|  |  |
| --- | --- |
|  |  |
| LÊ ĐẮC ĐẠT | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN ĐẠI HỌC NGÀNH HỆ THỐNG THÔNG TIN |
|  |
| **XÂY DỰNG MÔ HÌNH AI DỰ ĐOÁN GIÁ TRỊ BẤT ĐỘNG SẢN CHO CÔNG TY TNHH SUNSHINE GROUP** |
|  |
|  |
| |  |  | | --- | --- | | **CBHD:** | THS. Mai Thanh Hồng | | **Sinh viên:** | Lê Đắc Đạt | | **Mã số sinh viên:** | 2020605449 | |
| HỆ THỐNG THÔNG TIN |  |
|  |
|  |
|  |
|  |
| Hà Nội – 2024 |
|  |

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến Trường đại Công Nghiệp Hà Nội và khoa Công nghệ thông tin đã tạo điều kiện cho em được thực hiện đồ án này. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến cô Mai Thanh Hồng đã hướng dẫn tận tình cho em trong suốt thời gian vừa qua. Trong thời gian học tập em đã được cô truyền đạt những kiến thức, kỹ năng hết sức quý báu, bổ ích và đó sẽ là hành trang để em có thể vững bước trên con đường tương lai.

Sự tận tâm và kiến thức sâu rộng của cô không chỉ giúp chúng em hiểu sâu hơn về các khái niệm phức tạp trong Machine Learning mà còn truyền đạt những kinh nghiệm quý báu từ thực tế, giúp chúng tôi áp dụng kiến thức vào thực tế một cách hiệu quả. Sự hỗ trợ và sự hướng dẫn chi tiết của thầy cô đã giúp chúng em vượt qua những khó khăn trong quá trình tiến hành nghiên cứu và phân tích dữ liệu. Sự cởi mở và sẵn lòng giúp đỡ của thầy cô đã là nguồn động viên quý giá cho chúng em.

Lời động viên và sự khích lệ từ thầy cô là động lực lớn để chúng em không ngừng nỗ lực và hoàn thành bài tập một cách thành công. Sự tin tưởng và niềm tin đã giúp chúng em tự tin hơn trong việc khám phá và thử nghiệm các phương pháp mới. Bài tập này không chỉ là một cơ hội để áp dụng những kiến thức đã học mà còn là một trải nghiệm quý báu giúp chúng em phát triển kỹ năng làm việc nhóm, tự học và giải quyết vấn đề. Lời cảm ơn của chúng em không đủ để bày tỏ lòng biết ơn sâu sắc đến tất cả các thầy cô đã dành thời gian và nỗ lực để hỗ trợ chúng em trong suốt quá trình này.

Chúng em hy vọng rằng những kinh nghiệm và kiến thức thu được từ bài tập này sẽ là nền tảng cho sự phát triển và thành công trong tương lai của chúng tôi. Một lần nữa, xin chân thành cảm ơn tới tất cả các thầy cô!

Em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc166838606)

[MỤC LỤC 2](#_Toc166838607)

[DANH MỤC HÌNH ẢNH 4](#_Toc166838608)

[DANH MỤC BẢNG BIỂU 6](#_Toc166838609)

[DANH MỤC VIẾT TẮT 6](#_Toc166838610)

[MỞ ĐẦU 7](#_Toc166838611)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc166838612)

[1.1. Giới thiệu chung 8](#_Toc166838613)

[1.2. Lý do chọn đề tài 8](#_Toc166838614)

[1.3. Giới thiệu về bộ dữ liệu Dataset 10](#_Toc166838615)

[1.4. Mục tiêu và phạm vi của đề tài 12](#_Toc166838616)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 14](#_Toc166838617)

[2.1. Tổng quan về khoa học dữ liệu 14](#_Toc166838618)

[2.1.1. Dữ liệu là gì? 14](#_Toc166838619)

[2.1.2. Tổng quan về KHDL 15](#_Toc166838620)

[2.1.3. Quá trình phát triển 16](#_Toc166838621)

[2.1.4. Ứng dụng 18](#_Toc166838622)

[2.1.5. Quy trình làm việc của khoa học dữ liệu 18](#_Toc166838623)

[2.1.6. Mục tiêu của khoa học dữ liệu 21](#_Toc166838624)

[2.2. Học máy (Machine Learning) là gì? 21](#_Toc166838625)

[2.2.1. Tại sao cần học máy? 22](#_Toc166838626)

[2.2.2. Học máy có bao nhiêu phương pháp? 23](#_Toc166838627)

[2.2.3. Các chức năng của học máy? 24](#_Toc166838628)

[2.2.4. Ứng dụng của học máy? 25](#_Toc166838629)

[2.3. Công cụ và ngôn ngữ sử dụng trong học máy? 25](#_Toc166838630)

[2.3.1. Giới thiệu về ngôn ngữ Python 25](#_Toc166838631)

[2.3.2. Giới thiệu công cụ và môi trường lập trình Python 27](#_Toc166838632)

[CHƯƠNG 3: NỘI DUNG VÀ KẾT QUẢ ĐỒ ÁN 32](#_Toc166838633)

[3.1. Các thư viện hỗ trợ 32](#_Toc166838634)

[3.1.1. Numpy 32](#_Toc166838635)

[3.1.2. Pandas 33](#_Toc166838636)

[3.1.3. Matplotlib 33](#_Toc166838637)

[3.1.4. Seaborn 34](#_Toc166838638)

[3.1.5. Scikit-learn 34](#_Toc166838639)

[3.2. Các bước thực hiện đề tài 35](#_Toc166838640)

[3.2.1. Thu thập dữ liệu (Data collection) 35](#_Toc166838641)

[3.2.2. Làm sạch và tiền xử lý dữ liệu (Data cleaning and Data processing) 35](#_Toc166838642)

[3.2.3. Khai phá và trực quan hóa dữ liệu ( Data mining và Data visualization ) 43](#_Toc166838643)

[3.2.4. Chuẩn bị dữ liệu cho mô hình 58](#_Toc166838644)

[3.3. Lựa chọn và huấn luyện mô hình 62](#_Toc166838645)

[3.4. Đánh giá kết quả mô hình 65](#_Toc166838646)

[KẾT LUẬN 75](#_Toc166838647)

[TÀI LIỆU THAM KHẢO 77](#_Toc166838648)

DANH MỤC HÌNH ẢNH

[**Hình 2.1:** Các lĩnh vực của khoa học dữ liệu 16](#_Toc165841330)

[**Hình 2.2:** Vòng đời của KHDL 21](#_Toc165841331)

[**Hình 2.3:** Công cụ Anaconda 29](#_Toc165841332)

[**Hình 2.4:** Jupyter Notebook 30](#_Toc165841333)

[**Hình 2.5:** Jupyter Lab 30](#_Toc165841334)

[**Hình 2.6:** Cài đặt Jupyter Notebook 31](#_Toc165841335)

[**Hình 3.1:** Các thư viện hỗ trợ 37](#_Toc165841336)

[**Hình 3.2:** Đọc dữ liệu từ file csv 38](#_Toc165841337)

[**Hình 3.3:** Chi tiết bộ dữ liệu huấn luyện 38](#_Toc165841338)

[**Hình 3.4:** Dự liệu dạng Numerical 39](#_Toc165841339)

