển của xã hội hiện đại, mở ra những khả năng đưa thông tin và giao tiếp lên một tầm cao mới. Từ việc kết nối người với người, người với máy, đến việc tạo ra những hệ thống phức tạp như cloud computing, trí tuệ nhân tạo và Internet of Things, công nghệ thông tin đã thay đổi cách chúng ta sống, làm việc và tương tác.

Với sự gia tăng mạnh mẽ về lượng dữ liệu được tạo ra mỗi ngày từ các nguồn khác nhau như mạng xã hội, thiết bị di động, giao dịch trực tuyến đã tạo nên một kho tàng thông tin rất lớn, đa dạng và có tiềm năng rất lớn. Việc tận dụng những lượng thông tin lớn đó có thể giúp doanh nghiệp hiểu rõ hơn về các xu hướng của người dùng mà còn mang lại giá trị lớn cho các lĩnh vực như kinh doanh, y tế, giáo dục.

Công nghệ phân tích dữ liệu ngày càng trở nên mạnh mẽ, với sự sáng tạo trong các thuật toán máy học và khai phá dữ liệu, giúp chúng ta đưa ra những dự đoán chính xác hơn và đưa ra quyết định thông minh. Quá trình này không chỉ là về việc thu thập dữ liệu, mà còn là quá trình kiểm tra, làm sạch, chuyển đổi và mô hình hóa dữ liệu với mục tiêu khám phá thông tin hữu ích, đưa ra kết luận và hỗ trợ việc ra quyết định.

1.1.2 Quy trình phân tích dữ liệu



Hình 1. 1 Quy trình phân tích dữ liệu

Quy trình phân tích dữ liệu thường bao gồm các bước chính:

- **Xác định mục tiêu và thu thập dữ liệu**:
    - ***Xác định mục tiêu***: là những kết quả cụ thể mà ta muốn đạt được thông qua việc xử lý và phân tích dữ liệu. Mục tiêu này xác định hướng đi và phạm vi của quá trình phân tích, giúp ta tập trung vào việc thu thập thông tin quan trọng và thực hiện các phân để đáp ứng các yêu cầu hoặc nhu cầu cụ thể.
    - ***Thu thập dữ liệu***: là thu thập dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, tệp tin, trang web, thiết bị cảm biến, và nhiều nguồn khác. Dữ liệu có thể là số liệu, văn bản, hình ảnh, hoặc âm thanh.
- **Tiền xử lý dữ liệu**: Dữ liệu thường không hoàn hảo và có thể chứa nhiễu, dữ liệu bị thiếu, hoặc không chính xác. Tiền xử lý dữ liệu bao gồm việc tóm lược dữ liệu, làm sạch dữ liệu, tích hợp dữ liệu, chuyển đổi dữ liệu, rút gọn dữ liệu và rời rạc hóa dữ liệu để chuẩn bị cho bước phân tích.
- **Phân tích dữ liệu**: Bước quan trọng này dựa vào kiến thức và kỹ thuật phân tích để tìm ra mối liên hệ và thông tin ẩn sau dữ liệu. Phân tích dữ liệu có thể sử dụng các phương pháp phân tích mô tả, phân

tích hồi quy, phân tích sự khác biệt, thống kê, machine learning, data mining, và nhiều kỹ thuật khác.

- **Kết luận và dự đoán**: Dựa trên phân tích và thông tin từ dữ liệu, chúng ta có thể rút ra kết luận, hiểu rõ hơn về tình hình, và thậm chí đưa ra dự đoán cho tương lai.

1.2 Tổng quan về bài toán dự báo

1.2.1 Lịch sử về bài toán dự báo

❖ **Phát triển ban đầu (1940 – 1950)**

Nguồn gốc của phân tích dự đoán có thể bắt nguồn từ những ngày đầu của phân tích thống kê. Các nhà toán học và thống kê bắt đầu phát triển các mô hình để đưa ra dự đoán dựa trên dữ liệu lịch sử. Những mô hình ban đầu này thường tập trung vào hồi quy tuyến tính và các phương pháp thống kê cơ bản.

❖ **Máy tính lớn (1960 - 1970)**

Với sự ra đời của máy tính lớn, sức mạnh tính toán tăng lên, cho phép tính toán và phân tích phức tạp hơn. Các doanh nghiệp và nhà nghiên cứu bắt đầu sử dụng máy tính để phân tích các tập dữ liệu lớn hơn và đưa ra dự đoán trong nhiều lĩnh vực khác nhau như tài chính và tiếp thị.

❖ **Hệ thống hỗ trợ quyết định (1980-1990)**

Trong những năm 1980 và 1990, hệ thống hỗ trợ quyết định (DSS) trở nên phổ biến. Các hệ thống này tích hợp các công cụ phân tích dữ liệu và mô hình hóa để hỗ trợ người ra quyết định đưa ra những lựa chọn sáng suốt. Phân tích dự đoán bắt đầu có sức hút trong môi trường kinh doanh để dự báo và đánh giá rủi ro.

❖ **Khai thác dữ liệu (1990-2000)**

Sự gia tăng của khai thác dữ liệu vào cuối thế kỷ 20 đã đánh dấu một cột mốc quan trọng trong phân tích dự đoán. Các kỹ thuật khai thác dữ liệu, bao gồm mạng lưới thần kinh và thuật toán học máy, cho phép phân tích các tập dữ

liệu lớn phức tạp hơn. Thời đại này chứng kiến sự áp dụng ngày càng tăng trong các ngành như tài chính, chăm sóc sức khỏe và bán lẻ.

❖ **Học máy và Trí tuệ nhân tạo (2010 - Nay)**

Trong những năm gần đây, học máy (ML) và trí tuệ nhân tạo (AI) đã trở thành một phần không thể thiếu trong phân tích dự đoán. Những công nghệ này cho phép phân tích phức tạp hơn, nhận dạng mẫu và lập mô hình dự đoán. Học sâu, một tập hợp con của học máy, đã cho thấy thành công đáng kể trong nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên và các ứng dụng khác.

❖ **Phân tích dự đoán theo thời gian thực (Hiện tại – Tương lai)**

Xu hướng hiện tại là hướng tới phân tích dự đoán theo thời gian thực, trong đó các tổ chức hướng đến việc đưa ra dự đoán và quyết định ngay lập tức. Điều này đặc biệt quan trọng trong các lĩnh vực như tài chính, an ninh mạng và IoT, nơi cần có những phản ứng kịp thời trước những điều kiện thay đổi.

Bài toán dự báo là một trong những thách thức quan trọng trong lĩnh vực phân tích dữ liệu, nơi chúng ta cố gắng dự đoán giá trị của một biến mục tiêu trong tương lai dựa trên dữ liệu lịch sử và các yếu tố ảnh hưởng. Mục tiêu chính của bài toán dự báo là xây dựng một mô hình có khả năng hiểu và ứng dụng các mẫu, xu hướng và quy luật từ dữ liệu để thực hiện việc dự đoán một cách chính xác và đáng tin cậy.

1.2.2 Tình hình nghiên cứu trong nước

Bài toán dự báo có sự ảnh hưởng to lớn tại cả Việt Nam. Dự báo giúp cải thiện quản lý, định hình chiến lược, và tối ưu hóa tài nguyên trong nhiều lĩnh vực. Có một số điểm đáng chú ý về tình hình phân tích dữ liệu tại Việt Nam:

- Phát triển đang ở giai đoạn đầu: Trong một số lĩnh vực, bài toán dự báo tại Việt Nam đang ở giai đoạn đầu của sự phát triển. Việc áp dụng các phương pháp phân tích dữ liệu và dự báo mới còn đang được tìm hiểu và thí nghiệm.
- Ứng dụng trong nông nghiệp và kinh tế: Tại Việt Nam, dự báo có ứng dụng quan trọng trong nông nghiệp, nhằm dự đoán thời tiết, mùa

màng, và nhu cầu năng lượng. Nó cũng được áp dụng trong kinh tế, dự báo tăng trưởng GDP, lạm phát, và tỷ giá.

- Thách thức từ dữ liệu: Một thách thức cho việc dự báo tại Việt Nam là khả năng thu thập và quản lý dữ liệu chất lượng. Dữ liệu thường không đầy đủ và có thể gặp vấn đề về tính nhất quán và độ tin cậy.

1.2.3 Tình hình nghiên cứu ở nước ngoài

Trong lĩnh vực nghiên cứu bài toán dự báo đã có một số công trình nghiên cứu ngoài nước có liên quan đến đề tài tiểu luận, ví dụ như: "Solar Forecast Reconciliation and Effects of Improved Base Forecasts" được đăng trên IEEE Xplore, tác giả: Gokhan Mert Yagli, Dazhi Yang, Dipti Srinivasan, Monika. Đề tài nghiên cứu này trình bày về dự báo sản lượng điện mặt trời đóng vai trò quan trọng trong vận hành hệ thống điện. Dự báo được yêu cầu trên các quy mô địa lý và thời gian khác nhau, có thể được mô hình hóa dưới dạng phân cấp. [1]

- Phát triển mạnh: Tại các quốc gia phát triển, bài toán dự báo đã được phát triển mạnh và có sự ứng dụng rộng rãi trong nhiều lĩnh vực như tài chính, thương mại điện tử, y tế, và năng lượng.
- Sự kết hợp của công nghệ mới: Các quốc gia nước ngoài thường kết hợp sự phát triển của công nghệ mới như trí tuệ nhân tạo, học máy, và big data analytics để cải thiện hiệu suất của bài toán dự báo.
- Tổng hợp dữ liệu: Một ưu điểm của các quốc gia phát triển là có khả năng tổng hợp dữ liệu từ nhiều nguồn khác nhau, tạo nền tảng cho việc dự báo chính xác hơn và đa dạng hơn.

1.3 Phát biểu bài toán

Bài toán "Phân tích dữ liệu và dự báo giá xe ô tô bằng phương pháp hồi quy tuyến tính" là một đề tài trong lĩnh vực phân tích dữ liệu, tập trung vào việc hiểu và dự đoán giá trị của xe ô tô cũ hoặc mới trong tương lai dựa trên các yếu tố như đặc điểm kĩ thuật, đặc tính của xe. Bài toán này đặt ra mục tiêu xác định các yếu tố ảnh hưởng đến mức lương và sử dụng phương pháp hồi quy tuyến tính để xây dựng một mô hình dự báo.

1.3.1 Xác định đầu vào, đầu ra của bài toán

Đầu vào bài toán là tập dữ liệu xe ô tô từ năm 1998 - 2019. Gồm nhiều bản ghi (hàng) với mỗi hàng đại diện cho một chiếc xe ô tô. Mỗi chiếc xe sẽ được mô tả bởi các biến (đặc trưng) như hãng sản xuất, mẫu xe, năm sản xuất, số lượng dặm đã đi, loại nhiên liệu, thông số kỹ thuật. Tập dữ liệu sẽ bao gồm thông tin đầy đủ và đa dạng để mô hình có thể học được các mối quan hệ giữa các biến đầu vào và giá của xe ô tô. Nó cũng có thể chứa các giá trị thiếu, ngoại lai, hoặc các vấn đề khác mà quá trình tiền xử lý cần phải giải quyết.

Đầu ra sẽ là giá xe ô tô theo từng năm, là biến mục tiêu mà mô hình sẽ được huấn luyện để dự đoán. Giá có thể được biểu diễn dưới dạng giá trị thực hoặc giá trị chuẩn hóa tùy thuộc vào yêu cầu cụ thể của bài toán. Ngoài ra còn có thông số để đánh giá độ chính xác mô hình.

1.3.2 Mục tiêu nghiên cứu

- ***Tiền xử lý*:** Với mục đích đảm bảo chất lượng và hiệu suất của mô hình dự đoán. Quá trình này giúp ta kỹ năng phân tích dữ liệu một cách chi tiết và sâu sắc để hiểu rõ về đặc điểm và đặc trưng của dữ liệu xe ô tô. Sự am hiểu này không chỉ hỗ trợ việc xác định và xử lý các vấn đề như giá trị thiếu, ngoại lai, mà còn giúp tạo ra cái nhìn tổng quan về ngữ cảnh và tính đặc biệt của dữ liệu.
- ***Phân tích yếu tố ảnh hưởng*:** Hiểu rõ các yếu tố có thể ảnh hưởng đến giá xe ô tô. Các yếu tố này có thể là năm sản xuất, số km đã đi, thông số động cơ, cách chế tạo và các yếu tố khác. Việc này không

chỉ giúp tối ưu hóa hiệu suất mô hình mà còn nâng cao khả năng hiểu biết sâu sắc về yếu tố quyết định giá xe ô tô trong ngữ cảnh cụ thể.

- ***Xây dựng mô hình hồi quy*:** Sử dụng phương pháp hồi quy tuyến tính để xây dựng mô hình dự báo giá xe dựa trên các yếu tố ảnh hưởng đã được xác định. Mô hình hồi quy sẽ cố gắng tìm ra mối quan hệ giữa các biến độc lập và biến phụ thuộc (giá xe).
- ***Dự đoán giá xe*:** Dựa trên mô hình hồi quy đã xây dựng, mục tiêu là dự đoán giá xe theo từng cho những chiếc xe có các thông tin liên quan đã được cung cấp.

1.3.3 Ý nghĩa khoa học và thực tiễn

- ***Khoa học dữ liệu***: Đề tài này đóng góp vào lĩnh vực Khoa học Dữ liệu bằng cách áp dụng các kỹ thuật phân tích dữ liệu và hồi quy để khám phá mối liên hệ giữa yếu tố ảnh hưởng và giá xe, từ đó cung cấp thông tin giá trị về ngành và thị trường lao động bên vận tải.
- ***Nắm bắt thị trường về các dòng xe***: Kết quả của nghiên cứu có thể giúp các công ty và tổ chức trong ngành Khoa học Dữ liệu hiểu rõ hơn về các yếu tố ảnh hưởng đến giá xe. Điều này có thể hỗ trợ trong việc đưa ra quyết định về thu mua xe đối với những người có nhu cầu.
- ***Tư duy phân tích***: Việc thực hiện phân tích dữ liệu và xây dựng mô hình hồi quy trong ngữ cảnh của bài toán này cũng giúp phát triển kỹ năng tư duy phân tích, sáng tạo, và khả năng áp dụng các phương pháp phân tích vào các vấn đề thực tế.

1.3.4 Cơ hội và khó khăn dự tính

Khi nghiên cứu các sắc thái của dự báo giá ô tô, nhóm chúng em gặp phải một bối cảnh phong phú với cả những lợi ích tiềm năng và những thách thức như:

1. ***Dự đoán biến động giá***: Rào cản lớn trong phân tích giá ô tô là khả năng dự báo chính xác sự thay đổi giá. Các yếu tố bên ngoài như chi phí nguyên liệu thô, xu hướng thị trường và hành động của đối thủ cạnh tranh có thể phá vỡ chiến lược giá cả, khiến sự ổn định trở thành một thách thức.
2. ***Tác động của các yếu tố bên ngoài***: Các yếu tố bên ngoài như sự thay đổi kinh tế, thay đổi chính trị hoặc tác động môi trường có thể ảnh hưởng đến giá ô tô một cách khó lường. Tính không thể đoán trước này làm tăng thêm sự phức tạp trong việc duy trì các chiến lược định giá nhất quán và đáng tin cậy.
3. ***Đáp ứng nhu cầu đang thay đổi của thị trường***: Ngành công nghiệp ô tô phải thích ứng với những thay đổi nhanh chóng trong sở thích của người tiêu dùng, chẳng hạn như sự phổ biến ngày càng tăng của xe điện hoặc công nghệ lái xe tự động. Những thay đổi này tạo thêm thách thức cho việc phân tích giá, đòi hỏi một cách tiếp cận năng động để hiểu nhu cầu thị trường.

Về bản chất, nhiệm vụ phân tích, dự báo giá ô tô là một công việc cân bằng, mang lại những cơ hội đáng kể để tối ưu hóa giá trị và lập kế hoạch chiến lược, đồng thời đòi hỏi sự linh hoạt để chống lại sự biến động của thị trường và những ảnh hưởng từ bên ngoài. Sự phân đôi này nhấn mạnh sự cần thiết phải đổi mới liên tục và khả năng thích ứng trong ngành công nghiệp ô tô.

CHƯƠNG 2. CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN

2.1 Phương pháp phân tích mô tả

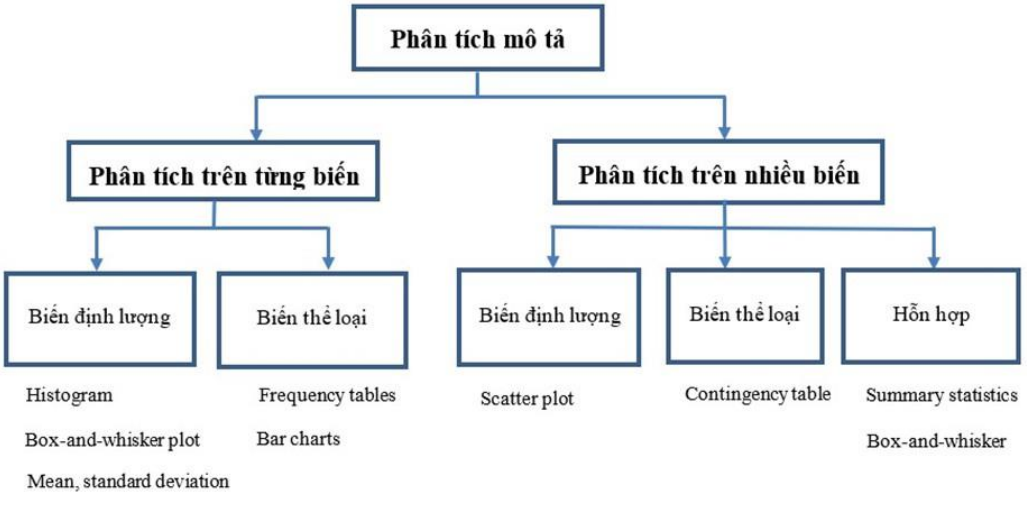
2.1.1 Phân tích mô tả

Phân tích mô tả (thường được hiểu là thống kê mô tả) là một phương pháp thống kê được sử dụng để tóm tắt, sắp xếp, đơn giản hóa, mô tả và trình bày dữ liệu đã thu thập dưới dạng số hoặc biểu đồ trực quan, nhằm mô tả và tóm tắt các đặc điểm chính của một tập dữ liệu một cách dễ hiểu và ngắn gọn.

Mục tiêu của phân tích mô tả là giúp hiểu sâu hơn về dữ liệu mà chúng ta đang làm việc, nhận ra các đặc trưng quan trọng, và cung cấp một cái nhìn tổng quan về phân phối và biến đổi của dữ liệu.

Tùy thuộc vào loại biến hay kiểu dữ liệu để quyết định sử dụng các phương pháp tiếp cận phù hợp. Dữ liệu được chia thành hai loại:

- ***Dữ liệu định lượng*** (quantitative data) hay biến định lượng (quantitative variable): thường đại diện cho số lượng hoặc giá trị số (ví dụ: tuổi, cân nặng, giá tiền, âm lượng, v.v.)
- ***Dữ liệu thể loại*** (categorical data) hay biến thể loại (categorical variable): mô tả chất lượng hoặc đặc điểm của các đối tượng (ví dụ: màu sắc, dân tộc, giới tính, v.v.)



Hình 1. 2 Phân tích mô tả

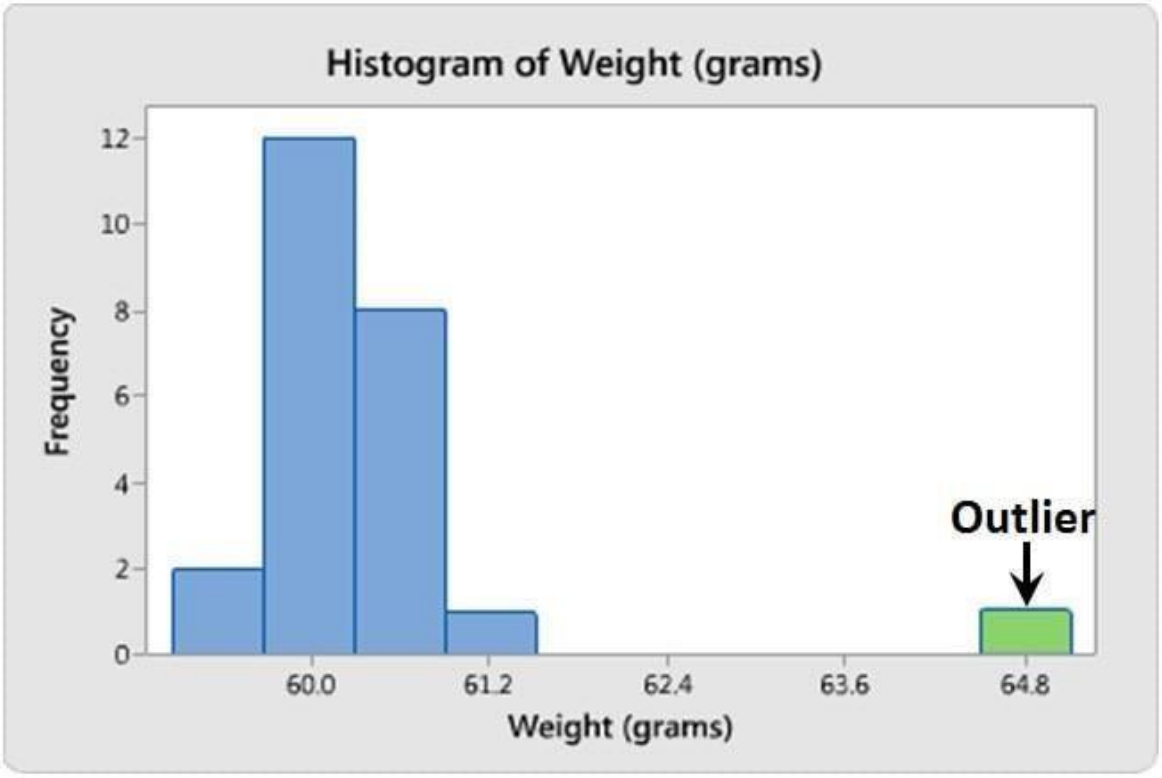
Phân tích mô tả thường bao gồm các khía cạnh sau:

- ***Thống kê tóm tắt***: Đây là các số liệu thống kê cơ bản như trung bình, trung vị, độ lệch chuẩn, và phân vị. Các số liệu này giúp ta hiểu về trung tâm và phân tán của dữ liệu.
- ***Biểu đồ***: Biểu đồ thường được sử dụng để biểu diễn dữ liệu một cách trực quan. Các biểu đồ như biểu đồ cột, biểu đồ đường, biểu đồ hình tròn, và biểu đồ hộp giúp ta thấy được sự phân bố và xu hướng của dữ liệu.
- ***Phân phối dữ liệu***: Phân tích phân phối dữ liệu giúp ta hiểu về tỷ lệ xuất hiện của các giá trị khác nhau trong tập dữ liệu. Điều này có thể làm bằng cách tạo biểu đồ phân phối tần số hoặc xây dựng biểu đồ kernel density.
- ***Kiểm tra sự tương quan***: Phân tích mô tả cũng có thể liên quan đến việc kiểm tra sự tương quan giữa các biến. Điều này có thể thực hiện bằng cách sử dụng biểu đồ tương quan hoặc tính toán hệ số tương quan Pearson.
- ***Xác định điểm ngoại lệ***: Phân tích mô tả cũng giúp xác định các điểm dữ liệu ngoại lệ, tức là những giá trị rất khác biệt so với phần còn lại của dữ liệu.
- ***Tổng kết và nhận xét***: Cuối cùng, phân tích mô tả thường đi kèm với việc tổng kết và nhận xét về các đặc điểm quan trọng của dữ liệu, những mẫu thú vị, và những điểm mạnh và điểm yếu của tập dữ liệu.

Phân tích mô tả giúp xây dựng một cái nhìn sâu hơn về tập dữ liệu ban đầu và tạo nền tảng cho các phân tích tiếp theo như dự báo, phân tích hồi quy, hay machine learning.

2.1.2 Phương pháp phân tích trên từng biến

Khi thực hiện phân tích trên một biến (hoặc một thuộc tính), mục tiêu chính là hiểu rõ các đặc điểm cơ bản của biến đó. Điều này thường bao gồm xác định và xử lý các giá trị ngoại lai hoặc bất thường (Outliers). Đây là các giá trị dữ liệu mà rất khác biệt so với phần lớn các giá trị khác trong tập dữ liệu. Các giá trị ngoại lai có thể xuất hiện do lỗi nhập liệu, lỗi đo lường, hoặc đơn giản là do các sự kiện hiếm gặp.



Hình 2. 1Biểu đồ Histogram giúp xác định giá trị ngoại lai (Outliers)

Việc xác định các Outliers có vai trò quan trọng và là mắt xích liên kết giữa phân tích mô tả và phân tích hồi quy, bởi vì ta có thể tiến hành làm sạch những giá trị này tại công đoạn tiền xử lý dữ liệu của phân tích hồi quy. Cụ thể với từng loại dữ liệu khác nhau, ta sẽ phân tích như sau:

Dữ liệu số:

- ***Biểu đồ Histogram***: Biểu đồ hiển thị tần suất xuất hiện của các khoảng giá trị dữ liệu.

- ***Các đại lượng thống kê***: Bao gồm mean (trung bình), stdev (độ lệch chuẩn), median (trung vị), quartile (phân vị)... Các giá trị này giúp mô tả trung bình, phương sai và phân phối của dữ liệu.
- ***Biểu đồ Box & Whisker (Boxplot)***: Biểu đồ hiển thị tổng quan giá trị đó bao gồm các giá trị đại lượng thống kê đã tính được.

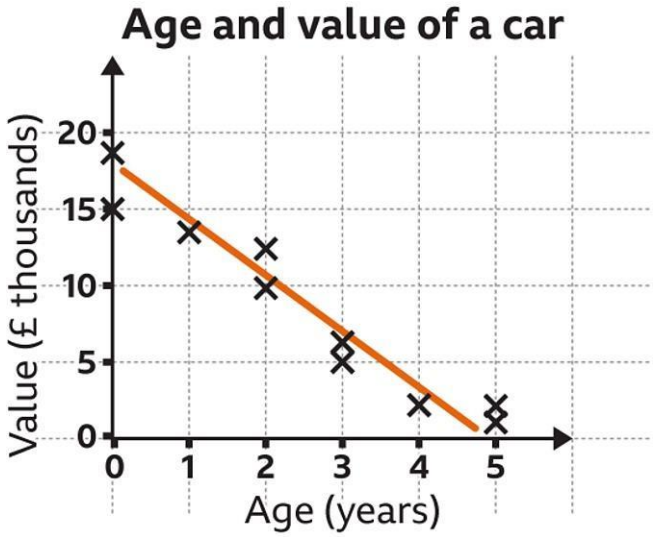
Dữ liệu phi số:

- ***Bảng tần suất (Frequency table)***: Biểu đồ liệt kê các giá trị khác nhau của biến và số lần xuất hiện của mỗi giá trị.
- ***Biểu đồ cột (Bar chart)***: Biểu đồ thể hiện tần suất của từng giá trị dữ liệu dưới dạng các cột đứng.
- ***Biểu đồ hình tròn hoặc donut (Pie chart, Donut chart)***: Biểu đồ thể hiện phần trăm tần suất của từng giá trị trong tổng số.

2.1.3 Phương pháp phân tích trên nhiều biến

Phân tích trên nhiều biến hướng tới việc hiểu mối quan hệ và tương tác giữa các biến trong tập dữ liệu. Điều này có thể giúp bạn phát hiện ra các mẫu, xu hướng hoặc tương quan có thể tồn tại giữa chúng.

Các mối liên hệ giữa các biến (Interrelationships) có thể là nhiều dạng khác nhau: Mối tương quan tuyến tính, tương quan không tuyến tính, tương quan ngược... Với mỗi mối liên hệ, ta có thể phân tích và tìm ra được cách các biến tương tác và ảnh hưởng lẫn nhau.



Hình 2. 2Biểu đồ Scatter thể hiện mối quan hệ giữa độ tuổi và giá bán

Việc phân tích trên nhiều biến cũng có mối liên hệ mật thiết đến phân tích hồi quy khi giúp ta xác định được các giá trị ngoại lai của dữ liệu. Do là phân tích nhiều biến, vậy nên sẽ có 3 kiểu dữ liệu phân tích khác nhau: Số, phi số và hỗn hợp (cả số và phi số):

Dữ liệu số:

- ***Scatter Plot (Biểu đồ Scatter)***: Biểu đồ thể hiện mối quan hệ giữa hai biến số. Mỗi điểm trên biểu đồ thể hiện một cặp giá trị của hai biến trên trục ngang và dọc. Biểu đồ này dùng để tìm kiếm sự

tương quan giữa 2 biến số như tương quan tuyến tính hoặc không tuyến tính.

- ***Bảng dữ liệu thống kê (Statistical Summary Table)***: Tạo bảng để liệt kê các đại lượng thống kê (mean, median, stdev…) giữa các biến số của dữ liệu.

Dữ liệu phi số:

- ***Bảng dữ liệu thống kê (Statistical Summary Table)***: Cũng là bảng dữ liệu thống kê nhưng với giá trị phi số, đó sẽ chỉ có giá trị tần suất xuất hiện (mode) của dữ liệu.

Dữ liệu hỗn hợp

- ***Bảng thống kê tổng hợp***: Đây là sự kết hợp giữa bảng dữ liệu thống kê của dữ liệu số và phi số. Sự kết hợp tổng quan này sẽ cho ta bao quát được phân bổ của dữ liệu.
- ***Biểu đồ Box-and-Whisker (Boxplot)***: Được sử dụng để so sánh phân phối của một dữ liệu số với tần suất của một dữ liệu phi số. Biểu đồ này sẽ cho ta mối quan hệ mật thiết về sự ảnh hưởng của các giá trị phi số lên giá trị số được phân tích.

2.2 Phương pháp phân tích hồi quy

2.2.1 Tổng quan về phân tích hồi quy

Phân tích hồi quy là một tập hợp các phương pháp thống kê được sử dụng để ước tính các mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Nó có thể được sử dụng để đánh giá sức mạnh của mối quan hệ giữa các biến và để mô hình hóa mối quan hệ trong tương lai giữa chúng.

Phân tích hồi quy là một cách phân loại toán học để xác định biến nào trong số những biến đó thực sự có tác động. Nó trả lời các câu hỏi: Yếu tố nào quan trọng nhất? Cái nào có thể bỏ qua? Các yếu tố đó tương tác với nhau như thế nào? Và quan trọng nhất, chúng ta chắc chắn như thế nào về tất cả những yếu tố này? [2]

Trong phân tích hồi quy, ta cần xác định một biến phụ thuộc – yếu tố chính mà ta đang cố gắng hiểu hoặc dự đoán. Phân tích hồi quy bao gồm một số biến thể, chẳng hạn như tuyến tính, nhiều tuyến tính và phi tuyến tính. Trong đó mô hình phổ biến là tuyến tính và nhiều tuyến tính. Đối với phân tích hồi quy phi tuyến, chúng thường được sử dụng cho các tập dữ liệu phức tạp hơn trong đó các biến phụ thuộc và độc lập thể hiện mối quan hệ phi tuyến.

2.2.2 Một số phương pháp phân tích hồi quy

Để phân tích hồi quy có rất nhiều phương pháp để phân tích. Dưới đây sẽ là một số phương pháp quan trọng dùng để phân tích hồi quy:

- ***Hồi quy tuyến tính (Linear Regression)***: Hồi quy tuyến tính dự đoán giá trị mục tiêu dựa trên biến độc lập bằng cách tìm đường thẳng "tốt nhất" vượt qua dữ liệu. Phương pháp này đơn giản và phù hợp với dữ liệu có mối quan hệ tuyến tính. Tuy nhiên, nó có thể không xử lý được dữ liệu phi tuyến và ảnh hưởng bởi nhiễu dữ liệu.
- ***Hồi quy Ridge (Ridge Regression)***: Hồi quy Ridge là phiên bản cải tiến của hồi quy tuyến tính bằng cách thêm hệ số điều chuẩn l2 vào hàm mất mát. Điều này giúp kiểm soát độ phức tạp của mô hình và tránh tình trạng quá khớp (overfitting). Tuy ưu điểm là giảm

overfitting và xử lý đa cộng tuyến, nhưng cần lựa chọn tham số điều chuẩn chính xác.

- ***Hồi quy Lasso (Lasso Regression)***: Hồi quy Lasso cũng cải tiến từ hồi quy tuyến tính, nhưng thay vì l2, nó sử dụng hệ số điều chuẩn l1 để thúc đẩy một số hệ số về 0. Điều này dẫn đến lựa chọn biến tự động và giảm biến quan trọng. Lasso giải quyết vấn đề "chọn biến" nhưng cần phải có tham số điều chuẩn chính xác.

2.2.3 Phương pháp Lasso Regression

2.2.3.1 Giới thiệu

Hồi quy LASSO (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator - Toán tử lựa chọn và co rút tuyệt đối nhỏ nhất). Co rút là nơi các giá trị dữ liệu được thu nhỏ về một điểm trung tâm, chẳng hạn như giá trị trung bình. Kỹ thuật LASSO khuyến khích các mô hình đơn giản, thưa thớt (tức là các mô hình có ít tham số hơn). Kiểu hồi quy cụ thể này rất phù hợp với các mô hình có mức độ đa cộng tuyến cao hoặc khi bạn muốn tự động hóa một số phần nhất định của việc lựa chọn mô hình, như lựa chọn biến/loại bỏ tham số.

Mục tiêu chính của hồi quy LASSO là tìm sự cân bằng giữa tính đơn giản và độ chính xác của mô hình. Nó đạt được điều này bằng cách thêm một số hạng phạt vào mô hình hồi quy tuyến tính truyền thống, mô hình này khuyến khích các giải pháp thưa thớt trong đó một số hệ số buộc phải chính xác bằng 0. Tính năng này làm cho LASSO đặc biệt hữu ích cho việc lựa chọn tính năng vì nó có thể tự động xác định và loại bỏ các biến không liên quan hoặc dư thừa.

2.2.3.2 Một số khái niệm chính

Chính quy hóa: Chính quy hóa là một kỹ thuật được sử dụng để ngăn chặn việc trang bị quá mức bằng cách không khuyến khích các mô hình quá phức tạp trong hồi quy. Điều này đạt được bằng cách thêm một số hạng phạt vào hàm mất mát.

Độ thưa thớt: Một tính năng chính của hồi quy LASSO là khả năng tạo ra các mô hình thưa thớt. Độ thưa thớt đề cập đến các mô hình trong đó một số

hệ số chính xác bằng 0, ngụ ý rằng các tính năng tương ứng bị loại trừ hoàn toàn khỏi mô hình. Điều này đặc biệt hữu ích cho việc lựa chọn tính năng trong bộ dữ liệu nhiều chiều.

Đánh đổi sai lệch-phương sai: Bằng cách đưa ra thuật ngữ chính quy hóa, hồi quy LASSO làm tăng độ lệch nhưng làm giảm phương sai của các dự đoán mô hình, dẫn đến khái quát hóa tốt hơn về dữ liệu không nhìn thấy.

Tham số điều chỉnh (λ): Cường độ của hình phạt được xác định bởi tham số điều chỉnh, λ. Việc lựa chọn λ có thể có tác động đáng kể đến kết quả của mô hình, với các giá trị cao hơn dẫn đến sự chính quy hóa nhiều hơn (nghĩa là nhiều hệ số được đặt thành 0).

2.2.3.3 Mô hình hồi quy LASSO

Hàm mục tiêu cho hồi quy LASSO có thể được viết theo công thức:

$$\text{Minimize} \left(\frac{1}{N} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

Trong đó:

y_i là biến phụ thuộc.

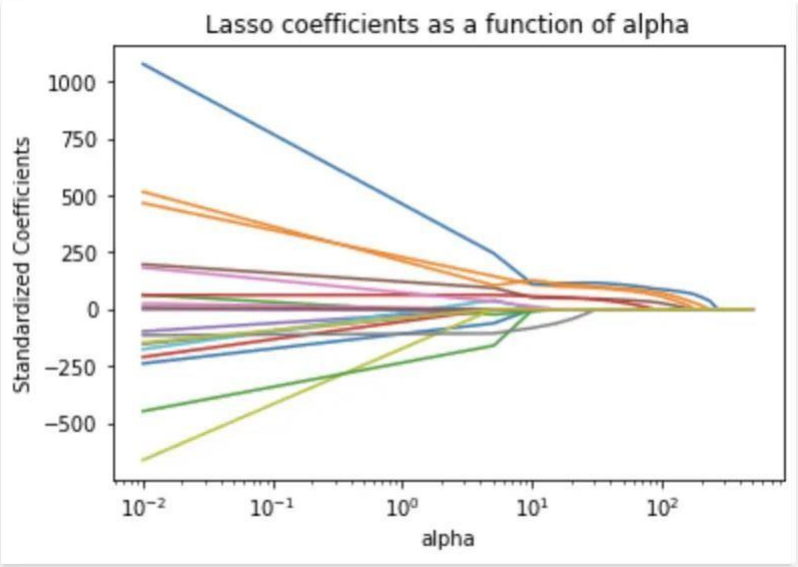
X_{ij} là biến dự đoán.

β_j là các hệ số cần tìm.

N là số lượng quan sát.

P là số lượng các biến dự đoán.

λ là tham số chính quy kiểm soát cường độ của hình phạt. Giá trị của λ càng lớn thì mức độ co ngót càng lớn.



Hình 2. 3Hệ số Lasso là hàm của λ

Tham số điều chỉnh, λ kiểm soát cường độ của hình phạt L1. λ về cơ bản là độ co ngót:

- Khi λ = 0, không có tham số nào bị loại bỏ. Ước tính này bằng với ước tính được tìm thấy bằng hồi quy tuyến tính.
- Khi λ tăng lên, ngày càng có nhiều hệ số được đặt về 0 và bị loại bỏ (về mặt lý thuyết, khi λ = ∞, tất cả các hệ số đều bị loại bỏ).
- Khi λ tăng, độ lệch tăng.
- Khi λ giảm, phương sai tăng

2.2.3.4 Ưu điểm của hồi quy LASSO

Hồi quy Lasso là một kỹ thuật hồi quy tuyến tính được sử dụng rộng rãi, mang lại một số lợi thế so với hồi quy tuyến tính truyền thống:

1. **Lựa chọn tính năng:**

Hồi quy Lasso có thể được sử dụng để lựa chọn tính năng, trong đó nó xác định các yếu tố dự đoán quan trọng nhất và loại bỏ phần còn lại. Điều này có thể đặc biệt hữu ích khi xử lý các tập dữ liệu nhiều chiều, trong đó số lượng yếu tố dự đoán lớn và nhiều trong số chúng có thể không liên quan hoặc dư thừa.

2. Chính quy hóa:

Hồi quy Lasso bao gồm một hình phạt chính quy hóa trong hàm mục tiêu, giúp ngăn chặn việc điều chỉnh quá mức và cải thiện hiệu suất tổng quát hóa của mô hình. Hình phạt thu hẹp các hệ số về 0, dẫn đến một mô hình đơn giản hơn và ít có khả năng bị trang bị quá mức.

3. Độ thưa thớt:

Hồi quy Lasso khuyến khích sự thưa thớt trong mô hình bằng cách thu nhỏ một số hệ số về chính xác bằng 0. Điều này dẫn đến một mô hình có ít yếu tố dự đoán hơn, dễ diễn giải hơn và có thể dẫn đến độ chính xác dự đoán tốt hơn.

4. Sự đánh đổi độ lệch-phương sai:

Hồi quy Lasso đưa ra sự cân bằng giữa sai lệch và phương sai bằng cách kiểm soát độ mạnh của hình phạt chính quy hóa. Bằng cách tăng mức phạt, Lasso có thể giảm phương sai trong mô hình với cái giá phải trả là độ lệch tăng lên hoặc ngược lại.

5. Có thể giải thích:

Hồi quy Lasso tạo ra một mô hình dễ diễn giải vì nó chỉ bao gồm một tập hợp con các biến dự đoán và gán hệ số 0 cho các hệ số không liên quan. Điều này có thể giúp hiểu được mối quan hệ giữa các yếu tố dự đoán và biến mục tiêu.

6. Tính linh hoạt:

Hồi quy Lasso có thể được áp dụng cho nhiều vấn đề hồi quy, bao gồm hồi quy tuyến tính và phi tuyến tính, cũng như các mô hình tuyến tính tổng quát. Nó cũng tương thích với các thuật toán tối ưu hóa khác nhau và có thể xử lý cả tập dữ liệu nhỏ và lớn.

Nhìn chung, hồi quy Lasso là một kỹ thuật mạnh mẽ và linh hoạt, mang lại một số lợi thế so với hồi quy tuyến tính truyền thống, khiến nó trở thành lựa chọn phổ biến trong các ứng dụng phân tích dữ liệu và học máy.

2.2.3.5 Nhược điểm của hồi quy LASSO

Mặc dù hồi quy Lasso là một kỹ thuật hồi quy phổ biến và hữu ích nhưng nó cũng có một số hạn chế và nhược điểm cần được xem xét:

1. Xu hướng lựa chọn tính năng:

Hình phạt chính quy hóa Lasso có thể dẫn đến một số tính năng bị loại trừ hoàn toàn khỏi mô hình, ngay cả khi chúng có thể là yếu tố dự báo quan trọng của biến mục tiêu. Điều này có thể dẫn đến việc lựa chọn tính năng sai lệch, đặc biệt nếu mối quan hệ thực sự giữa các yếu tố dự đoán và mục tiêu không thưa thớt.

2. Thông số không ổn định:

Lasso có thể nhạy cảm với những thay đổi nhỏ trong dữ liệu đầu vào, dẫn đến sự chênh lệch lớn trong các hệ số ước tính. Điều này có thể dẫn đến sự mất ổn định trong các tham số của mô hình, gây khó khăn cho việc diễn giải kết quả.

3. Sự đánh đổi độ lệch-phương sai:

Hình phạt chính quy hóa Lasso có thể làm giảm phương sai trong mô hình bằng cách thu hẹp các hệ số về 0, nhưng nó cũng có thể gây ra sai lệch bằng cách đánh giá thấp các hệ số thực. Sự cân bằng tối ưu giữa sai lệch và phương sai phụ thuộc vào vấn đề và tập dữ liệu cụ thể.

4. Trang bị quá mức:

Mặc dù Lasso được thiết kế để ngăn chặn việc khớp quá mức bằng cách đưa ra một thuật ngữ phạt, nhưng mô hình vẫn có thể khớp quá mức với dữ liệu huấn luyện nếu tham số chính quy hóa không được điều chỉnh đúng cách. Điều này có thể dẫn đến hiệu suất khái quát hóa kém trên dữ liệu mới.

5. Lựa chọn tham số chính quy:

Việc lựa chọn tham số chính quy α là rất quan trọng đối với hiệu suất của mô hình Lasso. Tuy nhiên, không có giải pháp phân tích nào để tìm giá trị tối ưu của α và nó phải được xác định bằng thực nghiệm bằng cách xác nhận chéo.

Việc này có thể tốn thời gian và không phải lúc nào cũng đưa đến sự lựa chọn tốt nhất về α.

6. Đa cộng tuyến:

Lasso có thể nhạy cảm với đa cộng tuyến, đó là khi hai hoặc nhiều yếu tố dự đoán có mối tương quan cao. Trong trường hợp này, Lasso có thể chọn một trong các yếu tố dự đoán tương quan và loại trừ yếu tố còn lại, ngay cả khi cả hai đều quan trọng để dự đoán biến mục tiêu.

Nhìn chung, mặc dù hồi quy Lasso là một kỹ thuật hữu ích để chính quy hóa và lựa chọn tính năng, nhưng điều quan trọng là phải xem xét cẩn thận các hạn chế và nhược điểm tiềm ẩn của nó khi áp dụng nó cho một vấn đề hoặc tập dữ liệu cụ thể.

2.2.3.6 Tính ứng dụng và hạn chế của hồi quy LASSO

LASSO được sử dụng rộng rãi trong các lĩnh vực mà khả năng diễn giải mô hình và lựa chọn tính năng là quan trọng, chẳng hạn như tin sinh học, tâm lý học và kinh tế. Tuy nhiên, hạn chế của nó nằm ở việc lựa chọn ngẫu nhiên giữa các biến có mối tương quan cao và các ước tính có khả năng sai lệch khi λ lớn.

❖ **Khi nào nên sử dụng hồi quy LASSO?**

Hồi quy Lasso đặc biệt hữu ích khi xử lý các tập dữ liệu nhiều chiều, trong đó số lượng yếu tố dự đoán (đặc điểm) lớn và nhiều trong số chúng có thể không liên quan hoặc dư thừa. Trong những trường hợp như vậy, các kỹ thuật hồi quy tuyến tính truyền thống có thể phù hợp quá mức với dữ liệu và không thể khái quát hóa tốt dữ liệu mới.