[**Hình 3.5:** Dữ liệu dạng Categorical 39](#_Toc165841340)

[**Hình 3.6:** Missing value tập huấn luyện 40](#_Toc165841341)

[**Hình 3.7:** Bảng thống kê vissing value tập huấn luyện 41](#_Toc165841342)

[**Hình 3.8:** Kết quả xử lý missing value 44](#_Toc165841343)

[**Hình 3.9:** Mô tả thống kê 46](#_Toc165841344)

[**Hình 3.10:** Phân phối dữ liệu của biến mục tiêu “Saleprice” 49](#_Toc165841345)

[**Hình 3.11:** Phương pháp IQR – xử lý Outliers 51](#_Toc165841346)

[**Hình 3.12:** Kết quả xử lí độ lệch của dữ liệu 52](#_Toc165841347)

[**Hình 3.13:** Mối tương quan của “SalePrice” và các trường khác 53](#_Toc165841348)

[**Hình 3.14:** Biểu đồ phân rã mối quan hệ giữa “GrLivArea” và “SalePrice” 54](#_Toc165841349)

[**Hình 3.15:** Biểu đồ phân rã mối quan hệ giữa “GarageArea” và “SalePrice” 54](#_Toc165841350)

[**Hình 3.16:** Biểu đồ phân rã mối quan hệ giữa “TotalBsmtSF” và “SalePrice” 55](#_Toc165841351)

[**Hình 3.17:** Biểu đồ phân rã mối quan hệ giữa “1stFlrSF” và “SalePrice” 55](#_Toc165841352)

[**Hình 3.18:** Biểu đồ phân rã tuyến tính giữa các trường so với biến mục tiêu 56](#_Toc165841353)

[**Hình 3.19:** Thống kê số lượng trường “Neighborhood” 57](#_Toc165841354)

[**Hình 3.20:** Thống kê số lượng trường “BldgType” 57](#_Toc165841355)

[**Hình 3.21:** Thống kê số lượng trường “GarageCond” 58](#_Toc165841356)

[**Hình 3.22:** Thống kê số lượng trường “GarageType” 59](#_Toc165841357)

[**Hình 3.23:** Kết quả chuẩn hóa dữ liệu dạng Categorical 61](#_Toc165841358)

[**Hình 3.24:** Kết quả sau khi co dãn dữ liệu 62](#_Toc165841359)

[**Hình 3.25:** Hàm dùng để huấn luyện và đánh giá hiệu suất 66](#_Toc165841360)

[**Hình 3.26:** Biểu đồ đánh giá hiệu suất mô hình Lasso Regression 68](#_Toc165841361)

[**Hình 3.27:** Biểu đồ đánh giá hiệu suất mô hình Ridge Regression 69](#_Toc165841362)

[**Hình 3.28:** Biểu đồ đánh giá hiệu suất mô hình XGBoost 70](#_Toc165841363)

[**Hình 3.29:** Kết quả dự đoán của mô hình Lasso Regression 72](#_Toc165841364)

[**Hình 3.30:** Kết quả dự đoán của mô hình Ridge Regression 73](#_Toc165841365)

[**Hình 3.31:** Kết quả dự đoán của mô hình XGBoost Regression 73](#_Toc165841366)

[**Hình 3.32:** Biểu đồ so sánh giá thực tế và giá dự đoán của mô hình Lasso Regression 74](#_Toc165841367)

[**Hình 3.33:** Biểu đồ so sánh giá thực tế và giá dự đoán của mô hình Ridge Regression 75](#_Toc165841368)

[**Hình 3.34:** Biểu đồ so sánh giá thực tế và giá dự đoán của mô hình XGBoost Regression 76](#_Toc165841369)

DANH MỤC BẢNG BIỂU

[**Bảng 1.1** Bảng chú thích ký hiệu viết tắt 6](#_Toc165842308)

[**Bảng 2.1** Thông tin về bộ dữ liệu 10](#_Toc165842309)

DANH MỤC VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký Hiệu** | **Giải Thích** |
| 1 | Dataset | Bộ dữ liệu |
| 2 | KHDL | Khoa học dữ liệu |
| 3 | ML | Machine learning |
| 4 | SQL | Structured Query Language |
| 5 | MSE | Mean Squared Error |
| 6 | MAE | Mean Absolute Error |
| 7 | R2 | R-squared |

MỞ ĐẦU

Trong thời đại số hóa ngày nay, lĩnh vực công nghệ thông tin và khoa học dữ liệu đang trở thành trọng tâm của nhiều nỗ lực nghiên cứu và ứng dụng trên toàn cầu. Sự phát triển của các thuật toán máy học và trí tuệ nhân tạo đã mở ra không gian rộng lớn cho việc áp dụng chúng trong nhiều lĩnh vực, đặc biệt là trong lĩnh vực dự đoán giá cả bất động sản.

Định giá bất động sản luôn là một vấn đề quan trọng trong lĩnh vực đầu tư và giao dịch bất động sản. Sự biến động của thị trường, cùng với các yếu tố kinh tế, xã hội và chính sách, tạo ra những thách thức đối với việc đưa ra dự đoán chính xác về giá trị tài sản. Trong bối cảnh này, sự kết hợp giữa khoa học dữ liệu và trí tuệ nhân tạo mang lại tiềm năng lớn để tạo ra các mô hình dự đoán hiệu quả hơn về giá nhà bất động sản.

Đồ án này tập trung vào việc xây dựng và đánh giá một mô hình trí tuệ nhân tạo nhằm dự đoán giá nhà bất động sản dựa trên các yếu tố kỹ thuật, vị trí địa lý, và các thông tin liên quan khác. Qua đó, mục tiêu là tăng cường khả năng dự báo đối với các bên liên quan trong thị trường bất động sản, từ các nhà đầu tư đến người tiêu dùng.

Đồ án này không chỉ đơn thuần là việc áp dụng các kỹ thuật dữ liệu và trí tuệ nhân tạo, mà còn là một nỗ lực nghiên cứu để hiểu rõ hơn về cách mà các phương pháp này có thể ứng dụng hiệu quả vào bối cảnh thực tiễn của thị trường bất động sản. Ngoài ra, việc phát triển mô hình này cũng sẽ mở ra những hướng đi mới trong nghiên cứu và ứng dụng trí tuệ nhân tạo trong lĩnh vực định giá tài sản.

# TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu chung

Trong thập kỷ gần đây, sự phát triển vượt bậc của machine learning đã tạo ra những tiến bộ đáng kể trong việc giải quyết các vấn đề phức tạp trong nhiều lĩnh vực, bao gồm cả thị trường bất động sản. Việc dự đoán giá nhà dựa trên các yếu tố như kích thước, vị trí, tiện ích xung quanh, và các yếu tố khác đã trở thành một ứng dụng phổ biến của machine learning trong thực tế.

Trong bài tập lớn này, chúng em sẽ khám phá việc áp dụng machine learning để dự đoán giá nhà. Bằng cách sử dụng tập dữ liệu chứa thông tin về các căn nhà cùng với giá bán tương ứng, chúng ta sẽ xây dựng một mô hình dự đoán giá nhà dựa trên các đặc trưng của căn nhà.

Mục tiêu của bài tập này là xây dựng một mô hình machine learning hiệu quả có khả năng dự đoán giá nhà dựa trên các đặc trưng của căn nhà. Mô hình này sẽ được huấn luyện trên tập dữ liệu đã được gán nhãn với giá bán thực tế của các căn nhà, sau đó sẽ được đánh giá dựa trên các phương pháp đánh giá hiệu suất thích hợp.

## Lý do chọn đề tài

Trong lĩnh vực bất động sản, việc định giá nhà là một thách thức đối với cả người mua và người bán. Giá nhà phụ thuộc vào nhiều yếu tố như vị trí, kích thước, chất lượng xây dựng, và các yếu tố môi trường xung quanh. Trong nền kinh tế hiện đại, dự đoán giá nhà dựa trên các yếu tố này đã trở thành một công việc quan trọng và phức tạp. Mục tiêu là đưa ra một mô hình dự đoán giá nhà chính xác và hiệu quả, có khả năng cung cấp thông tin hữu ích cho người mua và người bán trong quyết định mua bán và đầu tư vào bất động sản.

Để có một mô hình dự đoán giá nhà tốt, chúng ta cần xem xét một loạt các yếu tố ảnh hưởng đến giá nhà, bao gồm các thông tin chi tiết về diện tích, vị trí địa lý, tiện ích xung quanh, tình trạng nhà, và nhiều yếu tố khác. Đồng thời, việc xử lý và phân tích dữ liệu cũng đóng vai trò quan trọng trong việc xây dựng một mô hình chính xác

Dự án này sẽ tập trung vào việc xây dựng một mô hình dự đoán giá nhà dựa trên một tập dữ liệu chi tiết về các yếu tố ảnh hưởng đến giá nhà. Bằng cách sử dụng các kỹ thuật học máy và phân tích dữ liệu, chúng em hy vọng có thể cung cấp cho người dùng một công cụ hữu ích để dự đoán giá nhà dựa trên thông tin có sẵn.

Sau một thời gian tham khảo về các đề tài về dự đoán và nhận dạng trên Kaggle cũng như các nền tảng khác, chúng em đã quyết định lựa chọn một đề tài về bài toán dự đoán giá nhà bất động sản vì những lý do sau:

* Bộ dữ liệu Dataset chứa nhiều thông tin về giá nhà tại nhiều khu vực khác nhau ở California. Chúng em sẽ có cơ hội thực hiện phân tích và dự đoán giá nhà dựa trên nhiều yếu tố như độ tuổi trung bình của nhà, thu nhập trung bình, số phòng ngủ,.v.v.
* Dự đoán giá nhà là một thách thức thực tế và có ứng dụng rộng rãi trong thị trường bất động sản. Kỹ thuật dự đoán giá nhà có thể được ứng dụng trong các dự đoán thực tế như hỗ trợ việc quyết định mua bán nhà, đầu tư bất động sản,.v.v.
* Thực hiện dự đoán giá nhà bằng dữ liệu thực tế giúp sinh viên chúng em trau dồi, phát triển kỹ năng phân tích dữ liệu, xử lý dữ liệu và xây dựng mô hình dự đoán. Sẽ có cơ hội được làm việc nhiều với các thuật toán của machine learning như hồi quy,.v.v.
* Và cuối cùng, đa số các nhóm khác đều chọn đề tài về nhận dạng nên chúng em quyết định sẽ chọn một bài toán khác là dự đoán để làm đa dạng hơn về các đề tài trong bài báo cáo môn học. Chúng em hy vọng rằng kết quả của đề tài sẽ không chỉ là thành tựu cá nhân mà còn mang lại giá trị và ý nghĩa cho cộng đồng và ngành công nghiệp mà chúng em hướng đến. Điều này là nguồn động viên lớn, giúp chúng em không ngừng nỗ lực để đạt được những kết quả tích cực và bổ ích nhất từ đề tài này.

## Giới thiệu về bộ dữ liệu Dataset

**Thông tin cơ bản:**

Bộ dữ liệu Ames Housing được Dean De Cock biên soạn để sử dụng trong giáo dục khoa học dữ liệu. Đây là một giải pháp thay thế tuyệt vời dành cho các nhà khoa học dữ liệu đang tìm kiếm phiên bản mở rộng và hiện đại hóa của bộ dữ liệu Nhà ở Boston thường được trích dẫn.

Yêu cầu người mua nhà mô tả ngôi nhà mơ ước của họ, và họ có thể sẽ không bắt đầu bằng chiều cao của trần tầng hầm hoặc vị trí gần đường sắt đông-tây. Nhưng tập dữ liệu của cuộc thi sân chơi này chứng minh rằng ảnh hưởng đến việc đàm phán giá cả nhiều hơn số lượng phòng ngủ hoặc hàng rào cọc trắng.

Với 79 biến giải thích mô tả (gần như) mọi khía cạnh của nhà ở ở Ames, Iowa, cuộc thi này thách thức bạn dự đoán giá cuối cùng của mỗi ngôi nhà.

**Mô tả chi tiết bộ dữ liệu:**

Thông tin về bộ dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** |
| 1 | SalePrice | Giá bán tài sản bằng dollars |
| 2 | MSSubClass | Lớp xây dựng |
| 3 | MSZoning | Phân loại phân vùng chung |
| 4 | LotArea | Kích thước lô đất (feet) |
| 5 | Alley | Loại đường vào ngõ |
| 6 | GrLivArea | Diện tích sinh hoạt (feet) |
| 7 | GarageArea | Kích thước của gara(m2) |
| 8 | TotalBsmtSF | Tổng mét vuông diện tích tầng hầm |
| 9 | 1stFlrSF | Diện tích tầng 1(feet) |
| 10 | 2ndFlrSF | Diện tích tầng 2 (feet) |
| 11 | MasVnrArea | Diện tích ván lạng(feet) |
| 12 | OverallQual | Chất lượng vật liệu và hoàn thiện tổng thể |
| 13 | Neighborhood | Các vị trí giới hạn trong thành phố |
| 14 | BldgType | Loại nhà ở |
| 15 | GarageCond | Điều kiện của gara |
| 16 | GarageType | Loại gara |
| 17 | YearBuilt | Năm xây dựng |
| 18 | YearRemodAdd | Ngày sửa sang lại |
| 19 | MoSold | Tháng bán được |
| 20 | YrSold | Năm bán được |
| 21 | Electrical | Hệ thống điện |
| 22 | KitchenQual | Chất lượng nhà bếp |
| 23 | Exterior1st | Lớp phủ bên ngoài ngôi nhà |
| 24 | Exterior2nd | Lớp phủ loại 2 |
| 25 | SaleType | Loại hình bán |
| 26 | PoolArea | Diện tích hồ bơi(feet) |
| 27 | GarageArea | Diện tích gara |
| 28 | Fireplaces | Số lượng lò sửa |
| 29 | FireplaceQu | Chất lượng lò sưởi |
| 30 | FullBath | Phòng tắm đầy đủ |
| 31 | BsmtFullBath | Phòng tắm đầy đủ ở tầng hầm |
| 32 | OpenPorchSF | Diện tích hiên mở |
| 33 | 3SsnPorch | Diện tích hiên nhà ba mùa |
| 34 | EnclosedPorch | Diện tích hiên bao quanh |
| 35 | ScreenPorch | Màn hình diện tích hiên nhà |

Ở đây chúng ta chỉ xét đến những trường dữ liệu có thể ảnh hưởng trực tiếp đến giá nhà, để hiểu chi tiết hơn về bộ dữ liệu hãy tham khảo ở nguồn tài liệu bên dưới Kaggle đã cung cấp đầy đủ thông tin chi tiết.