Hồi quy Lasso có thể giúp giảm số lượng yếu tố dự đoán và chọn những yếu tố quan trọng nhất, từ đó cải thiện độ chính xác và khả năng diễn giải của mô hình. Điều này đặc biệt hữu ích trong những tình huống tập trung vào việc tìm hiểu mối quan hệ giữa các yếu tố dự đoán và biến mục tiêu, thay vì chỉ đơn giản là dự đoán kết quả.

❖ **Khi nào không nên sử dụng hồi quy LASSO?**

Mặc dù hồi quy Lasso có thể là một kỹ thuật mạnh mẽ để lựa chọn và chính quy hóa tính năng trong các mô hình hồi quy tuyến tính, nhưng nó có thể không phải lúc nào cũng là lựa chọn tốt nhất cho mọi tình huống. Dưới đây là một số tình huống mà hồi quy Lasso có thể không phải là cách tiếp cận tốt nhất:

- Cỡ mẫu nhỏ
- mối quan hệ phi tuyến
- Các yếu tố dự đoán tương quan
- Dự đoán phân loại
- Ngoại lệ

2.2.3.7 Kết luận

Tóm lại, hồi quy Lasso là một kỹ thuật mạnh mẽ và linh hoạt để lựa chọn và chính quy hóa tính năng trong các mô hình hồi quy tuyến tính. Nó cung cấp một số lợi thế so với hồi quy tuyến tính truyền thống, bao gồm lựa chọn tính năng, chính quy hóa, độ thưa thớt, cân bằng phương sai sai lệch, khả năng diễn giải và tính linh hoạt. Hồi quy Lasso đặc biệt hữu ích khi xử lý các tập dữ liệu nhiều chiều, trong đó số lượng yếu tố dự đoán lớn và nhiều trong số chúng có thể không liên quan hoặc dư thừa.

2.2.4 Phương pháp Ridge Regression

2.2.4.1 Giới thiệu

Hồi quy Ridge là một phương pháp ước tính các hệ số của mô hình hồi quy bội trong các kịch bản trong đó các biến độc lập có mối tương quan. Nó đã được sử dụng trong nhiều lĩnh vực bao gồm kinh tế lượng, hóa học và kỹ thuật. Còn được gọi là chính quy hóa Tikhonov, được đặt theo tên của Andrey Tikhonov, đây là một phương pháp chính quy hóa các vấn đề đặt ra sai lầm. Nó đặc biệt hữu ích để giảm thiểu vấn đề đa cộng tuyến trong hồi quy tuyến tính, thường xảy ra trong các mô hình có số lượng tham số lớn. Nói chung, phương

pháp này mang lại hiệu quả được cải thiện trong các vấn đề ước lượng tham số để đổi lấy một lượng sai lệch có thể chấp nhận được (xem sự đánh đổi độ lệch-phương sai).

Hồi quy sườn được phát triển như một giải pháp khả thi cho sự thiếu chính xác của các công cụ ước lượng bình phương nhỏ nhất khi mô hình hồi quy tuyến tính có một số biến độc lập đa cộng tuyến (tương quan cao) - bằng cách tạo một công cụ ước tính hồi quy sườn (RR). Điều này cung cấp ước tính các tham số đường vân chính xác hơn, vì phương sai và ước lượng bình phương trung bình của nó thường nhỏ hơn các ước lượng bình phương nhỏ nhất được rút ra trước đó.

2.2.4.2 Mô hình hồi quy Ridge Regression

Giả định dữ liệu đầu vào bao gồm N quan sát là những cặp các biến đầu vào và biến mục tiêu (x_1, y_1), (x_2, y_2), ..., (x_n, y_n). Quá trình hồi qui mô hình sẽ tìm kiếm một véc tơ hệ số ước lượng $w = [w_0, w_1, ..., w_p]$ sao cho tối thiểu hoá hàm mất mát dạng MSE:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\intercal \mathbf{x}_i) = \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2$$

Hàm mất mát cũng chính là mục tiêu tối ưu khi huấn luyện mô hình. Dữ liệu đầu vào X và y được xem như là cố định và biến số của bài toán tối ưu chính là các giá trị trong véc tơ w.

Giá trị hàm mất mát MSE chính là trung bình của tổng bình phương phần dư. Phần dư chính là chênh lệch giữa giá trị thực tế và giá trị dự báo. Tối thiểu hoá hàm mất mát nhằm mục đích làm cho giá trị dự báo ít chênh lệch so với giá trị thực tế, giá trị thực tế còn được gọi là ground truth. Trước khi huấn luyện mô hình chúng ta chưa thực sự biết véc tơ hệ số w là gì. Chúng ta chỉ có thể đặt ra một giả thuyết về dạng hàm dự báo (trong trường hợp này là phương trình dạng tuyến tính) và các hệ số hồi qui tương ứng. Chính vì vậy mục đích của tối thiểu hoá hàm mất mát là để tìm ra tham số w phù hợp nhất mô tả một cách khái quát quan hệ dữ liệu giữa biến đầu vào X với biến mục tiêu y trên tập huấn luyện

2.2.4.3 Sự thay đổi của hàm mất mát trong Ridge Regression

Hàm mất mát trong hồi qui Ridge sẽ có sự thay đổi so với hồi qui tuyến tính đó là thành phần điều chuẩn (regularization term) được cộng thêm vào hàm mất mát như sau:

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \alpha||\mathbf{w}||_2^2 \\ &= \frac{1}{N}||\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \underbrace{\alpha R(\mathbf{w})}_{\text{regularization term}}\end{aligned}$$

Trong phương trình trên thì $\boldsymbol{\alpha \geq 0.\frac{1}{}||\overline{X}w - y||^2}$ chính là tổng bình

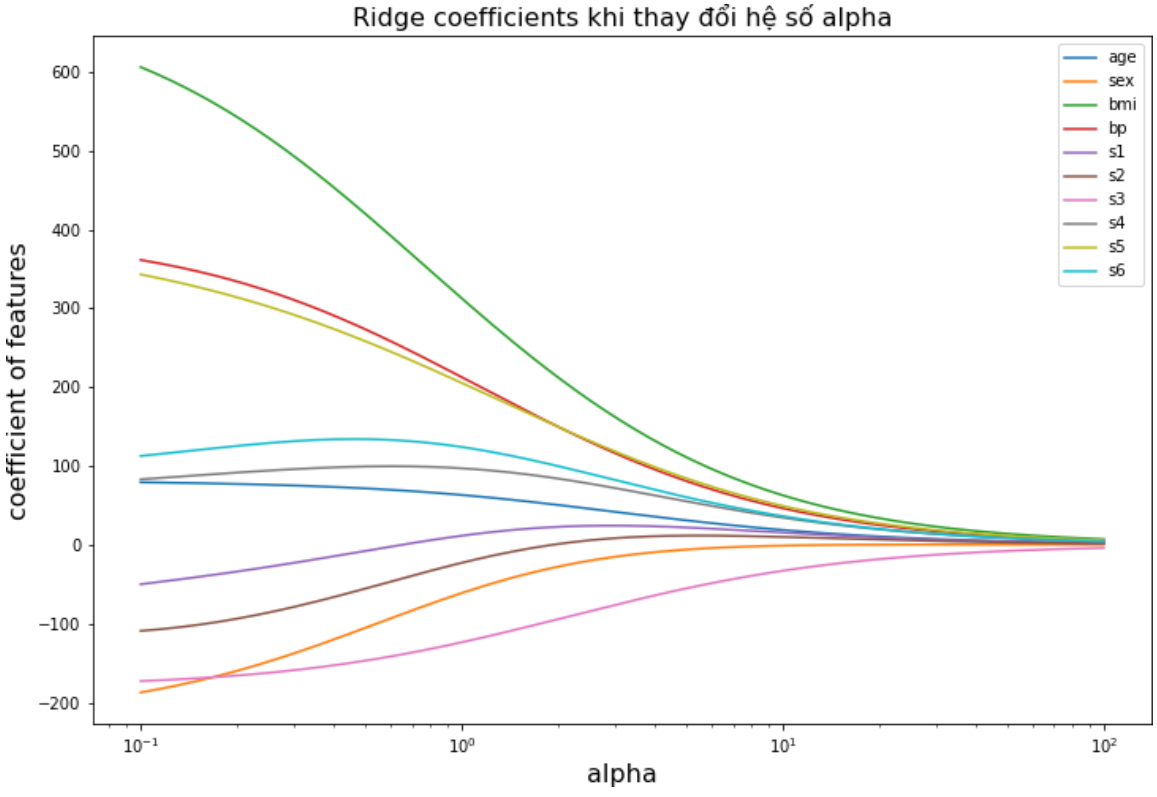
N 2

phương phần dư và $\boldsymbol{\alpha}\ ||\mathbf{w}||_2^2$ đại diện cho *thành phần điều chuẩn*.

Bài toán tối ưu hàm mất mát của hồi qui Ridge về bản chất là tối ưu song song hai thành phần bao gồm tổng bình phương phần dư và thành phần điều chuẩn. Hệ số α có tác dụng điều chỉnh độ lớn của thành phần điều chuẩn tác động lên hàm mất mát.

- Trường hợp **α =0**, thành phần điều chuẩn bị tiêu giảm và chúng ta quay trở về bài toán hồi qui tuyến tính.
- Trường hợp **α nhỏ** thì vai trò của thành phần điều chuẩn trở nên ít quan trọng. Mức độ kiểm soát quá khớp của mô hình sẽ trở nên kém hơn.
- Trường hợp **α lớn** chúng ta muốn gia tăng mức độ kiểm soát lên độ lớn của các hệ số ước lượng và qua đó giảm bớt hiện tượng qúa khớp.

Khi tăng dần hệ số α thì hồi qui Ridge sẽ có xu hướng thu hẹp hệ số ước lượng từ mô hình. Chúng ta sẽ thấy rõ thông qua ví dụ mẫu bên dưới.



Hình 2. 4 Sự thay đổi của độ lớn các hệ số ước lượng (coefficient of features) theo hệ số điều chuẩn α. Khi tăng dần độ lớn của α thì độ lớn của hệ

2.2.4.4 Nghiệm tối ưu của Ridge Regression

Giải bài toán tối ưu hàm mục tiêu của hồi qui Ridge theo đạo hàm bậc nhất của véc tơ w

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{N}\frac{\partial \|\bar{\mathbf{X}}\mathbf{w}-\mathbf{y}\|_2^2}{\partial \mathbf{w}} + \alpha\frac{\partial \|\mathbf{w}\|_2^2}{\partial \mathbf{w}} \\ &= \frac{2}{N}\bar{\mathbf{X}}^\intercal(\bar{\mathbf{X}}\mathbf{w}-\mathbf{y}) + 2\alpha\mathbf{w} \\ &= \frac{2}{N}[(\bar{\mathbf{X}}^\intercal\bar{\mathbf{X}} + N\alpha\mathbf{I})\mathbf{w} - \bar{\mathbf{X}}^\intercal\mathbf{y}] \\ &= 0\end{aligned}$$

Thật vậy, từ dòng 1 suy ra dòng 2 là vì theo công thức product-rule trong matrix caculus thì:

$$\nabla_{\mathbf{w}} f(\mathbf{w})^\intercal g(\mathbf{w}) = \nabla_{\mathbf{w}}(f)g + \nabla_{\mathbf{w}}(g)f$$

Nếu thay f(w) = g(w) = $\bar{\boldsymbol{X}}\boldsymbol{w} - \boldsymbol{y}$ ta suy ra:

$$\begin{aligned}\frac{\partial \|\bar{\mathbf{X}}\mathbf{w}-\mathbf{y}\|_2^2}{\partial \mathbf{w}} &= \frac{\partial(\bar{\mathbf{X}}\mathbf{w}-\mathbf{y})^\intercal(\bar{\mathbf{X}}\mathbf{w}-\mathbf{y})}{\partial \mathbf{w}} \\ &= \frac{2\partial(\bar{\mathbf{X}}\mathbf{w}-\mathbf{y})}{\partial \mathbf{w}}(\bar{\mathbf{X}}\mathbf{w}-\mathbf{y}) \\ &= 2\bar{\mathbf{X}}^\intercal(\bar{\mathbf{X}}\mathbf{w}-\mathbf{y})\end{aligned}$$

Như vậy ta nhận thấy dòng 1 suy ra dòng 2 là hoàn toàn đúng.

Ở dòng thứ 3 chúng ta áp dụng thêm một tính chất **Iw = w** trong đó **I** là ma trận đơn vị.

Sau cùng nghiệm của đạo hàm bậc nhất trở thành:

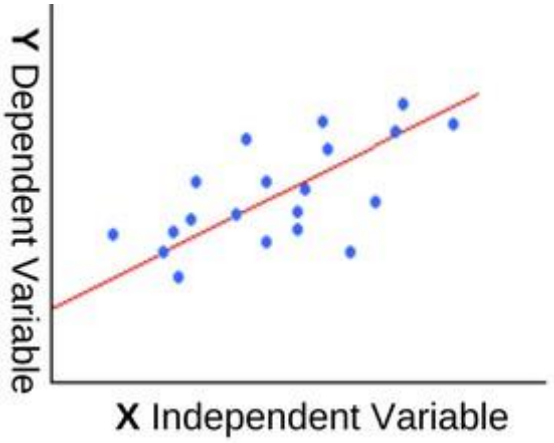
$$\begin{aligned}\frac{2}{N}[(\bar{\mathbf{X}}^\intercal\bar{\mathbf{X}} + N\alpha\mathbf{I})\mathbf{w} - \bar{\mathbf{X}}^\intercal\mathbf{y}] &= 0 \\ (\bar{\mathbf{X}}^\intercal\bar{\mathbf{X}} + N\alpha\mathbf{I})\mathbf{w} &= \bar{\mathbf{X}}^\intercal\mathbf{y} \\ \mathbf{w} &= (\bar{\mathbf{X}}^\intercal\bar{\mathbf{X}} + N\alpha\mathbf{I})^{-1}\bar{\mathbf{X}}^\intercal\mathbf{y}\end{aligned}$$

2.2.5 Phương pháp Linear Regression

2.2.5.1 Giới thiệu

"Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...

Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Trong không gian nhiều hơn 3 chiều, khái niệm mặt phẳng không còn phù hợp nữa, thay vào đó, một khái niệm khác ra đời được gọi là siêu mặt phẳng (hyperplane)



Hình 2. 5Mối quan hệ tuyến tính giữa biến đầu ra (y) và biến dự đoán

Biểu đồ trên trình bày mối quan hệ tuyến tính giữa các biến đầu ra (y) và biến dự đoán (X). Đường màu đỏ được gọi là đường thẳng phù hợp nhất. Dựa trên các điểm dữ liệu đã cho, mô hình sẽ cố gắng vẽ một đường phù hợp nhất với các điểm đó.

2.2.5.2 Công thức/Mô hình tuyến tính

Trong thực tế, chúng ta chỉ có thể có một biến độc lập X ảnh hưởng đến biến phụ thuộc Y. Hoặc có thể xảy ra trường hợp có nhiều biến độc lập ảnh hưởng đến Y.

Dựa trên cách tiếp cận này, có hai loại hồi quy tuyến tính chính:

❖ ***Mô hình hồi quy tuyến tính đơn biến:***

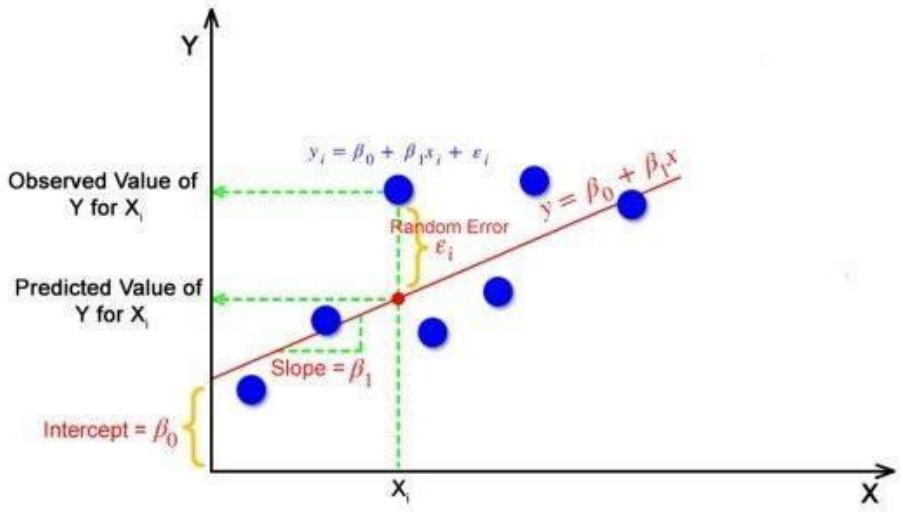
Hồi quy tuyến tính đơn giản có nghĩa là chỉ có một biến X độc lập mà những thay đổi này dẫn đến các giá trị khác nhau cho Y.

Công thức:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Trong đó:

- Y là biến phụ thuộc (cũng được gọi là biến mục tiêu hoặc biến phản hồi).
- X là biến độc lập (cũng được gọi là biến giải thích hoặc biến đầu vào).
- β_0 và β_1 là các hệ số hồi quy (cũng được gọi là hệ số góc và hệ số chặn).
- ε là sai số ngẫu nhiên (cũng được gọi là sai số hồi quy).



Hình 2. 6 Tính toán hồi quy đơn biến

Mục tiêu của thuật toán hồi quy tuyến tính là lấy giá trị tốt nhất cho β_0 và β_1 để tìm ra đường thẳng phù hợp nhất. Đường thẳng phù hợp nhất là đường thẳng có ít lỗi nhất, nghĩa là sai số giữa giá trị dự đoán và giá trị thực tế phải ở mức tối thiểu.

❖ ***Mô hình hồi quy tuyến tính đa biến (tuyến tính bội)***

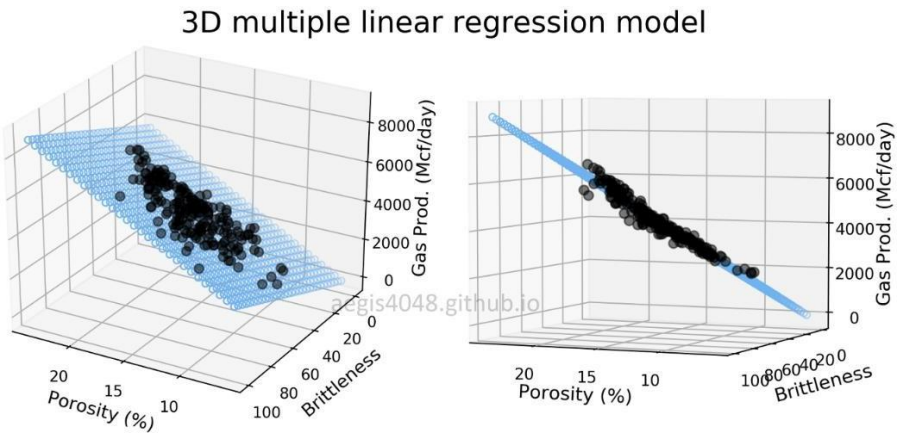
Mô hình hồi quy tuyến tính bội là loại phân tích hồi quy tuyến tính phổ biến nhất. Nó được sử dụng để thể hiện mối quan hệ giữa một biến phụ thuộc và hai hoặc nhiều biến độc lập. Trên thực tế, mô hình hồi quy tuyến tính đơn giản là một dạng đặc biệt của hồi quy tuyến tính đa biến.

Phương trình cho mô hình tuyến tính đa biến có thể được biểu diễn dưới dạng:

$$y = a + b_1x_1 + b_2x_2 + \ldots + b_nx_n + \varepsilon$$

Trong phương trình này:

- y là biến phụ thuộc mà chúng ta muốn dự đoán hoặc giải thích.
- a là điểm giao với trục tung
- b_1, b_2, …, b_n là hệ số hồi quy ứng với từng biến độc lập, biểu thị độ ảnh hưởng của chúng lên biến phụ thuộc
- x_1, x_2, …, x_n là các biến độc lập được sử dụng để dự đoán y
- ε là sai số ngẫu nhiên, biểu thị các yếu tố không thể dự đoán được trong mô hình (ε thực tế không tính được).



Hình 2. 7 Minh họa siêu phẳng phù hợp nhất của mô hình tuyến tính bội

Mô hình hồi quy tuyến tính đa biến xử lý nhiều biến độc lập. Nó nhằm mục đích tìm ra một siêu phẳng phù hợp nhất với mối quan hệ giữa nhiều biến độc lập và biến phụ thuộc.

2.2.5.3 Phương pháp đánh giá kết quả

Sức mạnh của bất kỳ mô hình hồi quy tuyến tính nào cũng có thể được đánh giá bằng các số liệu đánh giá khác nhau. Các số liệu đánh giá này thường cung cấp thước đo về mức độ hiệu quả của các kết quả đầu ra được quan sát bởi mô hình.

Các số liệu được sử dụng nhiều nhất là:

1. Hệ số xác định hoặc R-Squared (R2)
2. Lỗi bình phương trung bình gốc (RSME) và lỗi chuẩn dư (RSE)

❖ ***Hệ số xác định hoặc R-Squared (R2)***

R-Squared là con số giải thích mức độ biến thiên được giải thích/nắm bắt bởi mô hình đã phát triển. Nó luôn nằm trong khoảng từ 0 & 1. Nhìn chung, giá trị của R bình phương càng cao thì mô hình càng phù hợp với dữ liệu.

Về mặt toán học nó có thể được biểu diễn dưới dạng:

R2 = 1 – (RSS/TSS)

- **Tổng bình phương dư (RSS)** được định nghĩa là tổng bình phương của phần dư cho mỗi điểm dữ liệu trong biểu đồ/dữ liệu. Nó là thước đo sự khác biệt giữa kết quả dự kiến và kết quả thực tế được quan sát.

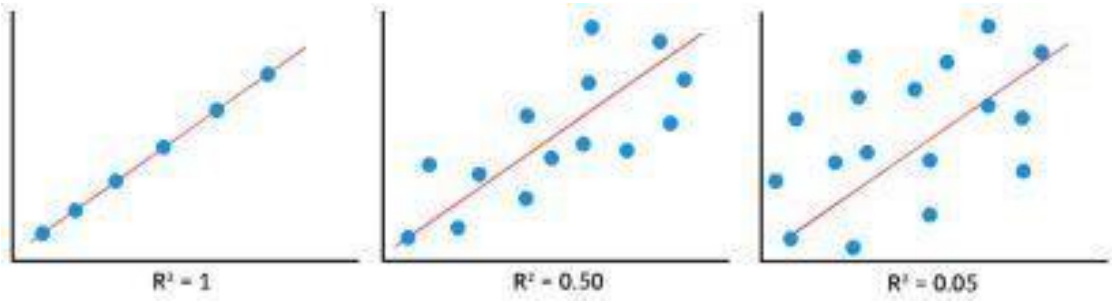
$$RSS = \sum_{i=1}^{n}(y_i - b_0 - b_1 x_i)^2$$

- **Tổng bình phương (TSS)** được định nghĩa là tổng sai số của các điểm dữ liệu so với giá trị trung bình của biến phản hồi. Về mặt toán học TSS là:

$$TSS = \sum (y_i - \bar{y}_i)^2$$

Trong đó y hat là giá trị trung bình của các điểm dữ liệu mẫu.

Ý nghĩa của bình phương R được thể hiện bằng các hình sau:



Hình 2. 8 Ý nghĩa của bình phương R

❖ *Lỗi bình phương gốc*

Sai số bình phương trung bình gốc là căn bậc hai của phương sai của phần dư. Nó chỉ định mức độ phù hợp tuyệt đối của mô hình với dữ liệu, tức là mức độ gần của các điểm dữ liệu được quan sát với các giá trị dự đoán. Về mặt toán học nó có thể được biểu diễn dưới dạng:

$$RMSE = \sqrt{\frac{RSS}{n}} = \sqrt{\sum_{i=1}^{n}\left(y_i^{Actual} - y_i^{Predicted}\right)^2 / n}$$

Để làm cho ước tính này không thiên vị, người ta phải chia tổng bình

phương số dư cho bậc tự do thay vì chia tổng số điểm dữ liệu trong mô hình. Thuật ngữ này khi đó được gọi là Lỗi chuẩn dư (RSE). Về mặt toán học nó có thể được biểu diễn dưới dạng.

$$RSE = \sqrt{\frac{RSS}{df}} = \sqrt{\sum_{i=1}^{n} \left(y_i^{Actual} - y_i^{Predicted} \right)^2 / (n-2)}$$

R bình phương là thước đo tốt hơn RSME. Bởi vì giá trị của Lỗi bình phương trung bình gốc phụ thuộc vào đơn vị của các biến (tức là nó không phải là thước đo chuẩn hóa), nên nó có thể thay đổi khi có sự thay đổi về đơn vị của các biến.

2.2.5.4 Ưu điểm của hồi quy tuyến tính

Hồi quy tuyến tính là một trong những kỹ thuật thống kê cơ bản và được sử dụng rộng rãi nhất trong mô hình dự đoán. Dưới đây là những ưu điểm chính của nó:

1. Tính đơn giản và dễ hiểu:

Các mô hình hồi quy tuyến tính rất đơn giản và dễ hiểu. Mối quan hệ giữa các biến độc lập và biến phụ thuộc được biểu diễn bằng một đường thẳng, giúp dễ hiểu và giải thích các dự đoán của mô hình.

2. Ít bị trang bị quá mức:

Trong trường hợp có một hoặc một vài yếu tố dự đoán, hồi quy tuyến tính có thể ít bị khớp quá mức so với các mô hình phức tạp hơn. Điều này đặc biệt có giá trị khi mối quan hệ cơ bản trong dữ liệu thực sự là tuyến tính hoặc gần tuyến tính.

3. Cơ sở để hiểu các mô hình phức tạp hơn:

Hồi quy tuyến tính đặt nền tảng để hiểu các thuật toán phức tạp hơn. Nhiều kỹ thuật tiên tiến có thể được coi là phần mở rộng hoặc biến thể của hồi quy tuyến tính, khiến nó trở thành một khối xây dựng thiết yếu trong việc học mô hình dự đoán.

4. Tính toán hiệu quả:

Các mô hình hồi quy tuyến tính có hiệu quả tính toán cao, đòi hỏi ít tài nguyên tính toán hơn. Điều này làm cho chúng có khả năng mở rộng cao đối với các tập dữ liệu lớn và phù hợp với các tình huống mà tốc độ đào tạo mô hình là rất quan trọng.

5. Tốt cho dự báo dự đoán:

Hồi quy tuyến tính có thể là một công cụ hiệu quả để dự báo mang tính dự đoán, đặc biệt khi dữ liệu cho thấy xu hướng tuyến tính. Điều này phổ biến trong nhiều lĩnh vực khác nhau như kinh tế, sinh học và kỹ thuật.

2.2.5.5 Nhược điểm của hồi quy tuyến tính

Mặc dù có những ưu điểm nhưng hồi quy tuyến tính cũng có những hạn chế:

1. Giả sử mối quan hệ tuyến tính:

Hồi quy tuyến tính giả định mối quan hệ tuyến tính giữa các biến độc lập

và biến phụ thuộc. Đây có thể là một hạn chế lớn nếu mối quan hệ thực tế phức tạp hoặc phi tuyến tính.

2. Nhạy cảm với các ngoại lệ:

Các mô hình hồi quy tuyến tính rất nhạy cảm với các ngoại lệ. Một vài ngoại lệ có thể ảnh hưởng đáng kể đến kết quả, làm cho mô hình kém tin cậy hơn.

3. Dễ xảy ra hiện tượng đa cộng tuyến:

Đa cộng tuyến xảy ra khi các biến độc lập có mối tương quan cao. Điều này có thể làm sai lệch các mối quan hệ ước tính và làm giảm độ tin cậy của mô hình.

4. Giả định về tính đồng nhất:

Hồi quy tuyến tính giả định rằng phương sai của các số hạng sai số là không đổi trên tất cả các cấp độ của các biến độc lập (tính đồng nhất). Nếu giả định này bị vi phạm (tính không đồng nhất), nó có thể dẫn đến các ước tính không hiệu quả.

5. Giới hạn ở các mối quan hệ tuyến tính:

Nó chỉ có thể mô hình hóa các mối quan hệ tuyến tính, khiến nó không phù hợp để mô hình hóa các mẫu phức tạp hơn trong dữ liệu.

6. Nguy cơ đơn giản hóa các mối quan hệ quá mức:

Khi cố gắng điều chỉnh một mô hình tuyến tính, có nguy cơ đơn giản hóa quá mức mối quan hệ giữa các biến, điều này có thể dẫn đến những dự đoán không chính xác.

2.2.5.6 Khả năng ứng dụng và hạn chế của hồi quy tuyến tính

Hồi quy tuyến tính được áp dụng rộng rãi trong các lĩnh vực mà mối quan hệ giữa các biến là tuyến tính hoặc gần tuyến tính. Nó thường được sử dụng trong kinh doanh, kinh tế, khoa học xã hội và một số khoa học tự nhiên. Tuy nhiên, hạn chế của nó nằm ở chỗ không có khả năng xử lý các mối quan hệ phi tuyến tính, dễ bị ảnh hưởng bởi các ngoại lệ và các vấn đề về đa cộng tuyến.

❖ **Khi nào nên sử dụng hồi quy tuyến tính?**

Hồi quy tuyến tính thích hợp nhất trong trường hợp dữ liệu thể hiện mối quan hệ tuyến tính. Nó phù hợp nhất để phân tích dự đoán trong trường hợp dữ liệu không quá phức tạp và mối quan hệ giữa các biến được hiểu rõ và có thể gần đúng bằng mô hình tuyến tính.

❖ **Khi nào bạn không nên sử dụng Hồi quy tuyến tính?**

Hồi quy tuyến tính có thể không phù hợp trong trường hợp mối quan hệ giữa các biến vốn không tuyến tính, khi xử lý các biến độc lập có mối tương quan cao hoặc khi dữ liệu chứa các giá trị ngoại lệ đáng kể.

2.2.5.7 Kết luận

Tóm lại, hồi quy tuyến tính, với tính đơn giản, dễ hiểu và hiệu quả, vẫn là nền tảng trong mô hình thống kê và phân tích dự đoán. Mặc dù nó mang lại nhiều lợi ích nhưng hiệu quả của nó phụ thuộc vào bản chất của dữ liệu và các mối quan hệ bên trong. Hiểu được những hạn chế của nó là rất quan trọng để có thể áp dụng nó một cách thích hợp và để có được những hiểu biết chính xác, đáng tin cậy từ việc sử dụng nó.

2.2.6 Lựa chọn phương pháp

Phương pháp phân tích hồi quy tuyến tính (Linear Regression) là sự kết hợp của tính đơn giản, khả năng ước lượng mối quan hệ tuyến tính, khả năng dự báo cùng với khả năng phân tích định lượng. Phương pháp này sẽ giúp ta đạt được mục tiêu nghiên cứu và trả lời những câu hỏi quan trọng. Vì vậy chúng em lựa chọn phương pháp hồi quy tuyến tính để giải quyết đề bài.

1. Quan hệ tuyến tính: Linear Regression giả định một mối quan hệ tuyến tính giữa biến đầu vào và biến đầu ra. Trong bài toán dự báo giá xe, có thể có một mối quan hệ tuyến tính giữa các yếu tố như đặc điểm kỹ thuật, tuổi của xe, số lượng dặm đã đi, v.v., và giá bán. Vì vậy, Linear Regression có thể phù hợp để mô hình hóa mối quan hệ này.
2. Đơn giản và dễ hiểu: Linear Regression là một phương pháp đơn giản và dễ hiểu. Nó chỉ sử dụng một hàm tuyến tính để dự đoán giá xe dựa trên các biến đầu vào. Điều này làm cho việc triển khai và diễn giải mô hình dễ dàng hơn so với các phương pháp phức tạp hơn.
3. Tính toán hiệu quả: Linear Regression có thể được tính toán nhanh chóng và hiệu quả. Việc dự báo giá xe có thể yêu cầu xử lý một lượng lớn dữ liệu, bao gồm nhiều biến đầu vào. Linear Regression có thể xử lý tập dữ liệu lớn một cách hiệu quả và có thể được sử dụng trong thời gian thực.

2.3 Công cụ phục vụ thực hiện bài toán

2.3.1 Python

Hình 2.10 Ngôn ngữ lập trình Python

Python là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, thường được sử dụng để xây dựng trang web và phần mềm, tự động hoá các tác vụ và tiến hành phân tích dữ liệu. Với sự phát triển của khoa học dữ liệu hiện nay, Python lại càng được ứng dụng rộng rãi hơn trong ngành Data Analyst. Với thư viện đa dạng trong các lĩnh vực như khai thác dữ liệu (Scrapy, BeautifulSoup4, …), xử lý dữ liệu và mô hình hóa (Pandas, Scikit-learn, …), trực quan hóa dữ liệu (Matplotlib, Plotly, …) thì đây là một lựa chọn tuyệt vời để phân tích dữ liệu. Tuy nhiên bên cạnh những ưu điểm về thư viện cũng như cộng đồng lập trình đông đảo, Python vẫn vướng phải một số nhược điểm, đó là bị giới hạn tốc độ, mức tiêu thụ bộ nhớ cao và không phải là một ngôn ngữ được hỗ trợ nhiều cho môi trường di động.

2.3.2 R

Hình 2.11 Ngôn ngữ lập trình R

Ngôn ngữ R là một ngôn ngữ lập trình và môi trường tính toán thống kê phổ biến trong lĩnh vực phân tích dữ liệu và thống kê. Nó cung cấp nền tảng mạnh mẽ cho việc thực hiện các phân tích thống kê, xử lý dữ liệu và tạo biểu đồ. R cũng là một cộng đồng mã nguồn mở lớn, điều này có nghĩa là người dùng có thể dễ dàng chia sẻ mã nguồn, gói phân tích và kiến thức với nhau. Vậy nên việc phân tích dữ liệu trên R cũng rất thuận tiện khi có đầy đủ các thư viện về phân tích dữ liệu và có khả năng tích hợp tốt với môi trường nghiên cứu khoa học. Dù vậy, R vẫn có một vài nhược điểm nhất định. Phổ biến trong số đấy là sự phức tạp của ngôn ngữ khi lập trình viên mới bắt đầu tiếp xúc và sử dụng, xử lý dữ liệu lớn không tốt so với nhiều ngôn ngữ khác và hiệu suất không phải lúc nào cũng ổn định.

2.3.3 Lựa chọn công cụ

Cả Python và R đều là hai ngôn ngữ phổ biến được sử dụng cho phân tích dữ liệu và thống kê. Việc lựa chọn sử dụng ngôn ngữ nào phụ thuộc vào nhiều yếu tố như mục tiêu, kinh nghiệm cá nhân, loại dữ liệu đang làm việc, và các thư viện hỗ trợ cần sử dụng. Sau đây là bảng so sánh để đưa ra quyết định lựa chọn công cụ phục vụ giải quyết bài toán:

Bảng 1: So sánh ngôn ngữ Python và R

Ngôn ngữ	Python	R
Ưu điểm	- ***Đa năng***: Python không chỉ giới hạn trong phân tích dữ liệu, mà còn có thể sử dụng cho nhiều mục đích khác như phát triển ứng dụng, web, automation, và machine learning. - ***Thư viện phong phú***: Có nhiều thư viện mạnh mẽ giúp thực hiện các tác vụ phân tích và xử lý dữ liệu một cách hiệu quả. - ***Cộng đồng lớn***: python có cộng đồng lớn giúp việc chia sẻ, học hỏi dễ dàng hơn.	- ***Thống kê chuyên sâu:*** R được thiết kế đặc biệt cho thống kê và phân tích dữ liệu, với nhiều gói như dplyr, ggplot2, tidyr, và lubridate giúp thực hiện các tác vụ phân tích chi tiết. - ***Biểu đồ phức tạp***: Gói ggplot2 trong R cho phép tạo ra biểu đồ phức tạp và tùy chỉnh một cách dễ dàng.
Nhược điểm	- ***Thống kê chuyên sâu***: Mặc dù Python có thư viện thống kê tốt, nhưng R vẫn là lựa chọn phổ biến hơn trong các nghiên cứu thống kê và phân tích dữ liệu chuyên sâu.	- ***Thiếu phổ biến:*** R có tính chuyên môn hơn so với Python. - ***Sử dụng bộ nhớ:*** R có xu hướng sử dụng nhiều bộ nhớ hơn so với Python. - ***Quản lý mã nguồn***: R không thể sử dụng mã nguồn mở rộng và phân chia mã nguồn dễ dàng như Python. Việc quản lý và tái sử dụng mã có thể trở nên khó khăn hơn khi dự án phát triển

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1 Dữ liệu thực nghiệm

Cái gọi là xe cũ có cơ sở thị trường rất lớn. Nhiều người cân nhắc mua Ô tô đã qua sử dụng thay vì mua ô tô mới vì điều đó khả thi và là khoản đầu tư tốt hơn.

Lý do chính cho thị trường khổng lồ này là khi bạn mua một chiếc Ô tô mới và bán nó chỉ trong một ngày nữa mà không có bất kỳ khoản nợ nào, giá ô tô sẽ giảm 30%.

Ngoài ra còn có nhiều hành vi lừa đảo trên thị trường, không chỉ bán sai sản phẩm mà còn có thể gây nhầm lẫn về giá cả.

Vì vậy, ở đây nhóm chúng em đã sử dụng tập dữ liệu sau đây để Dự đoán giá của bất kỳ chiếc ô tô đã qua sử dụng nào.

Cụ thể thông tin như sau:

- Tên bộ dữ liệu: Used Cars Price Prediction
- Nguồn: https://www.kaggle.com/datasets/avikasliwal/used-cars-price-prediction

Dữ liệu 15 hàng đầu của dataset:

0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
5	Hyundai EON LPG Era Plus Option	Hyderabad	2012	75000	LPG	Manual	First	21.1 km/kg	814 CC	55.2 bhp	5.0	NaN	2.35
6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	3.50
7	Toyota Innova Crysta 2.8 GX AT 8S	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmpl	2755 CC	171.5 bhp	8.0	21 Lakh	17.50
8	Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN	5.20
9	Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5.0	NaN	1.95
10	Maruti Ciaz Zeta	Kochi	2018	25692	Petrol	Manual	First	21.56 kmpl	1462 CC	103.25 bhp	5.0	10.65 Lakh	9.95
11	Honda City 1.5 V AT Sunroof	Kolkata	2012	60000	Petrol	Automatic	First	16.8 kmpl	1497 CC	116.3 bhp	5.0	NaN	4.49
12	Maruti Swift VDI BSIV	Jaipur	2015	64424	Diesel	Manual	First	25.2 kmpl	1248 CC	74 bhp	5.0	NaN	5.60
13	Land Rover Range Rover 2.2L Pure	Delhi	2014	72000	Diesel	Automatic	First	12.7 kmpl	2179 CC	187.7 bhp	5.0	NaN	27.00
14	Land Rover Freelander 2 TD4 SE	Pune	2012	85000	Diesel	Automatic	Second	0.0 kmpl	2179 CC	115 bhp	5.0	NaN	17.50

Hình 3. 1 15 hàng dữ liệu đầu tiên trong bộ dữ liệu

Mô tả thông tin các cột dữ liệu trong dataset:

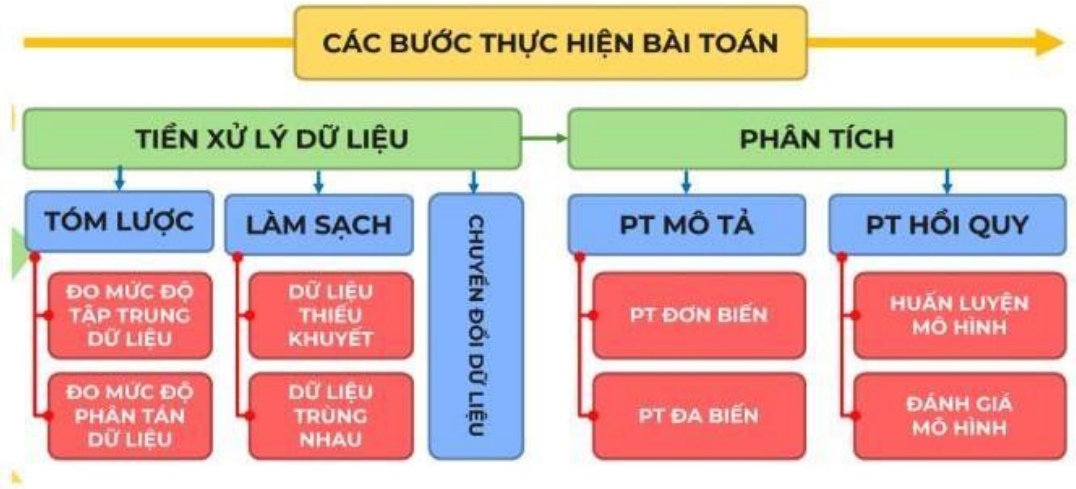
STT	Tên cột	Mô tả
1	**Name**	Thương hiệu và mẫu xe.
2	**Location**	Địa điểm nơi chiếc xe đang được bán hoặc có sẵn để mua.
3	**Year**	Năm hoặc phiên bản của mẫu xe.
4	**Kilometers_Driven**	Tổng số km mà chủ sở hữu trước đã lái xe tính bằng KM.
5	**Fuel_Type**	Loại nhiên liệu mà ô tô sử dụng. (Xăng, Diesel, Điện, CNG, LPG)
6	**Transmission**	Loại hộp số được ô tô sử dụng. (Tự động/Thủ công)
7	**Owner_Type**	Đời sở hữu thứ mấy của xe.
8	**Mileage**	Quãng đường tiêu chuẩn mà hãng xe cung cấp tính bằng kmpl hoặc km/kg.
9	**Engine**	Thể tích dịch chuyển của động cơ tính bằng CC

```
dataFrame.info()
print('Shape tập dữ liệu: ', dataFrame.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6019 entries, 0 to 6018
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         6019 non-null   int64
 1   Name               6019 non-null   object
 2   Location           6019 non-null   object
 3   Year               6019 non-null   int64
 4   Kilometers_Driven  6019 non-null   int64
 5   Fuel_Type          6019 non-null   object
 6   Transmission       6019 non-null   object
 7   Owner_Type         6019 non-null   object
 8   Mileage            6017 non-null   object
 9   Engine             5983 non-null   object
 10  Power              5983 non-null   object
 11  Seats              5977 non-null   float64
 12  New_Price          824 non-null    object
 13  Price              6019 non-null   float64
dtypes: float64(2), int64(3), object(9)
memory usage: 658.5+ KB
Shape tập dữ liệu:  (6019, 14)
```

Bảng 2: Mô tả thông tin các cột dữ liệu trong dataset

3.2 Quy trình thực nghiệm



Hình 3. 2 Quy trình thực nghiệm

3.2.1 Đặt mục tiêu

- Phân tích mô tả để thể hiện mối quan hệ giữa các giá trị của dữ liệu, từ đó đánh giá được tương quan của ngành Khoa học dữ liệu.
- Phân tích hồi quy để dự báo giá nhà dựa theo mô hình hồi quytuyến tính.

3.2.2 Tiền xử lý dữ liệu

3.2.2.1 Phân tích các cột dữ liệu kiểu phi số

Vì có các cột không phải kiểu số, nên chúng ta sẽ trích xuất số lượng các giá trị duy nhất, cũng như số lượng của mỗi giá trị để xem liệu cột đó có mang ý nghĩa, ảnh hưởng đến giá bán của chiếc xe không:

```
def categories_counts(label_name):
    print(f'-----------------------------------------------')
    print(f'Số các giá trị duy nhất của cột {label_name}:')
    print(dataFrame[label_name].value_counts())

# Đếm các giá trị duy nhất của các cột có Dtype phi số
categories_counts('Location')
categories_counts('Fuel_Type')
categories_counts('Transmission')
categories_counts('Owner_Type')
```

```
-----------------------------------------------
Số các giá trị duy nhất của cột Location:
Mumbai        790
Hyderabad     742
Kochi         651
Coimbatore    636
Pune          622
Delhi         554
Kolkata       535
Chennai       494
Jaipur        413
Bangalore     358
Ahmedabad     224
Name: Location, dtype: int64
-----------------------------------------------
Số các giá trị duy nhất của cột Fuel_Type:
Diesel      3205
Petrol      2746
CNG           56
LPG           10
Electric       2
Name: Fuel_Type, dtype: int64
-----------------------------------------------
Số các giá trị duy nhất của cột Transmission:
Manual       4299
Automatic    1720
Name: Transmission, dtype: int64
-----------------------------------------------
Số các giá trị duy nhất của cột Owner_Type:
First             4929
Second             968
Third              113
Fourth & Above       9
Name: Owner_Type, dtype: int64
```

Hình 3. 3 Thông tin dữ liệu dạng phi số

Từ kết quả thu được, ta nhận xét:

- Cột "Location" chứa nhiều giá trị phi số, khó có thể chuyển đổi sang kiểu số.
- Cột "Fuel_Type", "Transmission", "Owner_Type" có thể coi như một dạng danh mục. Nếu cần sử dụng, ta có thể dùng loại từ điển để ánh xạ sang các giá trị kiểu số.