Nguồn:<https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

## Mục tiêu và phạm vi của đề tài

Dự án này đặt ra một số mục tiêu cụ thể để đảm bảo rằng quá trình xây dựng mô hình dự đoán giá nhà là hiệu quả và có giá trị thực sự cho cả người mua và người bán trong thị trường bất động sản. Dưới đây là một phân tích chi tiết về các mục tiêu của dự án:

* Xây dựng mô hình dự đoán đáng tin cậy: Một trong những mục tiêu quan trọng nhất của dự án là phát triển một mô hình dự đoán giá nhà có độ chính xác cao và đáng tin cậy. Mô hình này cần có khả năng dự đoán giá nhà một cách chính xác dựa trên các yếu tố ảnh hưởng đến giá nhà như vị trí, kích thước, chất lượng xây dựng, và các yếu tố môi trường khác.
* Hỗ trợ quyết định mua bán thông minh: Mục tiêu tiếp theo là cung cấp cho người mua và người bán một công cụ hữu ích để họ có thể đưa ra quyết định mua bán và đầu tư vào bất động sản một cách thông minh và tự tin. Mô hình dự đoán giá nhà sẽ giúp họ có cái nhìn rõ ràng hơn về giá trị thực sự của căn nhà mà họ quan tâm.
* Tích hợp công nghệ và phát triển ứng dụng: Dự án cũng nhằm mục đích tích hợp các công nghệ tiên tiến và các kỹ thuật học máy để xây dựng mô hình dự đoán giá nhà. Sau đó, mô hình sẽ được triển khai trong một ứng dụng hoặc dịch vụ trực tuyến để người dùng có thể truy cập và sử dụng dễ dàng.
* Đánh giá và cải thiện hiệu suất: Cuối cùng, một phần quan trọng của dự án là đánh giá hiệu suất của mô hình dự đoán giá nhà và thực hiện các biện pháp cải thiện nếu cần thiết. Điều này bao gồm việc thực hiện các phân tích và kiểm tra để đảm bảo rằng mô hình hoạt động hiệu quả và có thể được cải thiện theo thời gian.
* Bằng cách đạt được những mục tiêu này, dự án sẽ mang lại giá trị thực sự cho cả người mua, người bán và các nhà đầu tư trong lĩnh vực bất động sản, đồng thời góp phần nâng cao hiệu quả và minh bạch của thị trường bất động sản.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về khoa học dữ liệu

### Dữ liệu là gì?

Dữ liệu là một khái niệm trừu tượng, thường được hiểu khái quát là một tổ hợp thông tin trong đó có hình ảnh, ký tự, số,... thường được lưu trữ và xử lý cho ra thông tin, phục vụ những mục đích khác nhau. Dữ liệu được chia làm hai phần: dữ liệu có cấu trúc và dữ liệu phi cấu trúc:

**Dữ liệu có cấu trúc:**

* Đây là loại dữ liệu dễ tìm kiếm, lưu trữ, phân tích và dễ sử dụng, bởi nó thường được hàm chứa trong các hàng và cột. Một trong những ví dụ điển hình của dữ liệu có cấu trúc là dữ liệu trong các tệp excel, dữ liệu tài chính như các giao dịch, chi tiết địa chỉ, thông tin nhân khẩu,... Chúng có thể được xây dựng bám theo một mô hình dữ liệu được tạo ra, cũng có thể được nhóm lại để tạo thành các mối liên hệ với nhau. Chẳng hạn như một bảng thống kê số lượng khách hàng trong một tháng, các khách hàng có thể được nhóm lại theo nhóm các sản phẩm họ tiêu dùng.
* Phần lớn dữ liệu được cho là có cấu trúc hiện nay chỉ chiếm trên dưới 20% tổng số dữ liệu thu được. Thông thường dữ liệu có cấu trúc được quản lý bằng Ngôn ngữ truy vấn có cấu trúc (Structured Query Language - SQL) - ngôn ngữ lập trình được phát triển bưởi IBM từ những năm 1970. Tổng quan về khoa học dữ liệu

**Dữ liệu không cấu trúc:**

* Chiếm phần lớn trong trọng số dữ liệu trên thế giới là dữ liệu phi cấu trúc. Một số ví dụ bao gồm: hình ảnh, phim, các tệp âm thanh, các tệp chữ cái, nội dung mạng xã hội, hình ảnh từ vệ tinh,...”
* Loại dữ liệu này không thể hàm chứa trong cơ sở dữ liệu hàng và cột như dữ liệu có cấu trúc, không được xây dựng trên một mô hình dữ liệu nhất định. Sự thiếu hụt về cấu trúc khiến nó trở nên khó quản lý và phân tích cho đến khi trí tuệ nhân tạo ra đời dạo gần đây, nâng tầm khả năng xử lý bằng các thuật toán khiến quá trình quản lý loại dữ liệu này trở nên dễ dàng hơn. Ngày nay dữ liệu không cấu trúc được xem là một mảnh đất màu mỡ và là nguồn tài nguyên vô cùng giá trị đối với các doanh nghiệp, tổ chức.

Dữ liệu đã dần trở thành phần không thể thiếu trong xã hội của truyền thông và số hóa. Song song đó, với sự phát triển của công nghệ, việc quản lý dữ liệu ngày càng trở nên dễ dàng hơn.

### Tổng quan về KHDL

Khoa học dữ liệu là môn khoa học nghiên cứu việc quản trị và phân tích dữ liệu. Do đó, có thể hiểu Data Science là một lĩnh vực liên quan đến xử lý dữ liệu, phân tích và trích xuất thông tin chi tiết từ dữ liệu bằng các phương pháp thống kê và thuật toán máy tính khác nhau. Đây là một lĩnh vực đa ngành kết hợp Toán học, Thống kê và Khoa học máy tính.



Các lĩnh vực của khoa học dữ liệu

### Quá trình phát triển

Thuật ngữ “Khoa học dữ liệu” được tạo ra vào đầu những năm 1960 để mô tả một lĩnh vực mới hỗ trợ việc hiểu và giải thích một lượng lớn dữ liệu đang được tích lũy vào thời điểm đó. (Vào thời điểm đó, không có cách nào để dự đoán lượng dữ liệu khổng lồ thực sự trong vòng 50 năm tới.) Khoa học dữ liệu tiếp tục phát triển như một ngành học sử dụng khoa học máy tính và phương pháp thống kê để đưa ra những dự đoán hữu ích và hiểu biết sâu sắc về nhiều lĩnh vực khác nhau. lĩnh vực. Mặc dù Khoa học dữ liệu được sử dụng trong các lĩnh vực như thiên văn học và y học, nhưng nó cũng được sử dụng trong kinh doanh để giúp đưa ra các quyết định sáng suốt hơn. Khoa học dữ liệu bắt đầu với số liệu thống kê. Sau này ta thấy sự ra đời của các ngành mới gắn liền với số hóa, cách mạng công nghệ và Khoa học dữ liệu là nền tảng như trí tuệ nhân tạo, máy học và Internet vạn vật,... Với sự phát triển của Internet, Internet vạn vật và sự tăng trưởng theo cấp số nhân của lượng dữ liệu, việc sử dụng Big Data bắt đầu được áp dụng cho các lĩnh vực khác, chẳng hạn như y học, kỹ thuật và khoa học xã hội.