3.2.2.2 Kiểm tra mức độ sạch dữ liệu

a) Kiểm tra các giá trị khuyết, trùng lặp trong tập dữ liệu:

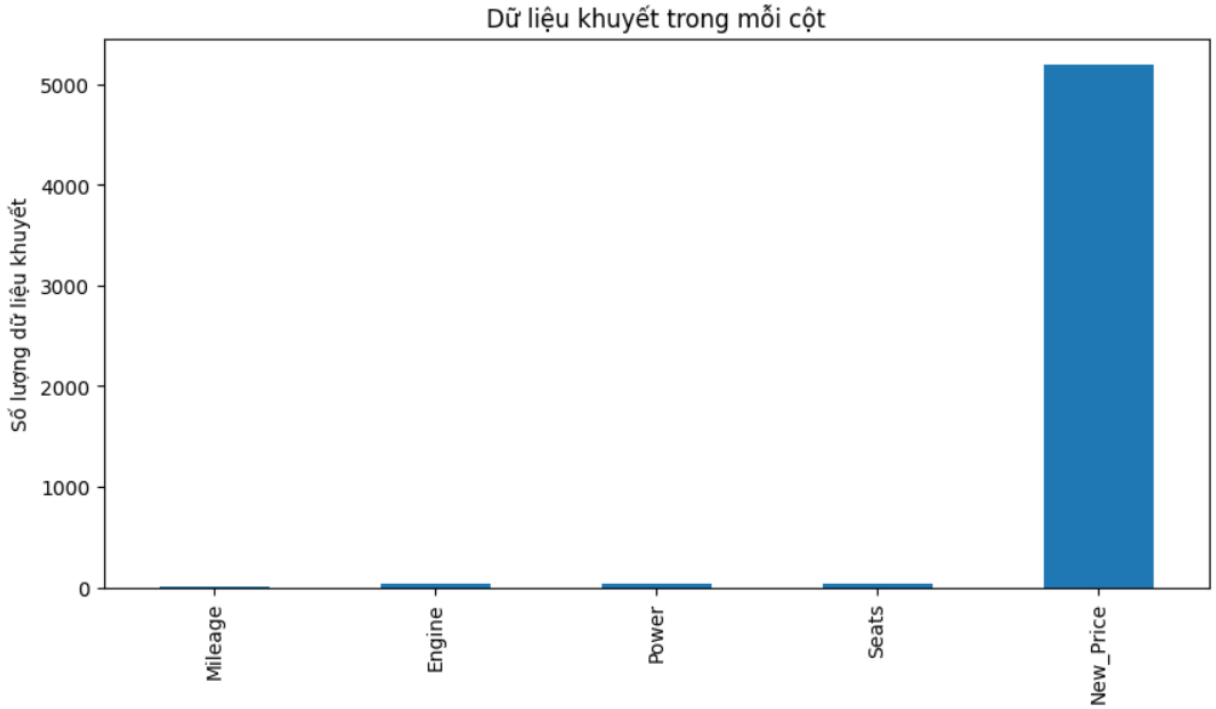
```python
def statistical_missing_data():
    missing_data = dataFrame.isnull().sum()
    duplicate_data = dataFrame.duplicated().sum()

    print("Số liệu thiếu trong mỗi cột:")
    print(missing_data)
    print("\nSố liệu trùng lặp:")
    print(duplicate_data)

# Thống kê dữ liệu khuyết
statistical_missing_data()
```

```
Số liệu thiếu trong mỗi cột:
Unnamed: 0              0
Name                    0
Location                0
Year                    0
Kilometers_Driven       0
Fuel_Type               0
Transmission            0
Owner_Type              0
Mileage                 2
Engine                 36
Power                  36
Seats                  42
New_Price            5195
Price                   0
dtype: int64

Số liệu trùng lặp:
0
```

Hình 3. 4 Thống kê dữ liệu khuyết trước tiền xử lý dữ liệu

b) Vẽ đồ thị cột thể hiện số lượng giá trị khuyết trong tập dữ liệu

```
def plt_statistics_missing_data(dataFrame):
    # Dữ liệu khuyết
    missing_data = dataFrame.isnull().sum()
    missing_data = missing_data[missing_data > 0]
    plt.figure(figsize=(10, 5))
    missing_data.plot(kind='bar')
    plt.title('Dữ liệu khuyết trong mỗi cột')
    plt.xlabel('Tên cột')
    plt.ylabel('Số lượng dữ liệu khuyết')
    plt.show()

    # Dữ liệu lặp
    duplicate_data = dataFrame.duplicated().sum()
    print("\nSố liệu trùng lặp:", duplicate_data)

# Thống kê dữ liệu khuyết
plt_statistics_missing_data(dataFrame)
```



Hình 3. 5 Biểu đồ cột thể hiện giá trị khuyết trong tập dữ liệu

Nhận xét:

- Cột "New_Price" có số lượng giá trị khuyết rất lớn, không mang nhiều ý nghĩa, cần loại bỏ.
- Các cột khác không có hoặc có số lượng giá khuyết ít. Vì vậy, ta có thể làm sạch bằng các lựa chọn như:
  - Loại bỏ các hàng có giá trị khuyết
  - Điền khuyết bằng giá trị phù hợp

3.2.2.3 Loại bỏ cột

Ý tưởng và lựa chọn các cột để xóa:

- Vì cột "**New_Price**" có rất nhiều phần tử chứa giá trị null nên ta sẽ bỏ cột này
- Ngoài ra, cột "**Name**" và "**Location**" chứa rất nhiều giá trị phi số, không thể chuyển đổi sang kiểu số nên ta sẽ loại bỏ các cột này.
- Cuối cùng, cột "**Unamed: 0**" biểu thị giá trị chỉ mục – không mang ý nghĩa nên ta cũng sẽ bỏ cột này.

```
[15] print('Shape tập dữ liệu trước khi xóa cột:\t', dataFrame.shape)

     # Xóa cột New_Price vì có nhiều ô chứa giá trị NULL
     dataFrame = dataFrame.drop('New_Price', axis=1)

     # Bỏ các cột không còn cần thiết
     dataFrame = dataFrame.drop(columns = ['Unnamed: 0', 'Name', 'Location'])

     print('Shape tập dữ liệu sau khi xóa cột:\t', dataFrame.shape)

     Shape tập dữ liệu trước khi xóa cột:     (6019, 14)
     Shape tập dữ liệu sau khi xóa cột:       (6019, 10)
```

```
# Kết quả sau khi xóa cột
dataFrame.head()
```

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	1.75
1	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	12.50
2	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	4.50
3	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	6.00
4	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	17.74

Hình 3. 6 Dữ liệu sau khi loại bỏ các cột

3.2.2.4 Trích xuất các giá trị số

Các cột "**Engine**", "**Power**", "**Mileage**" tuy là kiểu chuỗi vì chúng bao gồm các đơn vị đo, nhưng ta có thể loại bỏ các đơn vị đo để thu được giá trị thực sự.

Ở bước này, chúng ta sẽ trích xuất các giá trị số, sau đó loại bỏ các đơn vị đo vì chúng đã cùng đơn vị.

```
# Trích xuất giá trị số từ cột 'Engine_Number' và chuyển đổi thành kiểu số
dataFrame['Engine'] = dataFrame['Engine'].str.extract('(\d+)')
dataFrame['Engine'] = pd.to_numeric(dataFrame['Engine'], errors='coerce')
# dataFrame['Engine'] = dataFrame['Engine'].astype('int64')
# Co gia tri NULL nen khong chuyen ve int64


# Trích xuất giá trị số từ cột 'Power_Number' và chuyển đổi thành kiểu số
dataFrame['Power'] = dataFrame['Power'].str.extract('(\d+\.*\d*)')
dataFrame['Power'] = pd.to_numeric(dataFrame['Power'], errors='coerce')

# Kết quả
dataFrame.head()
```

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	2010	72000	CNG	Manual	First	26.6 km/kg	998.0	58.16	5.0	1.75
1	2015	41000	Diesel	Manual	First	19.67 kmpl	1582.0	126.20	5.0	12.50
2	2011	46000	Petrol	Manual	First	18.2 kmpl	1199.0	88.70	5.0	4.50
3	2012	87000	Diesel	Manual	First	20.77 kmpl	1248.0	88.76	7.0	6.00
4	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968.0	140.80	5.0	17.74

Hình 3. 7 Dữ liệu sau trích xuất các giá trị số

Đối với cột "**Mileage**", do có nhiều đơn vị đo khác nhau nên ta sẽ chuyển đổi về cùng một đơn vị (*kmpl*) để thống nhất.

```
# Chuyển đổi giá trị 'Mileage' từ km/kg sang kmpl
def convert_mileage(x):
    if pd.isna(x):
        return x  # Return NaN
    elif 'km/kg' in x:
        return float(x.split()[0]) * 2.352
    else:
        return float(x.split()[0])

dataFrame['Mileage'] = dataFrame['Mileage'].apply(convert_mileage)

# Kết quả
dataFrame.head()
```

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	2010	72000	CNG	Manual	First	62.5632	998.0	58.16	5.0	1.75
1	2015	41000	Diesel	Manual	First	19.6700	1582.0	126.20	5.0	12.50
2	2011	46000	Petrol	Manual	First	18.2000	1199.0	88.70	5.0	4.50
3	2012	87000	Diesel	Manual	First	20.7700	1248.0	88.76	7.0	6.00
4	2013	40670	Diesel	Automatic	Second	15.2000	1968.0	140.80	5.0	17.74

3.2.2.5 Kiểm tra lại dữ liệu khuyết sau khi trích xuất các giá trị số

Kiểm tra lại dữ liệu khuyết sau khi trích xuất các giá trị số.

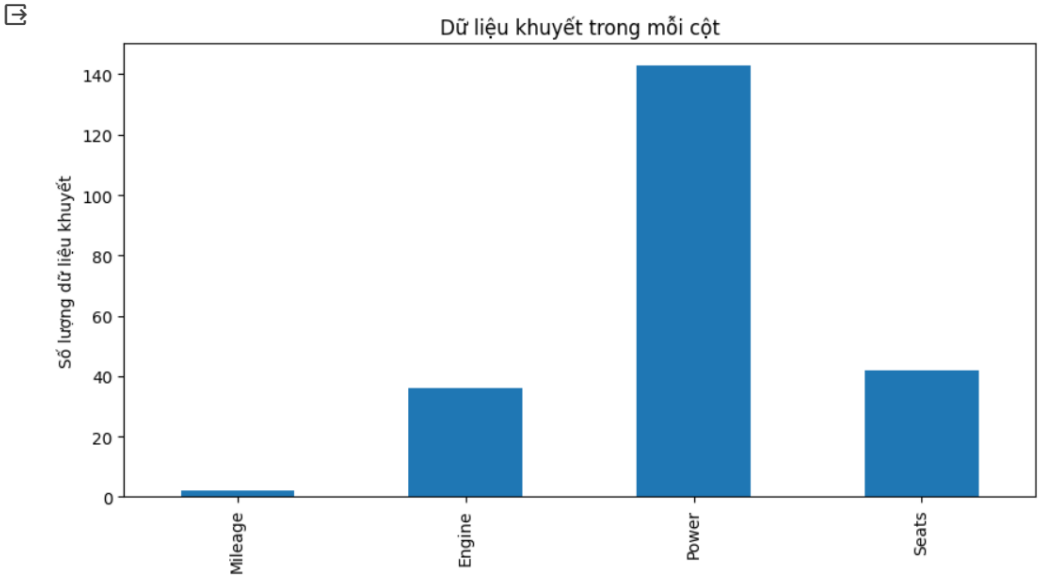
```
# Thống kê dữ liệu khuyết sau khi trích xuất giá trị số
statistical_missing_data()
```

```
Số liệu thiếu trong mỗi cột:
Year                 0
Kilometers_Driven    0
Fuel_Type            0
Transmission         0
Owner_Type           0
Mileage              2
Engine              36
Power              143
Seats               42
Price                0
dtype: int64

Số liệu trùng lặp:
2
```

Hình 3. 8 Dữ liệu khuyết sau khi trích xuất các giá trị số từ đơn vị

Như vậy, sau khi trích xuất các giá trị từ các cột đơn vị đo, ta thấy số lượng giá trị khuyết tăng lên. Điều này thể hiện, có những ô chỉ chứa đơn vị mà không chứa giá trị đo.

```
# Thống kê dữ liệu khuyết
plt_statistics_missing_data(dataFrame)
```



Hình 3. 9 Biểu đồ thanh thể hiện số lượng giá trị khuyết sau khi trích xuất giá trị số từ các cột chứa đơn vị đo

3.2.2.6 Điền khuyết dữ liệu

a) Lựa chọn giá trị điền khuyết

Vì các giá trị trong cột có độ lệch chuẩn lớn, nên ta sẽ sử dụng các giá trị có tần số xuất hiện nhiều nhất để điền khuyết.

Để làm điều này, trước tiên chúng ta cùng đếm các giá trị duy nhất trong mỗi cột.

```
# Đếm các giá trị duy nhất của các cột có Dtype phi số
# để xác định giá trị mode
categories_counts('Engine')
categories_counts('Power')
categories_counts('Mileage')
categories_counts('Seats')
```

Hình 3. 10 Thông tin các khiếu thiếu của các trường

```
----------------------------------------
Số các giá trị duy nhất của cột Engine:
1197.0    606
1248.0    512
1498.0    304
998.0     259
2179.0    240
          ...
2999.0      1
2147.0      1
2495.0      1
3200.0      1
1797.0      1
Name: Engine, Length: 146, dtype: int64
----------------------------------------
Số các giá trị duy nhất của cột Power:
74.00     235
98.60     131
73.90     125
140.00    123
88.50     112
          ...
80.90       1
68.10       1
301.73      1
174.57      1
181.04      1
Name: Power, Length: 369, dtype: int64
17.0000    172
18.6000    119
20.3600     88
21.1000     86
           ...
27.2800      1
14.5700      1
53.6256      1
8.0000       1
17.2400      1
Name: Mileage, Length: 442, dtype: int64
----------------------------------------
Số các giá trị duy nhất của cột Seats:
5.0     5014
7.0      674
8.0      134
4.0       99
6.0       31
2.0       16
10.0       5
9.0        3
0.0        1
Name: Seats, dtype: int64
```

b) *Điền khuyết bằng giá trị có tuần số xuất hiện lớn nhất.*

```
# Điền khuyết bằng giá trị Mode
def fillna_with_mode(col_name):
    mode_value = dataFrame[col_name].mode().iloc[0]
    dataFrame[col_name].fillna(mode_value, inplace=True)

fillna_with_mode('Engine')
fillna_with_mode('Power')
fillna_with_mode('Mileage')
fillna_with_mode('Seats')

# Kết quả
dataFrame.head()
```

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	2010	72000	CNG	Manual	First	62.5632	998.0	58.16	5.0	1.75
1	2015	41000	Diesel	Manual	First	19.6700	1582.0	126.20	5.0	12.50
2	2011	46000	Petrol	Manual	First	18.2000	1199.0	88.70	5.0	4.50
3	2012	87000	Diesel	Manual	First	20.7700	1248.0	88.76	7.0	6.00
4	2013	40670	Diesel	Automatic	Second	15.2000	1968.0	140.80	5.0	17.74

c) *Kết quả sau quá trình điền khuyết dữ liệu:*

```
# Kết quả sau khi điền khuyết
statistical_missing_data()
```

```
Số liệu thiếu trong mỗi cột:
Year                 0
Kilometers_Driven    0
Fuel_Type            0
Transmission         0
Owner_Type           0
Mileage              0
Engine               0
Power                0
Seats                0
Price                0
dtype: int64

Số liệu trùng lặp:
2
```

Hình 3. 11 Kết quả sau quá trình điền khuyết dữ liệu

3.2.2.7 Ánh xạ các giá trị kiểu category

a) Xác định từ điển ánh xạ

Bên cạnh đó, chúng ta còn có các cột "**Fuel_Type**", "**Transmission**", "**Owner_Type**" có số các giá trị duy nhất giới hạn, nên chúng ta có ánh xạ để chuyển dổi các dữ liệu kiểu nhãn này sang kiểu số, phục vụ cho mô hình dự đoán giá bán xe.

Để lập được từ điển ánh xạ, trước tiên chúng ta cùng kiểm tra xem các loại nhãn trong mỗi cột:

```
# Tìm các giá trị duy nhất để chuẩn bị từ điển mapping
categories_counts('Fuel_Type')
categories_counts('Transmission')
categories_counts('Owner_Type')
```

```
------------------------------------------------
Số các giá trị duy nhất của cột Fuel_Type:
Diesel      3205
Petrol      2746
CNG           56
LPG           10
Electric       2
Name: Fuel_Type, dtype: int64
------------------------------------------------
Số các giá trị duy nhất của cột Transmission:
Manual       4299
Automatic    1720
Name: Transmission, dtype: int64
------------------------------------------------
Số các giá trị duy nhất của cột Owner_Type:
First             4929
Second             968
Third              113
Fourth & Above       9
Name: Owner_Type, dtype: int64
```

b) Ánh xạ các giá trị phân loại thành kiểu số:

```
# Ánh xạ các giá trị phân loại thành nhãn kiểu số
def map_categorical(col_name, dictionary):
    dataFrame[col_name] = dataFrame[col_name].map(dictionary)

fuel_type_mapping = {'CNG': 1, 'Diesel': 2, 'Petrol': 3, 'LPG': 4, 'Electric': 5}
transmission_type_mapping = {'Manual': 1, 'Automatic': 2}
owner_type_mapping = {'First': 1, 'Second': 2, 'Third': 3, 'Fourth & Above': 4}

map_categorical('Fuel_Type', fuel_type_mapping)
map_categorical('Transmission', transmission_type_mapping)
map_categorical('Owner_Type', owner_type_mapping)

# Kết quả sau khi ánh xạ
dataFrame.head()
```

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	2010	72000	1	1	1	62.5632	998.0	58.16	5.0	1.75
1	2015	41000	2	1	1	19.6700	1582.0	126.20	5.0	12.50
2	2011	46000	3	1	1	18.2000	1199.0	88.70	5.0	4.50
3	2012	87000	2	1	1	20.7700	1248.0	88.76	7.0	6.00
4	2013	40670	2	2	2	15.2000	1968.0	140.80	5.0	17.74

Hình 3. 12 Dữ liệu sau khi ánh xạ các giá trị phân loại thành kiểu số

3.2.3. Phân tích mô tả

Phân tích mô tả trong phân tích dữ liệu là quá trình tóm tắt, mô tả và hiểu sâu về các đặc điểm, mẫu thái và thông tin quan trọng của tập dữ liệu. Trước khi đi vào bài toán phân tích hồi quy, bằng các biểu đồ chúng ta sẽ tiến hành tóm tắt, mô tả phân tích đơn biến (trên từng biến) và phân tích đa biến (trên nhiều biến) cho bộ dữ liệu nhằm hiểu sâu về các đặc điểm, mẫu thái và thông tin quan trọng của tập dữ liệu.

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Seats	Price	Engine_Number	Power_Number
0	2010	72000	1	1	1	0.793083	5.0	1.75	998	0.045569
1	2015	41000	2	1	1	0.249347	5.0	12.50	1582	0.174971
2	2011	46000	3	1	1	0.230712	5.0	4.50	1199	0.103652
3	2012	87000	2	1	1	0.263291	7.0	6.00	1248	0.103766
4	2013	40670	2	2	2	0.192683	5.0	17.74	1968	0.202739
5	2012	75000	4	1	1	0.629100	5.0	2.35	814	0.039939
6	2013	86999	2	1	1	0.292574	5.0	3.50	1461	0.054964
7	2016	36000	2	2	1	0.144005	8.0	17.50	2755	0.261126
8	2013	64430	2	1	1	0.260375	5.0	5.20	1598	0.131989
9	2012	65932	2	1	2	0.282686	5.0	1.95	1248	0.075694
10	2018	25692	3	1	1	0.273306	5.0	9.95	1462	0.131324
11	2012	60000	3	2	1	0.212965	5.0	4.49	1497	0.156143
12	2015	64424	2	1	1	0.319448	5.0	5.60	1248	0.075694
13	2014	72000	2	2	1	0.160992	5.0	27.00	2179	0.291936
14	2012	85000	2	2	2	0.000000	5.0	17.50	2179	0.153671

Hình 3. 13 Dữ liệu sau khi tiền xử lý

Tập dữ liệu sau quá trình tiền xử lý bao gồm hai loại chính: dữ liệu định lượng và dữ liệu định tính. Dữ liệu định lượng là các thông tin số học được sử dụng để đo lường và tính toán, chẳng hạn như giá xe (Price), số km đã đi (Kilometers_Driven), công suất động cơ (Power), và dung tích động cơ (Engine_Number). Các thông tin này cho phép thực hiện các phép toán số học cũng như áp dụng các phương pháp thống kê như tính trung bình, độ lệch chuẩn, và vẽ các biểu đồ phân bố. Ngược lại, dữ liệu định tính là các thông tin không mang tính số học, dùng để phân loại hoặc nhóm, ví dụ như loại nhiên liệu (Fuel_Type), hộp số (Transmission), và loại chủ sở hữu (Owner_Type). Dữ liệu này thường được trình bày thông qua tần suất hoặc tỷ lệ phần trăm để minh họa sự phân bổ giữa các nhóm. Để trực quan hóa, các biểu đồ như biểu đồ cột (bar chart) hoặc biểu đồ tròn (pie chart) thường được sử dụng.