**Tổng quan lịch sử phát triển:**

* 1974: Peter Naur đề cập đến thuật ngữ “Khoa học dữ liệu” trong Khảo sát ngắn gọn về phương pháp máy tính (Concise Survey of Computer Methods).
* Vào năm 1994, Business Week đã đăng câu chuyện trang bìa: Tiếp thị cơ sở dữ liệu, tiết lộ rằng một số công ty đã bắt đầu thu thập một lượng lớn thông tin cá nhân, với kế hoạch bắt đầu các chiến dịch tiếp thị mới kỳ lạ. Cơn lũ dữ liệu khiến nhiều nhà quản lý công ty bối rối.
* Năm 1999, Jacob Zahavi đã chỉ ra sự cần thiết của các công cụ mới để xử lý lượng dữ liệu khổng lồ và không ngừng tăng lên dành cho các doanh nghiệp.
* Năm 2001, Phần mềm dịch vụ (Software as a Service-SaaS) được tạo ra. Đây là tiền thân của các ứng dụng đám mây.
* Năm 2001, William S. Cleveland vạch ra kế hoạch đào tạo các nhà khoa học dữ liệu để đáp ứng nhu cầu của tương lai.
* Năm 2002, Hội đồng Khoa học Quốc tế: Ủy ban Dữ liệu Khoa học và Công nghệ bắt đầu xuất bản Tạp chí Khoa học Dữ liệu, một ấn phẩm tập trung vào các vấn đề như mô tả hệ thống dữ liệu, xuất bản chúng trên internet, ứng dụng và các vấn đề pháp lý.
* Năm 2006, Hadoop 0.1.0, một cơ sở dữ liệu mã nguồn mở, phi quan hệ, đã được phát hành. Hadoop dựa trên Nutch, một cơ sở dữ liệu mã nguồn mở khác. Hadoop đã giải quyết hai vấn đề lớn là lưu trữ lượng dữ liệu khổng lồ và sau đó xử lý dữ liệu lưu trữ đó.
* Vào năm 2008, tiêu đề "chuyên viên khoa học dữ liệu" (Data scientist) đã trở thành một từ thông dụng.
* Trong năm 2011, danh sách việc làm cho các nhà khoa học dữ liệu đã tăng lên. Cũng có sự gia tăng các hội thảo và hội nghị dành riêng cho Khoa học dữ liệu và Big data. Khoa học dữ liệu đã chứng tỏ mình là một nguồn lợi nhuận và đã trở thành một phần của văn hóa doanh nghiệp.
* Vào năm 2013, IBM đã chia sẻ số liệu thống kê cho thấy 90% dữ liệu trên thế giới đã được tạo ra trong vòng hai năm qua.
* Vào năm 2015, bằng cách sử dụng các kỹ thuật Học sâu, tính năng nhận dạng giọng nói của Google, Google Voice, đã có một bước nhảy vọt về hiệu suất đáng kể là 49%. Vào năm 2015, Jack Clark của Bloomberg đã viết rằng đó là một năm mang tính bước ngoặt đối với trí tuệ nhân tạo (AI). Trong Google, tổng số dự án phần mềm sử dụng AI đã tăng từ "việc sử dụng lẻ tẻ" lên hơn 2.700 dự án trong năm.

### Ứng dụng

Xung quanh ta luôn có hàng loạt ứng dụng của khoa học dữ liệu, đã, đang và sẽ len lỏi vào mọi ngóc ngách trong cuộc sống hằng ngày của chúng ta. Khi mua hàng online trên các sàn thương mại điện tử, Amazon, Shopee, … phân tích các lần mua hàng trước của bạn và những món đồ bạn tìm kiếm để gửi quảng cáo tới. Việc điều trị cho bệnh nhân trong lĩnh vực y tế trở nên dễ dàng hơn rất nhiều khi có và phân tích được các bệnh án điện tử nhờ vào các phương pháp và đột phá của Khoa học dữ liệu. Một trong những ứng dụng nổi tiếng nhất của Khoa học dữ liệu là mạng xã hội Facebook và Twitter. Đằng sau những trải nghiệm kết nối và tương tác giữa bạn bè và người thân, đó là sự vận hành của các thuật toán đánh giá người dùng được xây dựng bởi những kỹ sư hàng đầu thế giới.

Thành phần của khoa học dữ liệu:

* Khoa học dữ liệu gồm 3 phần chính: tạo ra và quản trị dữ liệu; phân tích dữ liệu; chuyển kết quả phân tích thành giá trị của hành động. Việc phân tích và sử dụng dữ liệu chủ yếu dựa vào nền tảng tri thức toán học, công nghệ thông tin , và tri thức của lĩnh vực ứng dụng cụ thể.
* Không đơn thuần là làm việc với số liệu, Khoa học dữ liệu là sự tổng hợp đa dạng nhiều lĩnh vực. Trong đó, các lĩnh vực của khoa học dữ liệu có thể kể đến là Khai thác dữ liệu (Data mining), Thống kê (Statistic), Phân tích (Analyze) và Lập trình (Programming), Học máy (Machine learning).

### Quy trình làm việc của khoa học dữ liệu

#### Quy trình làm việc với khoa học dữ liệu (KHDL)

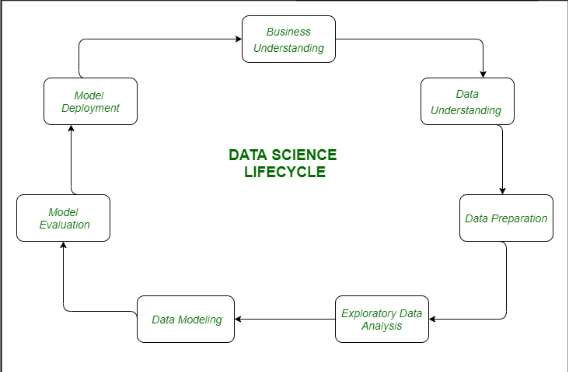
Các bước làm việc của Khoa học dữ liệu tổng quát:

* **Thu thập dữ liệu:** Dữ liệu được lấy từ các nguồn khả dụng khác nhau.
* Chuẩn bị dữ liệu: thường được xem như bước tiền xử lý dữ liệu, trong đó dữ liệu thô được làm sạch và tổ chức lại, cho ra dữ liệu chất lượng.
* Có nhiều cách để chúng ta có thể thu thập dữ liệu :
* Thu thập dữ liệu từ các trang web (Data Scraping )
* Thu thập dữ liệu từ các nguồn dữ liệu mở
* Thu thập dữ liệu từ các Log files
* **Xử lý dữ liệu:** Bước vô cùng quan trọng trong quy trình xử lý dữ liệu, có thể sử dụng các thuật toán học máy (machine learning).
* **Chuyển đồi dữ liệu:** Chuyển đổi dữ liệu là quá trình chuyển đổi dữ liệu từ một định dạng, chẳng hạn như tệp cơ sở dữ liệu, tài liệu XML hoặc bảng tính Excel sang một định dạng khác.
* **Phân tích dữ liệu:** Dữ liệu được làm sạch và tổ chức ở bước trước được đưa vào phân tích bằng các phương pháp khác nhau như dự báo, thống kê mô tả, …
* **Truyền đạt kết quả phân tích:** Giai đoạn diễn giải là giai đoạn mà dữ liệu cuối cùng có thể sử dụng được đối với những người không phải là chuyên viên khoa học dữ liệu. Nó có thể đọc được và thường ở dạng biểu đồ, video, hình ảnh, văn bản thuần túy, ...

#### Vòng đời của khoa học dữ liệu

Hầu hết dữ liệu được hình thành trong doanh nghiệp đều trải qua một vòng đời nhiều giai đoạn. Vòng đời của KHDL xoay quanh việc sử dụng máy học và các chiến lược phân tích khác nhau để tạo ra những hiểu biết và dự đoán từ thông tin nhằm đạt được mục tiêu của doanh nghiệp thương mại.

Vòng đời của KHDL tổng quát được mô tả qua biểu đồ:



Vòng đời của KHDL

- ***Hiểu biết về doanh nghiệp (business understanding):*** Xác định vấn đề của

doanh nghiệp liên đới đến vấn đề dữ liệu cần xử lý.

- ***Hiểu dữ liệu (data understanding):*** Sau khi hiểu doanh nghiệp, bước tiếp theo là hiểu dữ liệu. Bước này bao gồm mô tả dữ liệu, cấu trúc, mức độ liên quan, loại bản ghi của chúng.

- ***Chuẩn bị dữ liệu (data preparation):*** Tiếp theo là giai đoạn chuẩn bị dữ liệu.

Điều này bao gồm các bước như chọn dữ liệu áp dụng, tích hợp dữ liệu bằng

cách hợp nhất các bộ dữ liệu, làm sạch dữ liệu, xử lý các giá trị còn thiếu, xây

dựng dữ liệu mới, rút ra các phần tử mới từ những phần tử hiện tại, định dạng

dữ liệu theo cấu trúc nhất định, loại bỏ các tính năng không mong muốn. Chuẩn

bị dữ liệu là bước tốn nhiều thời gian nhất nhưng được cho là thiết yếu nhất

trong toàn bộ chu trình.

- ***Phân tích dữ liệu (exploratory data analysis):*** Nhiều chiến lược trực quan hóa dữ liệu được sử dụng đáng kể để khám phá từng đặc điểm riêng lẻ và bằng cách kết hợp chúng với các đặc điểm khác nhau.

- ***Mô hình hóa dữ liệu (data modeling):*** Mô hình hóa dữ liệu là cốt lõi của phân tích dữ liệu. Một mô hình lấy dữ liệu được tổ chức làm đầu vào và đưa ra đầu ra ưu tiên. Bước này bao gồm việc lựa chọn loại mô hình phù hợp, cho dù vấn đề là vấn đề phân loại, vấn đề hồi quy hay vấn đề phân cụm.

- ***Đánh giá mô hình (data evaluation):*** Ở đây, mô hình được đánh giá để kiểm tra xem nó có sẵn sàng để triển khai hay không. Việc đánh giá mô hình giúp chúng ta lựa chọn và xây dựng một mô hình lý tưởng.

- ***Triển khai mô hình (data deployment):*** Mô hình sau khi đánh giá nghiêm ngặt cuối cùng sẽ được triển khai trong cấu trúc và kênh ưu tiên. Đây là bước cuối cùng trong vòng đời của khoa học dữ liệu. Mỗi bước trong vòng đời khoa học dữ liệu được xác định ở trên phải được thực hiện cẩn thận. Nếu bất kỳ bước nào được thực hiện không đúng cách, sẽ ảnh hưởng đến bước tiếp theo và toàn bộ quá trình phân tích xử lý dữ liệu.

Ví dụ: Nếu thông tin không được làm sạch đúng cách, mô hình sẽ không hoạt động.

### Mục tiêu của khoa học dữ liệu

Mục tiêu của KHDL không chỉ là phân tích và xử lý dữ liệu thành mô hình kinh doanh, giúp đỡ các tổ chức doanh nghiệp cắt giảm chi phí mà còn là tạo ra sự đổi mới trong doanh nghiệp, xã hội và thói quen tiêu dùng thông qua các ứng dụng có liên quan mà nó tạo ra. Tương lai phát triển của KHDL sẽ hứa hẹn nhiều điều thú vị và có ý nghĩa lớn đối với cuộc sống, giúp cho con người tận dụng được tối đa tiềm năng của dữ liệu mang lại.

## Học máy (Machine Learning) là gì?

Học Máy (Machine Learning - ML) là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc phát triển và nghiên cứu các phương pháp và mô hình máy học có khả năng tự động học từ dữ liệu. Trong khi các hệ thống máy học truyền thống thường yêu cầu sự can thiệp của con người để định rõ các quy tắc

và quy luật, Học Máy giúp máy tính "học" từ kinh nghiệm và dữ liệu mà không

cần sự lập trình cụ thể.

Ba khái niệm chính trong Học Máy:

* Huấn luyện (Training): Quá trình huấn luyện máy tính bao gồm cung
* cấp cho nó một tập dữ liệu huấn luyện, trong đó mỗi mẫu dữ liệu được
* gán nhãn (label) hoặc có giá trị đầu ra mong muốn. Máy tính sẽ sử dụng
* các thuật toán học máy để tìm hiểu các mẫu và xây dựng một mô hình
* dự đoán.
* Mô hình (Model): Mô hình học máy là một biểu diễn toán học hoặc
* thuật toán mà máy tính sử dụng để dự đoán hoặc phân loại dữ liệu mới
* dựa trên thông tin đã học từ quá trình huấn luyện. Mô hình có thể là một
* cây quyết định, một mạng neural, một phương trình toán học, hoặc một
* thuật toán khác.
* Dự đoán (Prediction): Sau khi máy tính đã được huấn luyện, nó có thể
* sử dụng mô hình đã xây dựng để dự đoán hoặc phân loại dữ liệu mới mà nó chưa từng thấy trước đó. Quá trình này gọi là dự đoán hoặc gán nhãn (labeling) dữ liệu.

Trong học máy, một hệ thống máy tính được huấn luyện thông qua việc xem xét các dữ liệu đầu vào và tìm hiểu các mẫu, quy tắc hoặc cấu trúc ẩn trong dữ liệu đó. Máy tính sử dụng thông tin đã học được để tự động dự đoán hoặc đưa ra quyết định cho các dữ liệu mới mà nó chưa từng thấy trước đó.

### Tại sao cần học máy?

Học Máy đóng vai trò quan trọng trong việc giải quyết những thách thức phức tạp và khám phá những tri thức ẩn sau dữ liệu lớn mà con người không thể xử lý một cách hiệu quả bằng các phương pháp truyền thống. Một trong những đặc điểm quan trọng của Học Máy là khả năng tự học và cải thiện theo thời gian mà không cần phải lập trình cụ thể cho từng tác vụ.

Học Máy đã và đang thay đổi cách chúng ta làm việc và tư duy về dữ liệu. Trong lĩnh vực y tế, Học Máy có thể giúp dự đoán và chẩn đoán bệnh, tìm kiếm phác đồ điều trị tối ưu, và phát hiện sớm các nguy cơ sức khỏe. Trong ngành công nghiệp, Học Máy hỗ trợ quá trình tự động hóa, tối ưu hóa chuỗi cung ứng, và dự báo nhu cầu thị trường. Trong tài chính, nó giúp phân tích rủi ro, dự đoán xu hướng thị trường, và tối ưu hóa quyết định đầu tư.