Cách tiếp cận phân tích được áp dụng phù hợp với từng loại dữ liệu. Đối

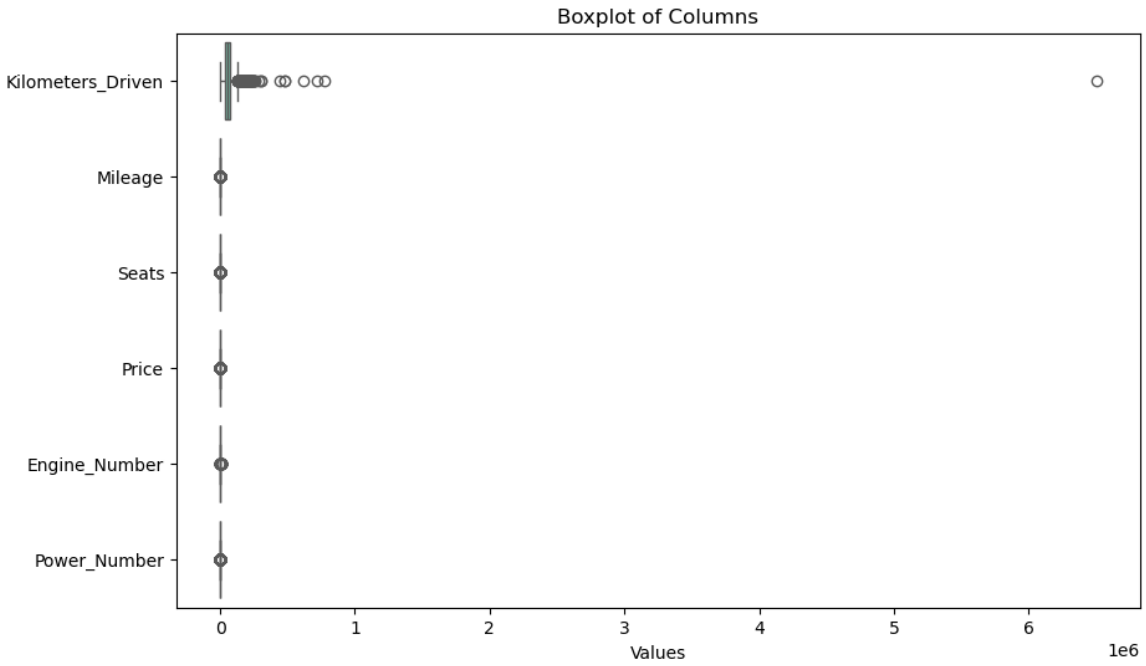
với dữ liệu định lượng, các phương pháp thống kê mô tả sẽ tập trung vào việc tóm tắt và trình bày xu hướng trung tâm cũng như mức độ biến thiên. Đối với dữ liệu định tính, việc phân tích chủ yếu xoay quanh việc tìm hiểu sự phân bổ và so sánh giữa các nhóm.

3.2.3.1. Dữ liệu định lượng trong tập dữ liệu

	Year	Kilometers_Driven	Mileage	Seats	Price	Engine_Number	Power_Number
count	5975.000000	5.975000e+03	5975.000000	5975.000000	5975.000000	5975.000000	5975.000000
mean	2013.386778	5.867431e+04	0.235090	5.278828	9.501647	1621.606695	0.150393
std	3.247238	9.155851e+04	0.079140	0.808959	11.205736	601.036987	0.101589
min	1998.000000	1.710000e+02	0.000000	0.000000	0.440000	624.000000	0.000000
25%	2012.000000	3.390800e+04	0.193444	5.000000	3.500000	1198.000000	0.081400
50%	2014.000000	5.300000e+04	0.230712	5.000000	5.650000	1493.000000	0.122480
75%	2016.000000	7.300000e+04	0.267474	5.000000	9.950000	1984.000000	0.197604
max	2019.000000	6.500000e+06	1.000000	10.000000	160.000000	5998.000000	1.000000

Bảng 3 Bảng thống kê cho các cột dữ liệu định lượng

- Quan sát cột count, tất cả các cột đều có 5.975 mẫu, chứng tỏ không có giá trị bị thiếu trong các cột định lượng này, điều này đảm bảo dữ liệu đầy đủ để phân tích.
- Cột Kilometers_Driven có giá trị trung bình khoảng **58.674 km**, nhưng giá trị tối đa (max) lên đến **6.500.000 km**, cho thấy có khả năng tồn tại giá trị ngoại lệ lớn.
- Các cột như Seats và Mileage có phạm vi giá trị hẹp hơn, phản ánh dữ liệu tương đối đồng nhất. Ngược lại, Kilometers_Driven, Price, và Engine_Number cho thấy sự chênh lệch lớn giữa các giá trị nhỏ nhất và lớn nhất

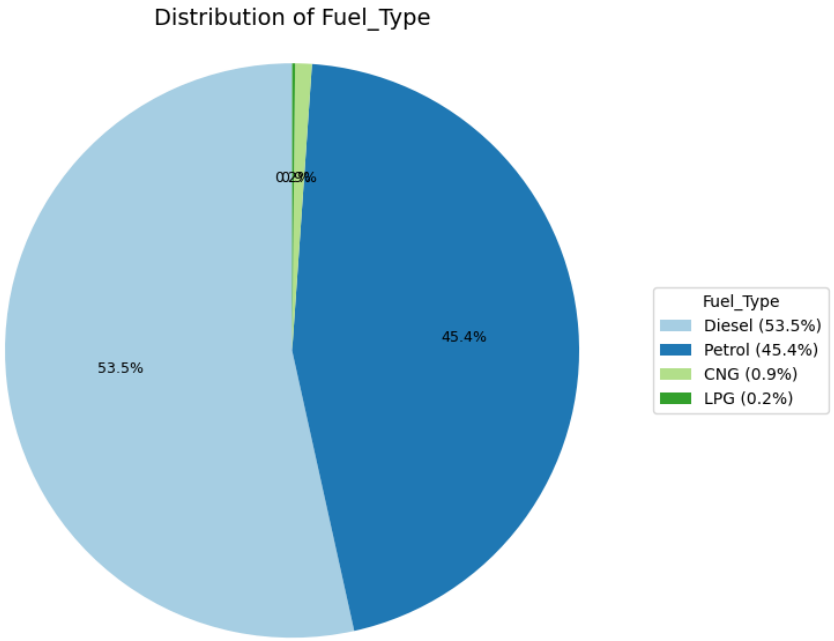


Hình 3. 14 Biểu đồ Boxplot của các cột định lượng

- Kilometers_Driven: Biểu đồ hộp cho thấy cột này chứa rất nhiều giá trị ngoại lệ nằm xa mức phân phối chính, ví dụ như các mẫu có giá trị vượt trên 3 triệu km.
- Các cột còn lại như Mileage, Seats, Price, Engine_Number, và Power_Number ít có ngoại lệ hơn hoặc ngoại lệ không đáng kể.

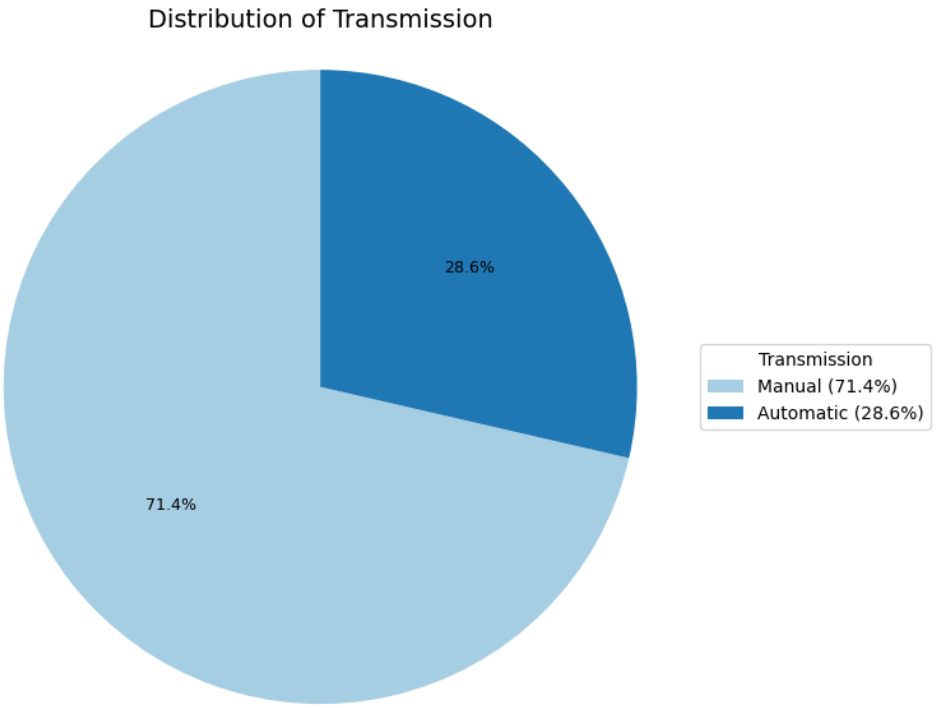
3.2.3.2. Dữ liệu định tính trong tập dữ liệu:

```
Thống kê cho cột Fuel_Type:
Fuel_Type
2    53.472803
3    45.422594
1     0.937238
4     0.167364
Name: proportion, dtype: float64

Thống kê cho cột Transmission:
Transmission
1    71.39749
2    28.60251
Name: proportion, dtype: float64

Thống kê cho cột Owner_Type:
Owner_Type
1    82.058577
2    15.949791
3     1.857741
4     0.133891
Name: proportion, dtype: float64
```



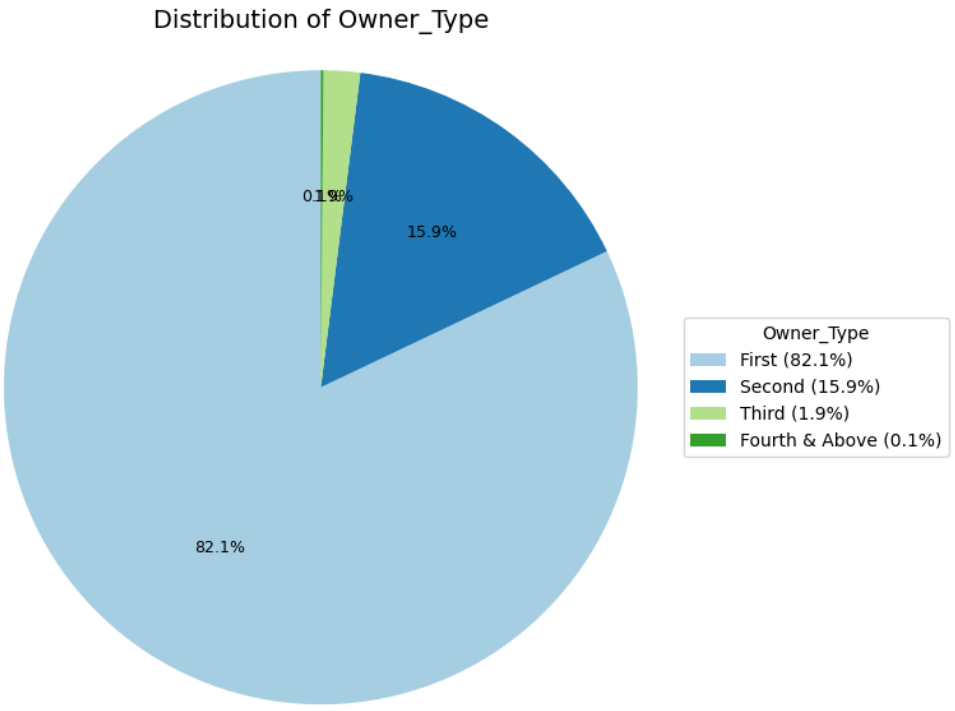
Hình 3. 15 Phân phối dữ liệu của Fuel_Type

- Với tỷ lệ cao của Diesel (53.47%) và Petrol (45.42%), dữ liệu này phản ánh rằng thị trường ô tô vẫn chủ yếu dựa vào nhiên liệu truyền thống. Điều này có thể do giá thành thấp hơn hoặc khả năng tương thích tốt hơn với hạ tầng hiện tại (trạm xăng dầu).
- Tỷ lệ thấp của CNG (0.93%) và Electric (0.17%) cho thấy sự chấp nhận của các nhiên liệu thay thế còn hạn chế.



Hình 3. 16 Phân phối dữ liệu của Transmission

Sự phổ biến của xe số tay (Manual - 71.40%) cho thấy thị trường này vẫn ưa chuộng xe với giá thành thấp hơn và chi phí bảo trì dễ chịu hơn. Tuy nhiên, nhu cầu về xe số tự động (Automatic - 28.60%) có thể gia tăng trong các phân khúc cao cấp hoặc với khách hàng ở khu vực đô thị, nơi xe tự động giúp tối ưu hóa việc lái xe trong điều kiện giao thông đông đúc.



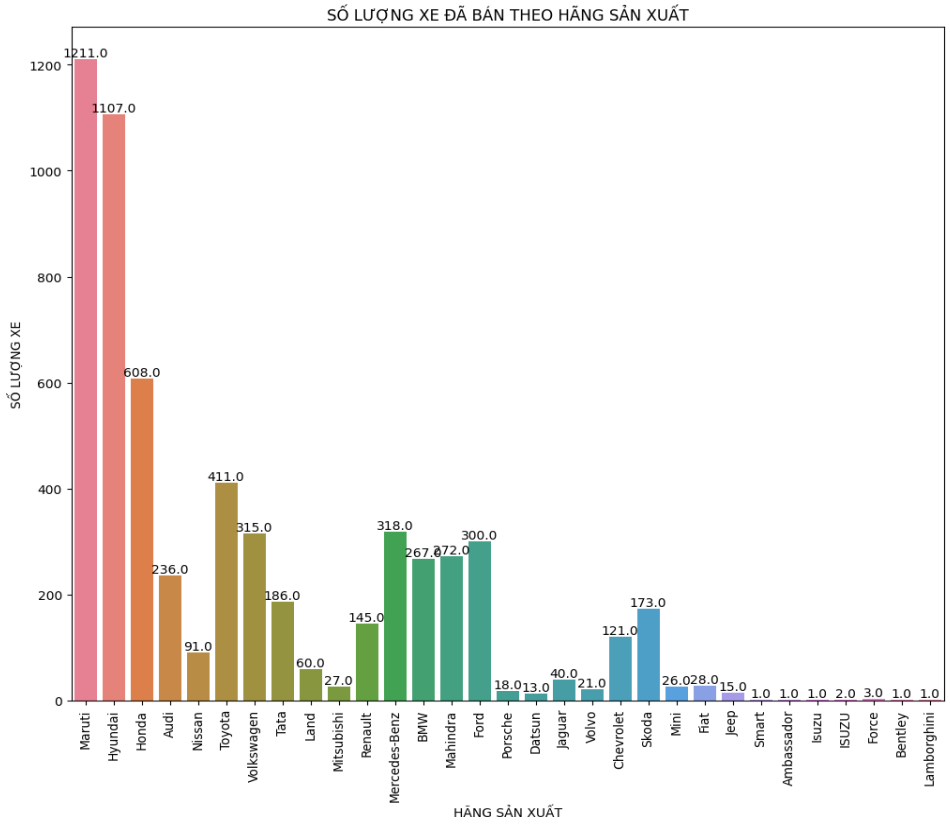
Hình 3. 17 Phân phối dữ liệu của Owner_Type

- Với tỷ lệ First Owner (82.06%) chiếm ưu thế, thị trường xe đã qua sử dụng tập trung chủ yếu vào các xe chỉ trải qua một đời chủ. Điều này có thể liên quan đến chất lượng xe tốt hơn hoặc chính sách bảo hành, hỗ trợ dịch vụ từ nhà sản xuất khi xe chỉ qua tay một chủ.
- Các loại xe từ Third Owner (1.86%) trở lên có tỷ lệ thấp, nhưng đây cũng là nhóm thường gặp các vấn đề như hao mòn hoặc khó khăn trong việc xác minh nguồn gốc

3.2.3.3. Biểu đồ phân bổ các hãng xe và số lượng bán tương ứng

- Dạng biểu đồ: Cột (Vertical bar chart)
- Loại phân tích: Đơn biến (‘Manufacturer’)
- Kiểu dữ liệu: Phi số (object)
- Cách vẽ:

- Tách cột ‘Name’ để tạo cột ‘Manufacture’, chứa tên các nhà sản xuất.
- Dùng ‘sns.countplot’ để vẽ biểu đồ cột thể hiện số lượng dữ liệu của mỗi nhà sản xuất.



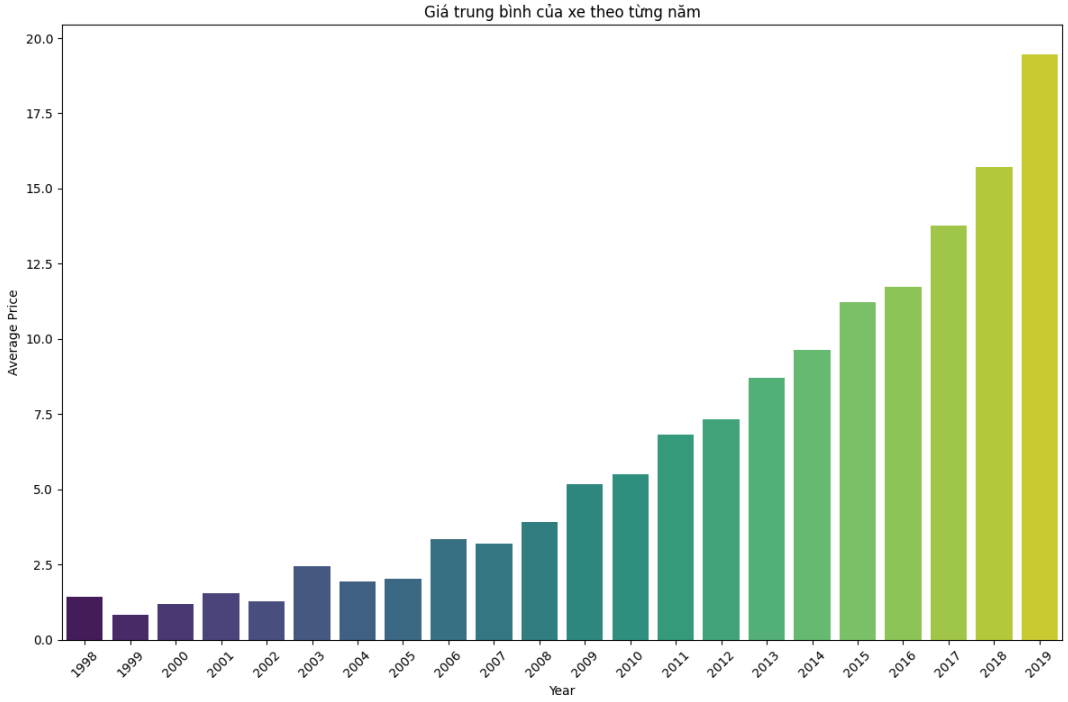
Hình 3. 18 Phân bố các hãng xe và số lượng bán tương ứng

Biểu đồ cột là một công cụ hữu ích để trình bày số lượng hoặc giá trị cụ thể của các mục khác nhau. Trong ví dụ này, chúng ta quan sát được sự chênh lệch rõ ràng về số lượng xe đã được bán theo từng hãng. Chẳng hạn, hãng "Maruti" có số lượng xe bán ra nhiều nhất, số lượng lên tới 1200 chiếc. Trong khi đó, các hãng khác như "Bentley" hoặc "Lamborghini" đều có số lượng bán ra khá gần nhau và rất thấp. Điều này cho thấy sự đa dạng về lựa chọn của người tiêu dùng Ấn Độ và cũng thể hiện mức độ phổ biến của mỗi hãng xe trên thị trường tại đất nước đó.

3.2.3.4. Biểu đồ về sự thay đổi giá xe theo từng năm

- Dạng biểu đồ: Cột (Vertical bar chart)
- Loại phân tích: Đa biến (‘Year’, ‘Price’)

- Kiểu dữ liệu: Số nguyên, số thực (int64, float)
- Cách vẽ:
  - Sử dụng câu lệnh "groupby ('Year')['Price'].mean()" để tính giá trị trung bình của cột 'Price' theo mỗi năm.
  - Sử dụng "sns.barplot" với đầu vào tập dữ liệu trên, trục x là 'Year', trục y là 'Price' để vẽ biểu đồ.



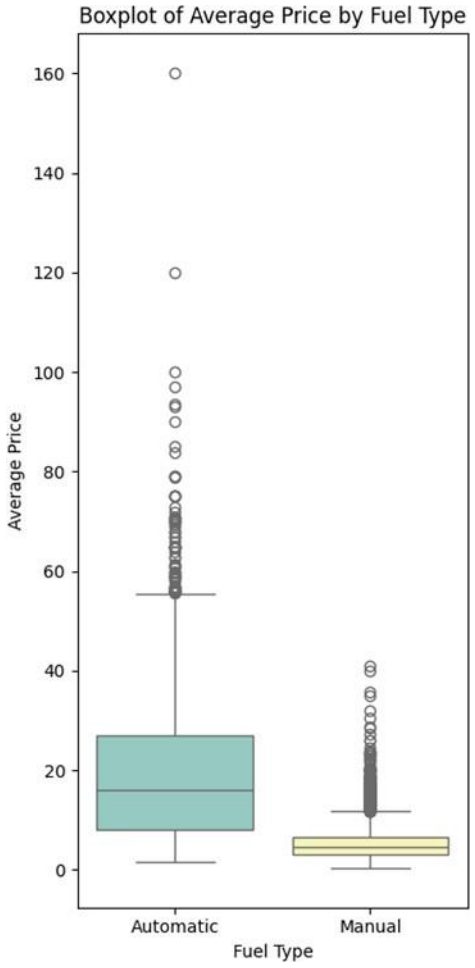
Hình 3. 19 Biểu đồ về sự thay đổi giá xe theo từng năm

Biểu đồ cột là một phương tiện mạnh mẽ để trình bày sự thay đổi về giá trị cụ thể của một đối tượng qua thời gian. Trong trường hợp này, chúng ta quan sát được xu hướng tăng giá của xe trung bình theo từng năm. Đặc biệt, từ năm gần đây nhất, giá xe đã có những bước tăng đáng kể. Ví dụ, từ năm 2009 và 2019, giá trung bình của mỗi mẫu xe đã tăng gần như gấp đôi, thể hiện giá trị ngày càng cao của các mẫu xe so với các năm trước. Điều này cũng phản ánh xu hướng chung của thị trường xe hơi, khi các công nghệ và tính năng mới được tích hợp, giá cả của chúng thường có xu hướng tăng.

3.2.3.5. Biểu đồ phân bố giá xe theo phương thức sản xuất

- Dạng biểu đồ: Hộp (Boxplot chart)

- Loại phân tích: Đa biến (‘Transmission, ‘Price’)
- Kiểu dữ liệu: Hỗn hợp (int64, float)
- Cách vẽ:
  - Sử dụng câu lệnh “groupby (‘Transmission’)[‘Price’].mean()” để tính giá trị trung bình của cột ‘Price’ theo mỗi phương thức sản xuất. Rồi sắp xếp theo thứ tự giảm dần của ‘Price’.
  - Sử dụng “sns.boxplot” để vẽ biểu đồ



Hình 3. 20 Biểu đồ phân bố giá xe theo phương thức sản xuất

Biểu đồ boxplot là một công cụ mạnh mẽ để mô tả phân phối dữ liệu và sự biến động của nó. Trong trường hợp này, chúng ta sử dụng biểu đồ này để so sánh giá xe trung bình giữa hai phương thức sản xuất: Automatic và Manual.

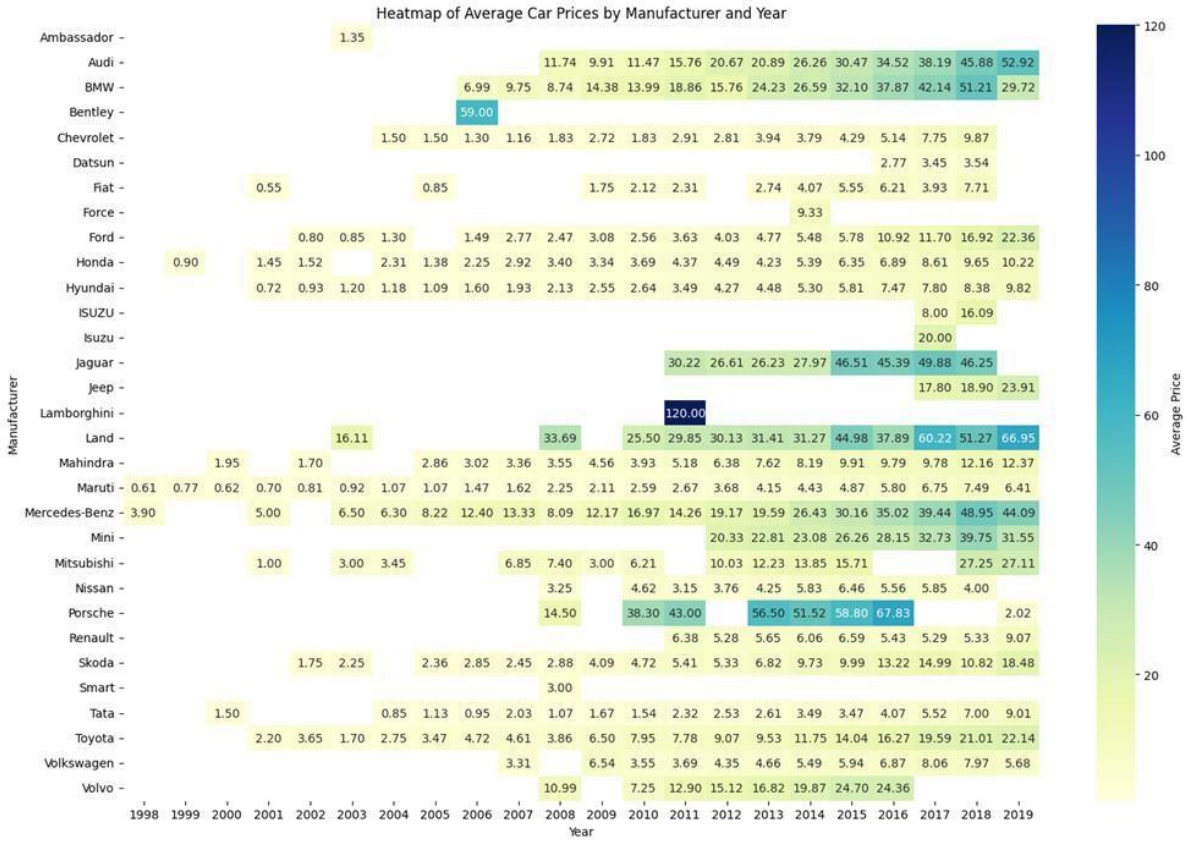
Khi nhìn vào biểu đồ, ta thấy rằng phương thức sản xuất Automatic thường có giá xe trung bình cao hơn so với phương thức Manual. Điều này có thể phản ánh sự khác biệt về công nghệ, tiện ích và chi phí sản xuất giữa hai phương thức này.

Tuy nhiên, phần trên và dưới của biểu đồ boxplot cho phương thức Automatic đều rộng, cho thấy sự phân tán dữ liệu lớn và có thể có sự chênh lệch về giá giữa các mẫu xe cụ thể. Điều này cũng đề xuất rằng, ngoài phương thức sản xuất, còn có những yếu tố khác như thương hiệu, tính năng, và chất lượng có thể ảnh hưởng đến giá của mỗi mẫu xe.

Trong khi đó, nếu phương thức Manual có khoảng cách trên và dưới hẹp, điều này ám chỉ rằng có sự đồng nhất hơn về giá cả của các mẫu xe dựa trên phương thức sản xuất Manual. Điều này có thể nghĩa là, dù có những sự khác biệt về chất lượng và tính năng giữa các mẫu xe Manual, thị trường hoặc các yếu tố khác không tạo ra sự chênh lệch lớn về giá cả giữa chúng.

3.2.3.6. Biểu đồ phân bổ trung bình giá xe theo từng năm đối với mỗi hãng xe.

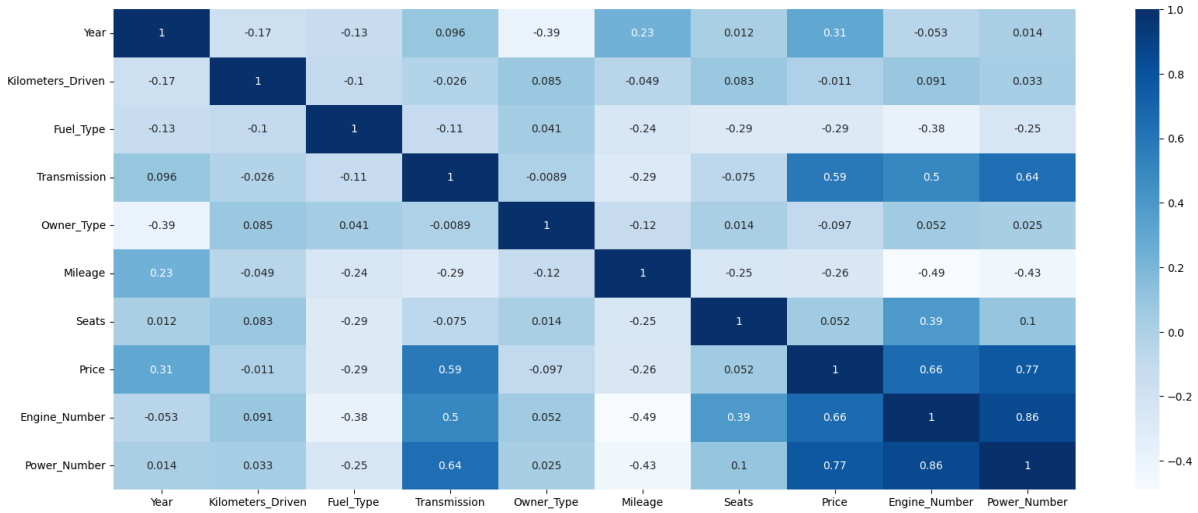
- Dạng biểu đồ: Heatmap (Heatmap chart)
- Loại phân tích: Đa biến (‘Manufacturer’, ‘Price’, ‘Year’)
- Kiểu dữ liệu: Hỗn hợp (object, float, int64)
- Cách vẽ:
  - Sử dụng câu lệnh “groupby([‘Year’, ‘Manufacture’])[‘Price’].mean()” để tính giá trị trung bình của cột ‘Price’ theo mỗi năm với mỗi nhà sản xuất.
  - Chuyển đổi (pivot) dữ liệu sang dạng khác cụ thể là các hàng là ‘Manufacturer’, các cột là ‘Year’, các giá trị là giá trị trung bình của ‘Price’ theo mỗi nhóm. Việc này giúp ta dễ dàng xây dựng biểu đồ heatmap.
  - Sử dụng “sns.heatmap” để vẽ biểu đồ.



Hình 3. 21 Biểu đồ phân bố trung bình giá xe theo từng năm đối với mỗi thuộc tính

Biểu đồ heatmap sử dụng màu sắc để biểu thị mối quan hệ giữa hai chiều dữ liệu và hiển thị sự tương tác giữa các yếu tố trong một ma trận. Cụ thể với biểu đồ trên, hai chiều dữ liệu ở đây là ‘Year và ‘Manufacturer’, giá trị màu sắc là trung bình giá xe (mean của ‘Price’). Qua quan sát ta thấy một vài hãng xe lâu đời và vẫn phát triển ở Ấn Độ cho tới 2019 như Maruti, Mercedes-Benz, Honda, Toyota. Ngoài ra có một vài dòng xe mới phát triển gần đây như Datsun, Mini.

3.2.3.7. Biểu đồ tương quan giữa các thuộc tính



Hình 3. 22 Biểu đồ tương quan giữa các thuộc tính

Khi nhìn vào biểu đồ heatmap, các ô màu sẫm sẽ biểu thị mối tương quan mạnh và dương giữa các biến số, trong khi các ô màu sáng sẽ cho thấy mối tương quan mạnh và âm. Các con số được chú thích trong từng ô sẽ là giá trị tương quan chính xác, giúp chúng ta có cái nhìn cụ thể và định lượng hơn về mức độ tương quan giữa các biến.

Thông qua biểu đồ heatmap này, chúng ta có thể dễ dàng nhận diện và hiểu rõ hơn về các mối quan hệ tương quan giữa các biến số số học trong dữ liệu với các biến cần phân tích, đồng thời cũng có thể đưa ra những kết luận quan trọng về dữ liệu.

3.2.4.Phân tích hồi quy

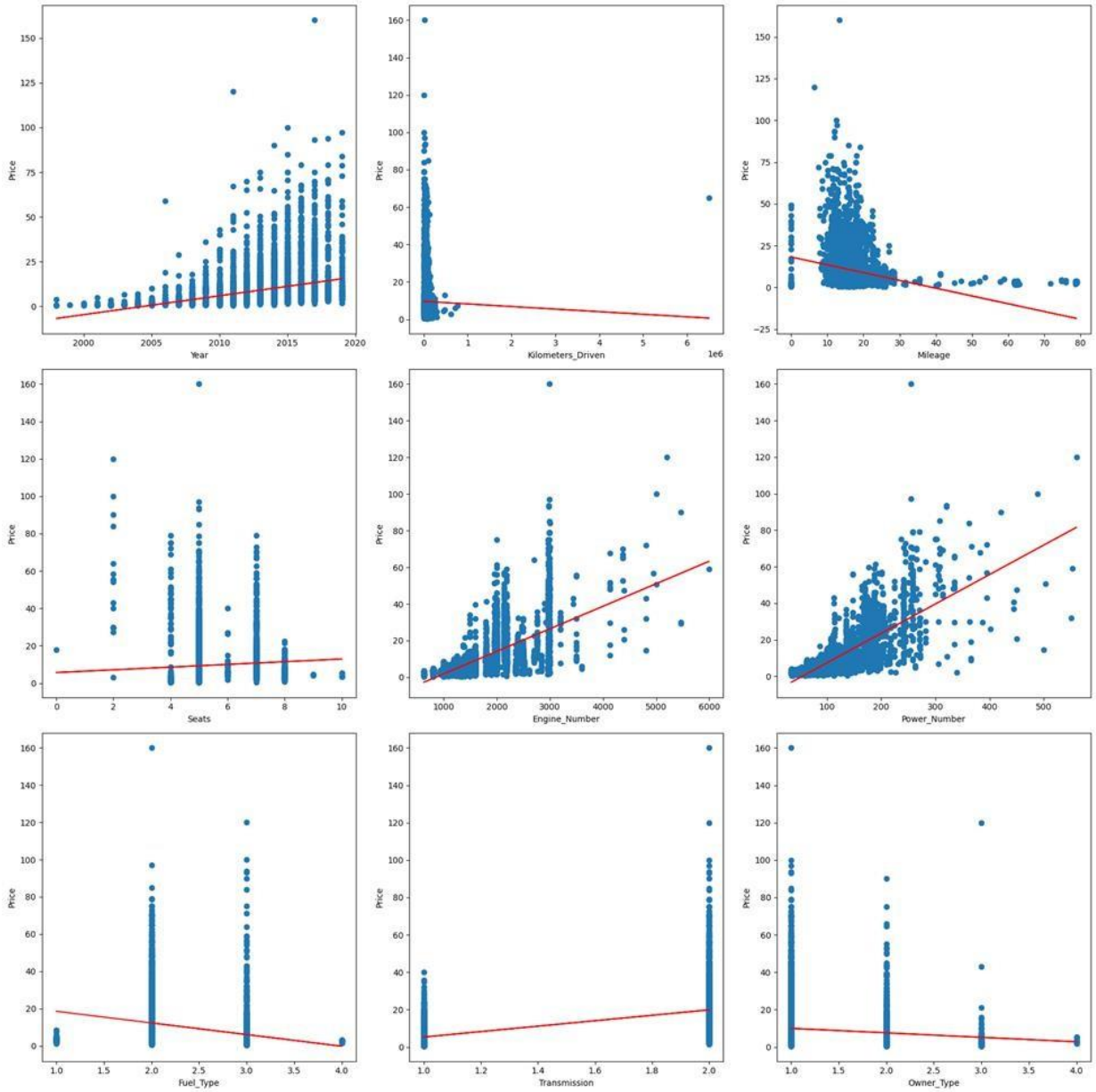
Với dự án hiện tại, mục tiêu được đặt ra là cần dự báo giá xe dựa theo mô hình hồi quy tuyến tính. Từ đó, ta đặt ra biến mục tiêu để dự báo (Target Value) chính là ‘Price’ của bộ dữ liệu.

Để huấn luyện mô hình hồi quy tuyến tính, ta cần dữ liệu đầu vào hoàn toàn là dữ liệu số. Để có được dữ liệu như vậy ta cần bỏ đi một vài cột phi số không cần thiết hoặc chuyển đổi dữ liệu phi số sang dữ liệu kiểu số mà trong quá trình chuẩn hóa dữ liệu đã đề cập. Ta có tập dữ liệu đầu vào như sau:

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Seats	Price	Engine_Number	Power_Number
0	2010	72000	1	1	1	0.793083	5.0	1.75	998	0.045569
1	2015	41000	2	1	1	0.249347	5.0	12.50	1582	0.174971
2	2011	46000	3	1	1	0.230712	5.0	4.50	1199	0.103652
3	2012	87000	2	1	1	0.263291	7.0	6.00	1248	0.103766
4	2013	40670	2	2	2	0.192683	5.0	17.74	1968	0.202739
5	2012	75000	4	1	1	0.629100	5.0	2.35	814	0.039939
6	2013	86999	2	1	1	0.292574	5.0	3.50	1461	0.054964
7	2016	36000	2	2	1	0.144005	8.0	17.50	2755	0.261126
8	2013	64430	2	1	1	0.260375	5.0	5.20	1598	0.131989
9	2012	65932	2	1	2	0.282686	5.0	1.95	1248	0.075694
10	2018	25692	3	1	1	0.273306	5.0	9.95	1462	0.131324
11	2012	60000	3	2	1	0.212965	5.0	4.49	1497	0.156143
12	2015	64424	2	1	1	0.319448	5.0	5.60	1248	0.075694
13	2014	72000	2	2	1	0.160992	5.0	27.00	2179	0.291936
14	2012	85000	2	2	2	0.000000	5.0	17.50	2179	0.153671

Sau khi đã xử lý cắt bỏ dữ liệu không cần thiết và chuyển đổi dữ liệu phù hợp, ta tiến hành huấn luyện cho mô hình hồi quy tuyến tính và đánh giá kết quả của thử nghiệm.

3.2.4.1. Hồi quy đơn biến



Hình 3. 23 Biểu đồ scatter tổng hợp các mối quan hệ giữa 'Price' và các biến

Tuy nhiên, từ các biểu đồ scatter, chúng ta có thể thấy rằng mối quan hệ giữa 'Price' và các biến độc lập không phải lúc nào cũng là tuyến tính. Có thể có sự biến đổi không đều, không đồng nhất hoặc phi tuyến trong mối quan hệ này. Điều này có thể xuất phát từ nhiều nguyên nhân, ví dụ như sự phụ thuộc tuyến tính của Price vào một số biến và phụ thuộc phi tuyến của nó vào các

biến khác, sự ảnh hưởng của các biến tương tác, hoặc sự hiện diện của các biến độc lập quan trọng chưa được xem xét.

Vì vậy, để mô tả và dự đoán Price một cách chính xác hơn, mô hình hồi quy đa biến có thể được sử dụng. Mô hình này cho phép chúng ta xem xét sự ảnh hưởng đồng thời của nhiều biến độc lập đến biến phụ thuộc, cung cấp một cái nhìn toàn diện hơn về mối quan hệ giữa chúng. Đồng thời, mô hình đa biến cũng giúp chúng ta phát hiện và đánh giá tốt hơn về sự tương tác giữa các biến, giúp mô phỏng mối quan hệ phức tạp hơn trong dữ liệu.

3.2.4.2. Hồi quy đa biến

Huấn luyện và lưu mô hình:

```
X = dataFrame.drop('Price', axis=1)
Y = dataFrame['Price']
X = np.array(X)

model = LinearRegression()
model.fit(X, Y)

from joblib import dump, load
dump(model, 'linear_regression_model.joblib')
model = load('linear_regression_model.joblib')
```

Đánh giá mô hình: Ta sử dụng bảng thông số OLS (Ordinary Least Squares)

OLS Regression Results

Dep. Variable:	Price	R-squared:	0.695
Model:	OLS	Adj. R-squared:	0.695
Method:	Least Squares	F-statistic:	1514.
Date:	Tue, 26 Dec 2023	Prob (F-statistic):	0.00
Time:	23:43:33	Log-Likelihood:	-19364.
No. Observations:	5975	AIC:	3.875e+04
Df Residuals:	5965	BIC:	3.881e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-2027.9616	57.311	-35.385	0.000	-2140.311	-1915.612
Year	1.0076	0.028	35.374	0.000	0.952	1.063
Kilometers_Driven	1.163e-06	8.97e-07	1.296	0.195	-5.96e-07	2.92e-06
Mileage	-0.0715	0.018	-3.959	0.000	-0.107	-0.036
Seats	-1.0361	0.130	-7.985	0.000	-1.290	-0.782
Engine_Number	0.0018	0.000	5.005	0.000	0.001	0.002
Power_Number	0.1213	0.004	32.653	0.000	0.114	0.129
Fuel_Type	-1.9264	0.204	-9.447	0.000	-2.326	-1.527
Transmission	2.7801	0.238	11.690	0.000	2.314	3.246
Owner_Type	-0.0356	0.192	-0.185	0.853	-0.413	0.341

Một trong những thông số quan trọng nhất mà chúng ta cần quan tâm tới là R-squared. Đây là một chỉ số được sử dụng trong việc đánh giá hiệu suất của mô hình hồi quy, thường được sử dụng để đo lường mức độ phù hợp của mô hình hồi quy với dữ liệu thực tế. Ở đây R Square 0.695 nói rằng mô hình hồi quy tuyến tính này giải thích được ~70% sự biến thiên liên quan đến các biến độc lập, còn lại 30% là các yếu tố ngẫu nhiên khác. Để một mô hình hồi quy gọi là phù hợp thì R Square phải càng gần 100% càng tốt. Như vậy có thể đánh giá là mô hình hồi quy tuyến tính này có khả năng dự đoán tương đối cao về giá xe ở thị trường Ấn Độ.

3.3. Đánh giá và đề xuất

Phần phân tích mô tả đã phân tích được bộ dữ liệu ra các biểu đồ phù hợp và cho ta được cái nhìn tổng quan xoay quanh giá xe ô tô tại thị trường Ấn Độ. Tuy nhiên, đối với bài toán dự báo, mô hình phân tích hồi quy tuyến tính của chủ đề nhóm em đang làm với thông số R Square tương đối ổn khoảng 70%. Điều này cho ta thấy mô hình có thể nắm bắt tốt mối quan hệ giữa các đặc điểm và biến mục tiêu (‘Price’). Để nâng cao hiệu suất của mô hình và cải thiện khả năng dự đoán, chúng em đề xuất sử dụng mô hình RandomForestRegressor. Mô hình RandomForest là một phương pháp học máy mạnh mẽ và linh hoạt, được xây dựng dựa trên việc kết hợp nhiều cây quyết định (decision trees).

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

X = dataFrame[['Year','Kilometers_Driven','Mileage','Seats','Engine_Number','Power_Number','Fuel_Type','Transmission','Owner_Type']]
Y = dataFrame['Price']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
n_estimators_value = 100
model = RandomForestRegressor(n_estimators=n_estimators_value)
model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)
mae = mean_absolute_error(Y_test, Y_pred)
mse = mean_squared_error(Y_test, Y_pred)
r2 = r2_score(Y_test, Y_pred)

print('Mean Absolute Error:', mae)
print('Mean Squared Error:', mse)
print('R Square (Test) :', r2)
print('R Square (Train) : ', model.score(X_train, Y_train))
```

Mô hình RandomForest đã được đào tạo và đạt được một R-squared (R^2) là 0.98, một sự cải thiện đáng kể so với mô hình hồi quy tuyến tính hiện tại.

Lợi ích khi chuyển sang RandomForest:

Hiệu suất: Với R^2 là 0.98, mô hình RandomForest có khả năng giải thích đến 98% sự biến động của biến mục tiêu, một mức độ khá cao và có thể đem lại kết quả dự đoán chính xác hơn.

Xử lý Dữ liệu: RandomForest có khả năng xử lý dữ liệu không rõ ràng và thiếu giá trị một cách tốt hơn so với hồi quy tuyến tính.

Khả năng Tổng quát hóa: Mô hình này có xu hướng ít overfitting hơn đối với dữ liệu huấn luyện.