Học Máy còn đóng vai trò lớn trong các ứng dụng hàng ngày như trích xuất thông tin từ văn bản, nhận diện giọng nói, và thậm chí là trong xe tự lái. Khả năng "học" từ dữ liệu và áp dụng kiến thức đã học để đưa ra dự đoán hoặc quyết định làm cho Học Máy trở thành một công cụ mạnh mẽ và linh hoạt…

### Học máy có bao nhiêu phương pháp?

Các mô hình học máy được chia thành ba loại chính:

#### Máy học có giám sát

Học máy có giám sát được hiểu là cách sử dụng các tập dữ liệu được gán nhãn để huấn luyện thuật phân loại hoặc dự đoán một kết quả chính xác.

Học tập có giám sát giúp các tổ chức giải quyết được nhiều vấn đề trong thực tế trên quy mô lớn. Một số phương pháp được sử dụng trong học có giám sát bao gồm mạng nơron, mô hình phân lớp (Naive Bayes), hồi quy tuyến tính, hồi quy logistic, rừng ngẫu nhiên (Random forest) và máy hỗ trợ (SVM-support vector machine).

Như trong dự án, chúng em sử dụng phương pháp học có giảm sát. Trong trường hợp của bài đồ án môn học dự đoán giá nhà, đầu vào (features) là các thuộc tính như độ tuổi trung bình nhà, số phòng ngủ,.v.v. Đầu ra (Label): sẽ là giá nhà, vì giá nhà là biến cần được dự đoán.

#### Học máy không giám sát

Học không giám sát, còn được gọi là học máy không giám sát, sử dụng các thuật toán học máy để phân tích và phân cụm các tập dữ liệu không được gán nhãn. Các thuật toán này phát hiện ra các mẫu hoặc nhóm dữ liệu ẩn mà không cần sự can thiệp của con người.

Khả năng phát hiện ra những điểm tương đồng và khác biệt trong dữ liệu của phương pháp này khiến nó trở nên lý tưởng cho việc phân tích dữ liệu khám phá, chiến lược bán chéo, phân tích phân khúc khách hàng,.v.v.

Nó cũng được sử dụng để giảm số lượng các tính năng trong một mô hình thông qua quá trình giảm kích thước. Phân tích thành phần chính (PCA-Principal component analysis) và phân tích giá trị đơn lẻ.

Các thuật toán khác được sử dụng trong học tập không giám sát bao gồm phân cụm k-means và các phương pháp phân cụm theo xác suất.

#### Học tập bán giám sát

Học tập bán giám sát là sự kết hợp giữa có giám sát và học không giám sát. Trong quá trình đào tạo, nó sử dụng một tập dữ liệu có nhãn nhỏ hơn học có giám sát để hướng dẫn phân loại, trích xuất tính năng từ một tập dữ liệu lớn hơn, không được gắn nhãn.

Học bán giám sát có thể giải quyết vấn đề trong trường hợp không có đủ dữ liệu được gán nhãn cho thuật toán học có giám sát.

### Các chức năng của học máy?

Dưới đây là các chức năng của học máy:

* Dự đoán và phân loại
* Dự đoán (Regression): Dự đoán một giá trị liên tục, ví dụ giá nhà dựa trên các thuộc tính nhất định.
* Phân loại (Classification): Phân loại dữ liệu vào các lớp hoặc nhóm khác nhau, ví dụ như phân loại email là spam hay không spam.
* Tự động hóa (Supervised Learning): Mô hình được huấn luyện trên tập dữ liệu có đầu vào và đầu ra được gán nhãn để học cách ánh xạ từ đầu vào đến đầu ra
* Phân cụm (Clustering)
* Tìm ra các nhóm tự nhiên trong dữ liệu mà không cần thông tin nhãn trước.
* Phân tích cụm giúp xác định sự tương đồng giữa các điểm dữ liệu và phân loại chúng vào các nhóm khác nhau.

Ngoài ra còn nhiều chức năng khác nữa nhưng chúng em sẽ không nêu ra hết tất cả chúng, một vài như: *Tối ưu hóa (Optimization), học tăng cường (Reinforcement Learning),.v.v*

Các chưc năng ở trên đóng vai trò quan trọng, đại diện cho sự đa dạng và sức mạnh của học máy, giúp học máy trở thành một công cụ quan trọng trong nhiều lĩnh vực như khoa học dữ liệu, trí tuệ nhân tọa và phân tích dữ liệu.

### Ứng dụng của học máy?

Học máy (Machine Learning - ML) đã ngày càng trở thành một công nghệ quan trọng và có ảnh hưởng sâu rộng đến nhiều lĩnh vực cuộc sống và doanh nghiệp. Dưới đây là một số ứng dụng chính của học máy:

* Nhận Diện Hình Ảnh và Video
* Dịch Ngôn Ngữ Tự Động
* Tư Vấn Y Tế và Chẩn Đoán
* Xử Lý Ngôn Ngữ Tự Nhiên
* Tổ Chức Dữ Liệu và Phân Loại
* Tự Động Hóa Sản Xuất và Quy Trình Sản Xuất
* Dự Đoán Đối Tượng Khách Hàng và Tiếp Thị
* Ô Tô Tự Lái và Giao Thông Thông Minh

## Công cụ và ngôn ngữ sử dụng trong học máy?

Bài tập sẽ được thực hiện bằng Python, với sự sử dụng của các thư viện phổ biến như NumPy, Pandas, và Scikit-learn. Sử dụng công cụ Anaconda, Google Colab, Jupyter Notebook… là một cách tiện lợi để tổ chức code và hiển thị kết quả một cách trực quan.

### Giới thiệu về ngôn ngữ Python

**Tổng quan về ngôn ngữ Python:**

Python lần đầu tiên được giới thiệu vào tháng 12/1989 bởi tác giả Guido van Rossum (Hà Lan), ông sinh năm 1956 và hiện đang làm việc tại Google.

Nó được kế thừa từ ngôn ngữ ABC, Python 2 được giới thiệu năm 2000 phiên bản kế tiếp python 3 phát hành năm 2008 và hiện đã có bản 3.9, Python 4 năm ra mắt dự kiến 2023 và phiên bản mới nhất hiện nay là 3.12.