```
Mean Absolute Error: 1.6187350541940626
Mean Squared Error: 24.204992257558605
R Square (Test) : 0.8337303221865516
R Square (Train) :  0.9863974650764467
```

3.4. Kết luận

Chương 3 đã trình bày phần thực nghiệm và đánh giá của dự án thông qua đầy đủ các bước từ tiền xử lý dữ liệu cho tới phân tích mô tả & bài toán dự báo. Từ đó đưa ra được các đánh giá và đề xuất phù hợp để cải thiện kết quả của dự án trong tương lai.

CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI

4.1 Cài đặt công cụ

4.1.1 Phần mềm Visual Studio Code chạy Python

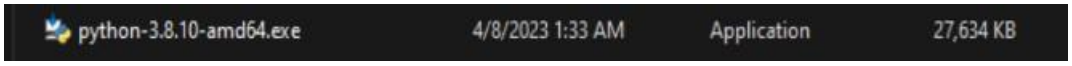
Bước 1: Download python version 3.8.10 environment theo link:

https://www.python.org/ftp/python/3.8.10/python-3.8.10-amd64.exe

(các hệ điều hành khác tương tự)

Bước 2: Cài đặt python trên máy local

+ Click đúp vào file .exe vừa tải về để cài đặt



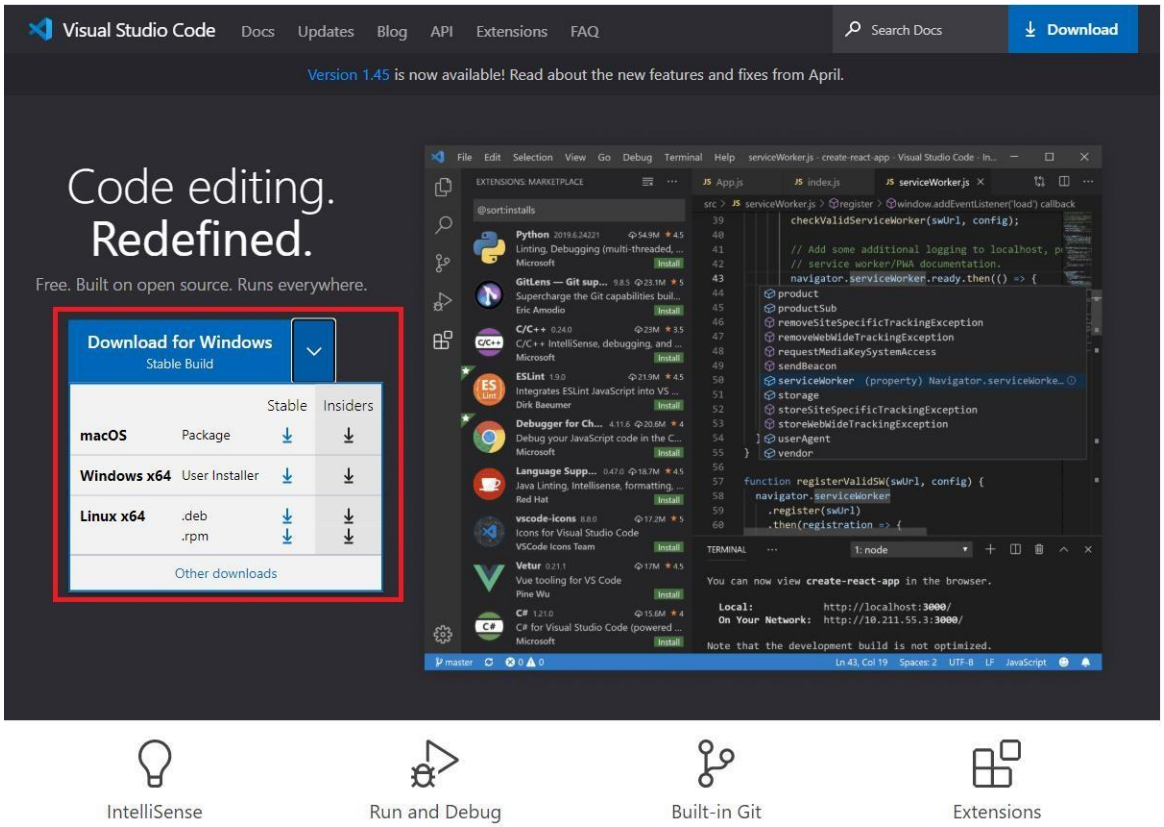
Chọn tick vào ô Add Python 3.8.10 to PATH (lựa chọn này giúp chạy lệnh python trên cmd, powershell trên windows)

Và click vào Install Now (Có thể đổi đường dẫn folder cài đặt bằng cách chọn phần Customize installation ở dưới)

Hình 4. 1 Cài đặt Visual Studio

Bước 3: Dowload Visual Studio Code

Vào trang chủ vscode link này, chọn phiên bản phù hợp với thiết bị của các bạn và tải về.

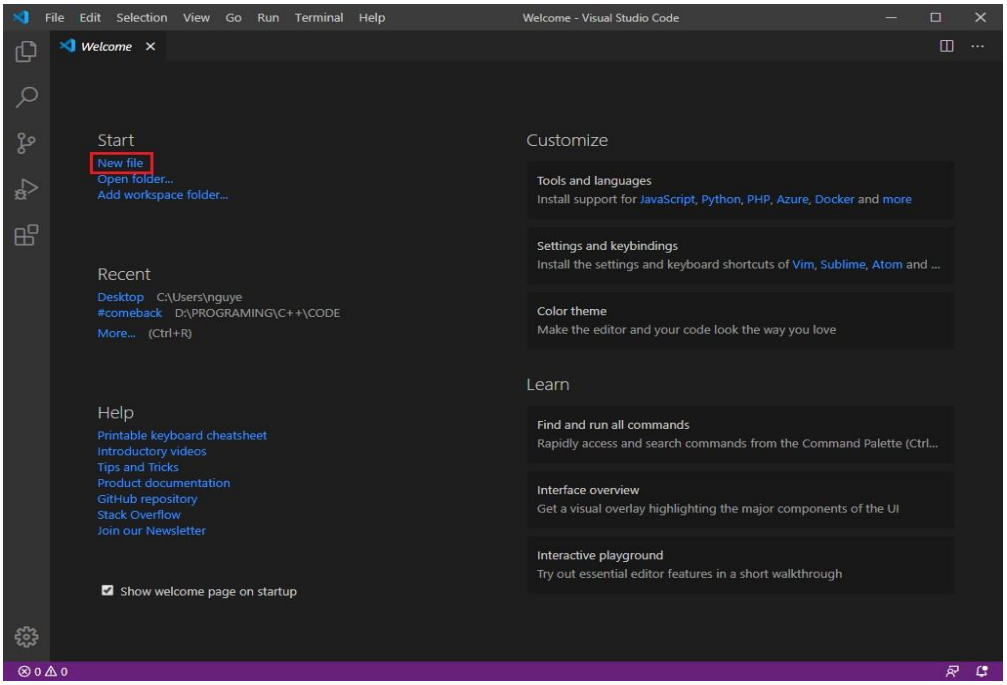


Hình 4. 2 Cài extension Python vào Visual

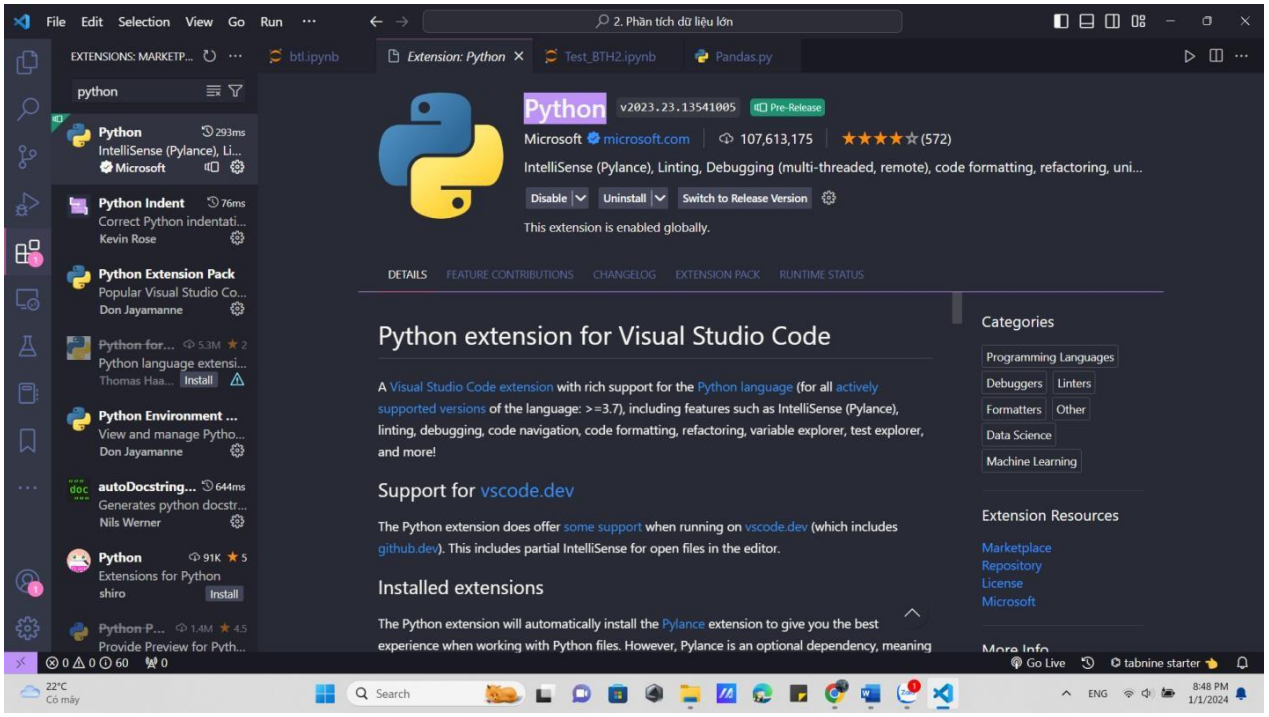
Sau khi tài về, tiến hành chạy file cài đặt. Việc cài đặt rất đơn giản, chỉ cần Next – Next – Next là xong.

+ Cài extension Python

Sau khi cài đặt, vscode sẽ có giao diện như thế này:



Các bạn chọn **New file** để tạo một text file đầu tiên.



Chúng ta ấn vào **Extensions** hoặc **Ctrl + Shift + X**, để mở giao diện như hình trên.

Tiếp theo các bạn gõ trên thanh tìm kiếm từ khóa "**Python**", sau đó chọn extension **Python** do **Microsoft** phát hành và ấn **Install** để cài đặt.

4.1.2. PyQt5

a) Giới thiệu

PyQt5 - một thư viện Python mạnh mẽ để tạo giao diện người dùng tương tác cho các mô hình máy học và ứng dụng AI.

b) Cài đặt

Để cài đặt PyQt5, chúng ta có thể sử dụng pip, một công cụ quản lý package Python phổ biến. Mở terminal hoặc command prompt và chạy lệnh sau: “pip install PyQt5”

c) Sử dụng

Import các thư viện cần thiết

Đầu tiên, chúng ta cần import các thư viện cần thiết trong Python:

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QPushButton, QWidget, QLabel, QLineEdit, QSpacerItem, QSiz
from sklearn.linear_model import LinearRegression
import numpy as np
```

Trong đó, chúng ta “import gradio” để sử dụng Gradio “LinearRegression” từ “sklearn.linear_model” để sử dụng mô hình hồi quy tuyến tính, “numpy” để làm việc với mảng đa chiều.

Định nghĩa hàm xử lý

Tiếp theo, chúng ta cần định nghĩa một hàm xử lý để thực hiện các tính toán, dự đoán hoặc xử lý dữ liệu. Đối với ví dụ này, chúng ta sẽ sử dụng một hàm đơn giản như sau:

```
def my_function(Year_input, Kilometers_Driven_input, Mileage_input, Seats_input,
                Engine_Number_input, Power_Number_input, Fuel_Type_input, Transmission_input, Owner_Type_input):
    x = np.array([[Year_input, Kilometers_Driven_input, Mileage_input, Seats_input,
                   Engine_Number_input, Power_Number_input, Fuel_Type_input, Transmission_input, Owner_Type_input]])
    result = model.predict(x)
    return  result, mean_squared_error(Y_test, Y_pred), r2_score(Y_test, Y_pred), model.score(X, Y), model.coef_, model.intercept_
```

Trong hàm này, chúng ta tạo một mảng x chứa các thông số đầu vào được nhập vào từ giao diện. Sau đó, chúng ta sử dụng mô hình đã được huấn luyện

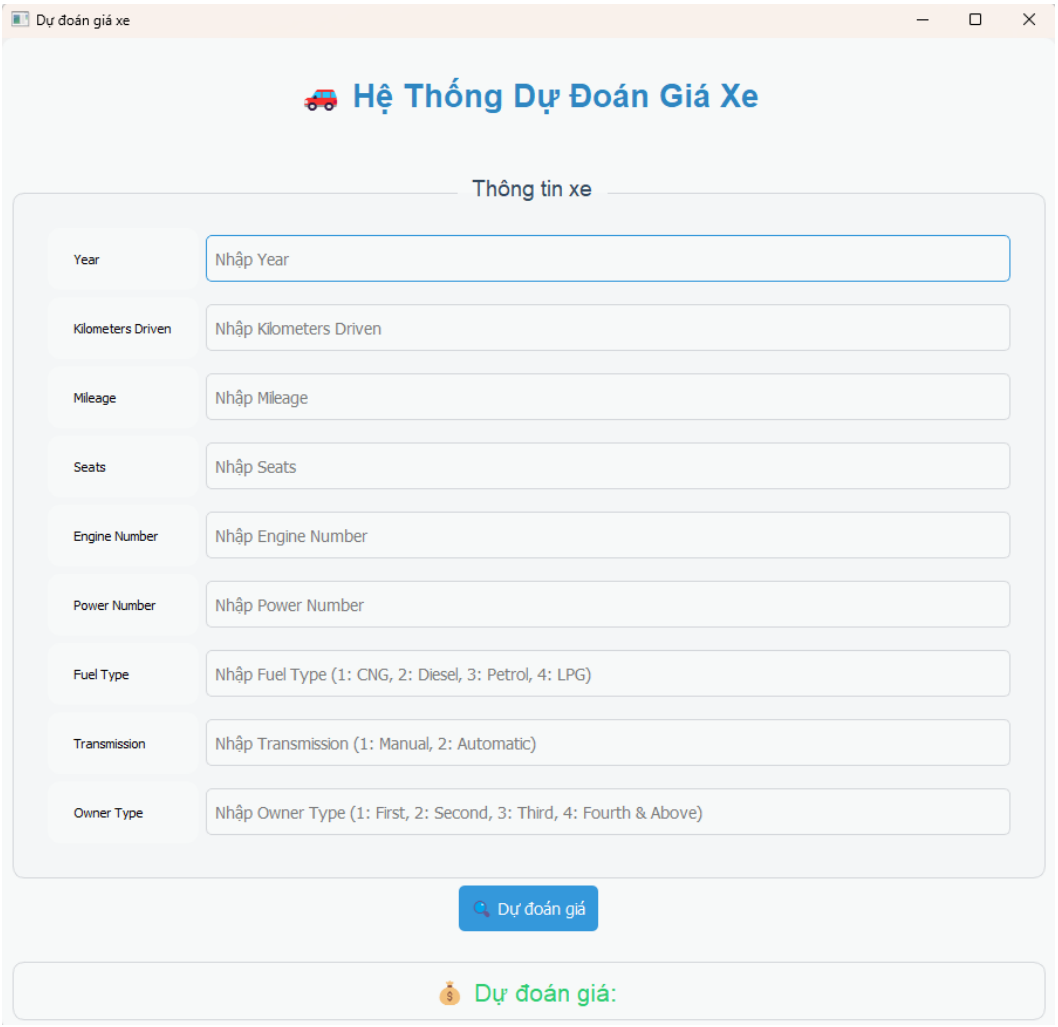
(`model`) để dự đoán giá xe hơi dựa trên `x`. Kết quả dự đoán được trả về cùng với các thông số khác như mean squared error, r-squared, slope, và intercept.

Tạo giao diện PyQt5

Bước tiếp theo là tạo giao diện PyQt5 bằng cách sử dụng các phần tử như self.setWindowTitle("Dự đoán giá xe"), self.setGeometry(100, 100, 1600, 900), self.layout = QVBoxLayout(). Dưới đây là một ví dụ đơn giản với một hàm đơn giản:

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QPushButton, QWidget,
from sklearn.linear_model import LinearRegression
import numpy as np

class PredictionApp(QMainWindow):
    Tabnine | Edit | Test | Explain | Document | Ask
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Dự đoán giá xe")
        self.setGeometry(100, 100, 1600, 900)  # Tăng kích thước cửa sổ lên 600x400

        self.layout = QVBoxLayout()

        # Các ô nhập liệu
        self.Year_input = QLineEdit(self)
        self.Year_input.setPlaceholderText("Nhập Year")
        self.Year_input.setFixedHeight(70)  # Điều chỉnh chiều cao của trường nhập liệu
        self.layout.addWidget(self.Year_input)

        self.Kilometers_Driven_input = QLineEdit(self)
        self.Kilometers_Driven_input.setPlaceholderText("Nhập Kilometers Driven")
        self.Year_input.setFixedHeight(70)  # Điều chỉnh chiều cao của trường nhập liệu
        self.layout.addWidget(self.Kilometers_Driven_input)

        self.Mileage_input = QLineEdit(self)
        self.Mileage_input.setPlaceholderText("Nhập Mileage")
        self.Year_input.setFixedHeight(70)
        self.layout.addWidget(self.Mileage_input)

        self.Seats_input = QLineEdit(self)
        self.Seats_input.setPlaceholderText("Nhập Seats")
        self.Year_input.setFixedHeight(70)
        self.layout.addWidget(self.Seats_input)

        self.Engine_Number_input = QLineEdit(self)
        self.Engine_Number_input.setPlaceholderText("Nhập Engine Number")
```

4.2 Giao diện



Hình 4. 3 Giao diện ứng dụng

Giao diện có tổng cộng 9 trường đầu vào.

1. Year: Trường nhập số để nhập năm sản xuất của xe.
2. Kilometers_Driven: Trường nhập số để nhập số kilomet đã đi của xe.
3. Mileage: Trường nhập số để nhập mức tiêu thụ nhiên liệu của xe.
4. Seats: Trường nhập số để nhập số ghế ngồi trong xe.
5. Engine_Number: Trường nhập số để nhập dung tích động cơ của xe.
6. Power_Number: Trường nhập số để nhập công suất động cơ của xe.
7. Fuel_Type: Trường nhập số để chọn loại nhiên liệu sử dụng (1: CNG, 2: Diesel, 3: Petrol, 4: LPG).

8. Transmission: Trường nhập số để chọn loại hộp số (1: Manual, 2: Automatic).
9. Owner_Type: Trường nhập số để chọn loại chủ sở hữu (1: First, 2: Second, 3: Third, 4: Fourth & Above).

Đầu ra sẽ bao gồm:

1. Dự đoán giá cho các thông số đầu vào là: Trường hiển thị kết quả dự đoán giá của xe dựa trên các thông số đầu vào.
2. Mean Squared Error: Trường hiển thị giá trị Mean Squared Error (MSE) của mô hình dự đoán.
3. R-squared: Trường hiển thị giá trị R-squared của mô hình dự đoán.
4. R Square: Trường hiển thị giá trị R Square của mô hình dự đoán.
5. Slope: Trường hiển thị giá trị Slope của mô hình dự đoán.
6. Intercept: Trường hiển thị giá trị intercept của mô hình dự đoán

Ứng dụng này cho phép người dùng nhập các thông số liên quan đến một chiếc xe và sau đó dự đoán giá trị của chiếc xe dựa trên các thông số đó. Các giá trị dự đoán cũng như các chỉ số đánh giá mô hình được hiển thị trong giao diện.

4.3 Kết luận

Trên đây là cách cài đặt và sử dụng PyQt5 để tạo giao diện cho các ứng dụng AI và mô hình máy học. PyQt5 là một thư viện mạnh mẽ và dễ sử dụng, giúp chúng ta tạo giao diện tương tác với mô hình một cách nhanh chóng và thuận tiện. Bằng cách sử dụng PyQt5, chúng ta có thể tạo ra các ứng dụng AI dễ dàng và chia sẻ chúng với người dùng một cách trực quan và hấp dẫn.

KẾT LUẬN

Trong nghiên cứu này, nhóm chúng tôi đã giải quyết bài toán dự đoán giá xe thông qua việc phân tích các dữ liệu liên quan đến các yếu tố ảnh hưởng đến giá trị của xe. Mục tiêu là sử dụng phương pháp hồi quy để xác định các yếu tố quan trọng và xây dựng mô hình dự đoán giá xe dựa trên các đặc tính như năm sản xuất, số km đã đi, kiểu dáng, màu sắc và nhiều yếu tố khác. Quá trình này bao gồm việc thu thập dữ liệu, tiền xử lý, phân tích dữ liệu và áp dụng các thuật toán hồi quy để đưa ra dự đoán.

Kết quả nhóm chúng tôi đạt được là đã xây dựng được mô hình hồi quy tuyến tính và hồi quy đa biến cho phép dự đoán giá xe khá chính xác. Dữ liệu sau khi được làm sạch và xử lý, mô hình đã cho ra các giá trị dự đoán có độ chính xác tương đối cao, đồng thời xác định được các yếu tố có ảnh hưởng mạnh nhất đến giá xe. Các mô hình này có thể ứng dụng trong việc xác định giá trị xe cũ một cách tự động, giúp người tiêu dùng và các đại lý xe dễ dàng tham khảo.

Tuy nhiên, một số vấn đề nhóm chúng tôi vẫn chưa giải quyết được là việc áp dụng phương pháp hồi quy đa biến với các dữ liệu phức tạp hơn, chẳng hạn như khi có quá nhiều yếu tố tương tác giữa các biến, hay khi có dữ liệu thiếu hoặc bất thường trong các biến số. Những yếu tố này cần phải được xử lý kỹ lưỡng hơn để tăng độ chính xác của mô hình. Đồng thời, việc cải tiến các mô hình học sâu hay các thuật toán phức tạp hơn cũng là một thách thức lớn mà nhóm chúng tôi chưa kịp thực hiện.

Trong tương lai, nhóm chúng tôi sẽ tiếp tục phát triển mô hình dự đoán giá xe này bằng cách thử nghiệm với các phương pháp phức tạp hơn như hồi quy Ridge, Lasso, và các mô hình học sâu như mạng nơ-ron nhân tạo (RNN). Bên cạnh đó, nhóm chúng tôi cũng sẽ mở rộng bộ dữ liệu để tăng tính đa dạng và khả năng tổng quát của mô hình. Nhóm chúng tôi dự định sẽ cải tiến khả năng xử lý dữ liệu thiếu và cải thiện độ chính xác của các dự đoán thông qua các kỹ thuật xử lý nâng cao và tối ưu hóa mô hình.

TÀI LIỆU THAM KHẢO

[1] Vũ Hữu Tiếp: Machine Learning cơ bản. Funda, 2016.

[2] “Data Analytics Made Accessible” Dr.Anil Maheshwari - NXB TechWorld, 2023.

[3] Python Documentation - https://docs.python.org/3/ . Lần truy cập gần nhất: 1/12/2024.

[4] Bài 3: Linear Regression. URL: https://machinelearningcoban.com/2016/12/28/linearregression/ . Lần truy cập gần nhất: 18/1/2024.

[5] Khái niệm về tóm lược dữ liệu - https://toolkit.ncats.nih.gov/glossary/data-Summary/. Lần truy cập gần nhất: 18/1/2024.

[6] Khái niệm về làm sạch dữ liệu. URL: https://www.tableau.com/learn/articles/what-is-data-cleaning . Lần truy cập gần nhất: 21/1/2024.

[7] Gradio. URL: https://www.gradio.app/guides/quickstart. Lần truy cập gần nhất: 31/1/2024.

[8] Car-Price-Prediction. URL: https://github.com/sagnikghoshcr7/Car-Price-Prediction/tree/master/data. Lần truy cập gần nhất: 1/12/2024.