Là ngôn ngữ có mã nguồn mở

Là ngôn ngữ kịch bản (scripting programming language):

* Thích hợp với DevOps ( Người viết code cũng là người vận hành)
* Khai báo biến tự nhiên , phong phú và động
* Nhiều phép tính cấp cao được cung cấp sẵn
* Thường được thông dịch thay vì phiên dịch
* Vừa hướng thủ tục , vừa hướng đối tượng
* Hỗ trợ module và hỗ trợ gói (package)
* Xử lý ngoại lệ ( exception )
* Kiểu dữ liệu động ở mức cao
* Có khả năng tương tác với các module viết bằng ngôn ngữ lập trình khác
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản

Ưu điểm của ngôn ngứ Python:

* Có ngữ pháp đơn giản, dễ đọc
* Viết mã ngắn gọn hơn những chương trình tương đương được viết trong C, C++, C#, Java,…
* Có các bộ thư viện chuẩn và các module ngoài, đáp ứng gần như mọi nhu cầu lập trình
* Có khả năng chạy trên nhiều nền tảng (Windows, Linux, Unix, OS/2, Mac, Amiga, máy ảo .NET, máy ảo Java, Nokia Series 60,…)
* Có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, mã nguồn chia sẻ nhiều

Nhược điểm:

* Chương trình chạy chậm
* Giao tiếp với các thư viện viết bằng các ngôn ngữ khác tương đối khó khăn
* Yếu trong hỗ trợ tính toán trên di động
* Cách viết khối lệnh dễ gây nhầm lẫn cho người mới bắt đầu lập trình
* Gỡ lỗi đòi hỏi kinh nghiệm
* Kém hỗ trợ các cơ sở dữ liệu

### Giới thiệu công cụ và môi trường lập trình Python

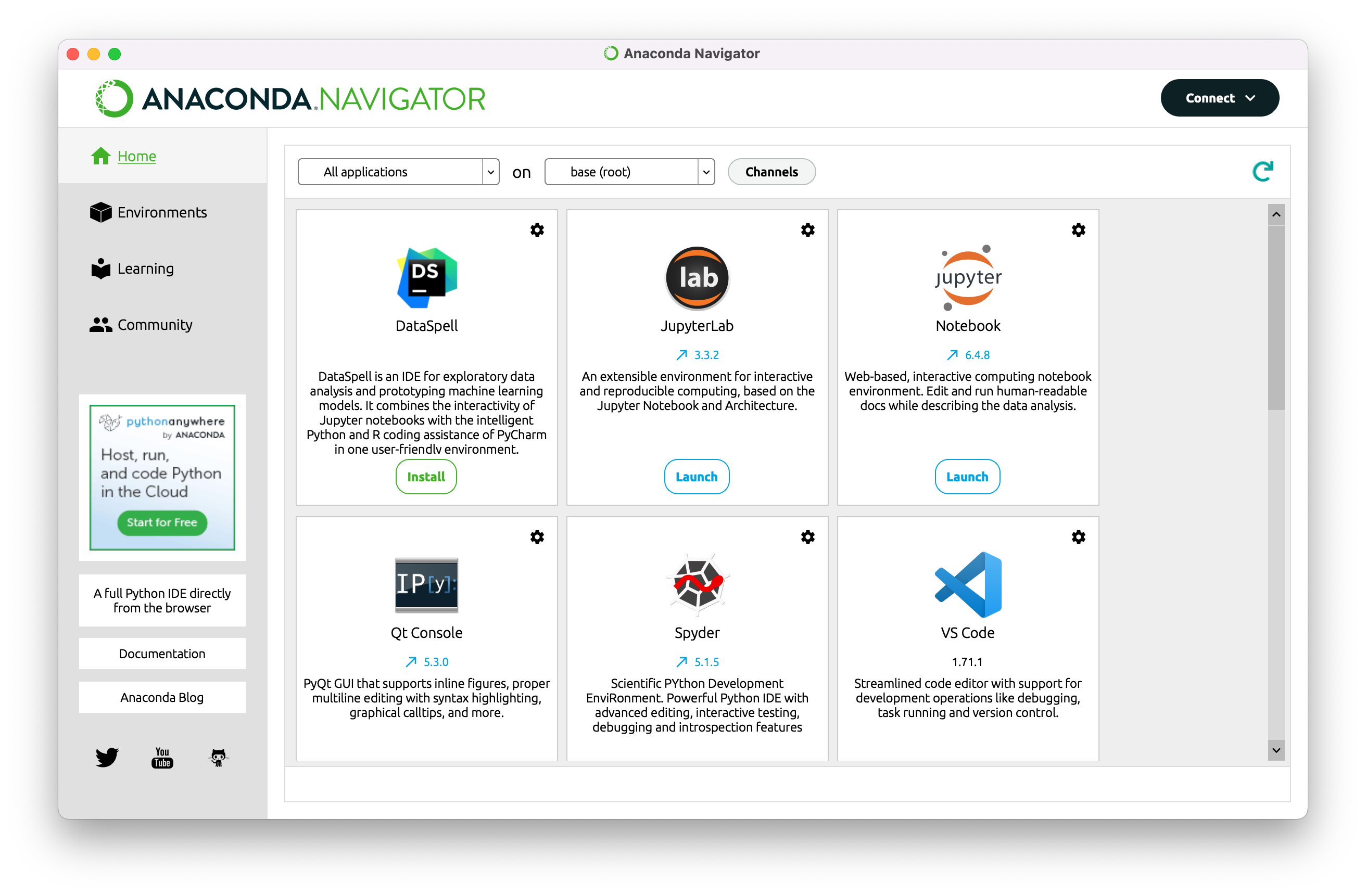
#### Anaconda

Anaconda là 1 nền tảng mã nguồn mở về Khoa học Dữ liệu sử dụng ngôn ngữ lập trình Python thông dụng nhất hiện nay. Với hơn 25 triệu người dùng (theo thống kê tại trang chủ), Anaconda là giải pháp nhanh nhất và dễ nhất để các bạn có thể tìm hiểu Khoa học Dữ liệu với Python hoặc R trên Windows, Linux và Mac OS X. Và ngày hôm nay, ZootoPi xin phép được khai bút chuỗi series Khoa học Dữ liệu cho người mới bắt đầu bằng những công cụ thiết yếu dành cho nhà Khoa học Dữ liệu, bắt đầu với Anaconda.

Anaconda đơn giản, dễ sử dụng với 1 hệ sinh thái nguồn mở, đáp ứng nhu cầu về khoa học dữ liệu với Python.

Thông thường, khi bạn cài Anaconda thì nó sẽ tự động cài thêm cho bạn Python và 1 số thư viện Python hữu ích như Matplotlib, NumPy và SciPy…

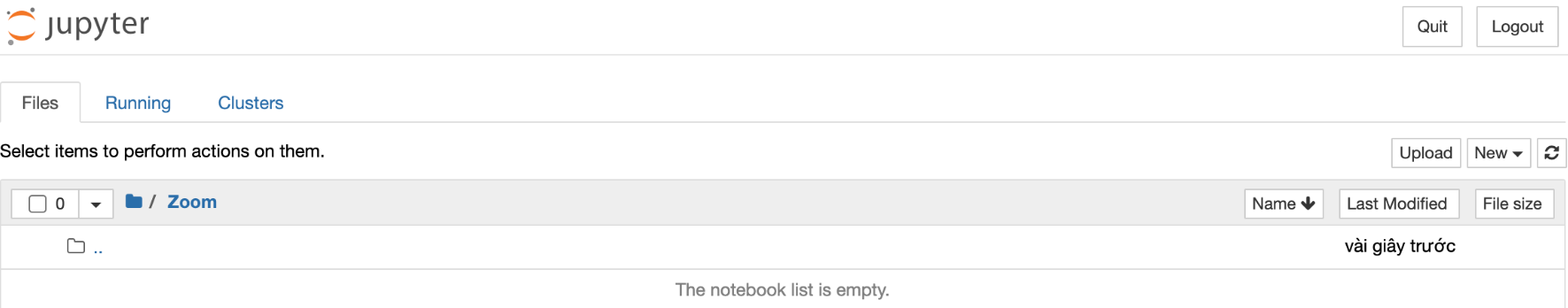
Cài đặt Anaconda tại: <https://www.anaconda.com/>



Công cụ Anaconda

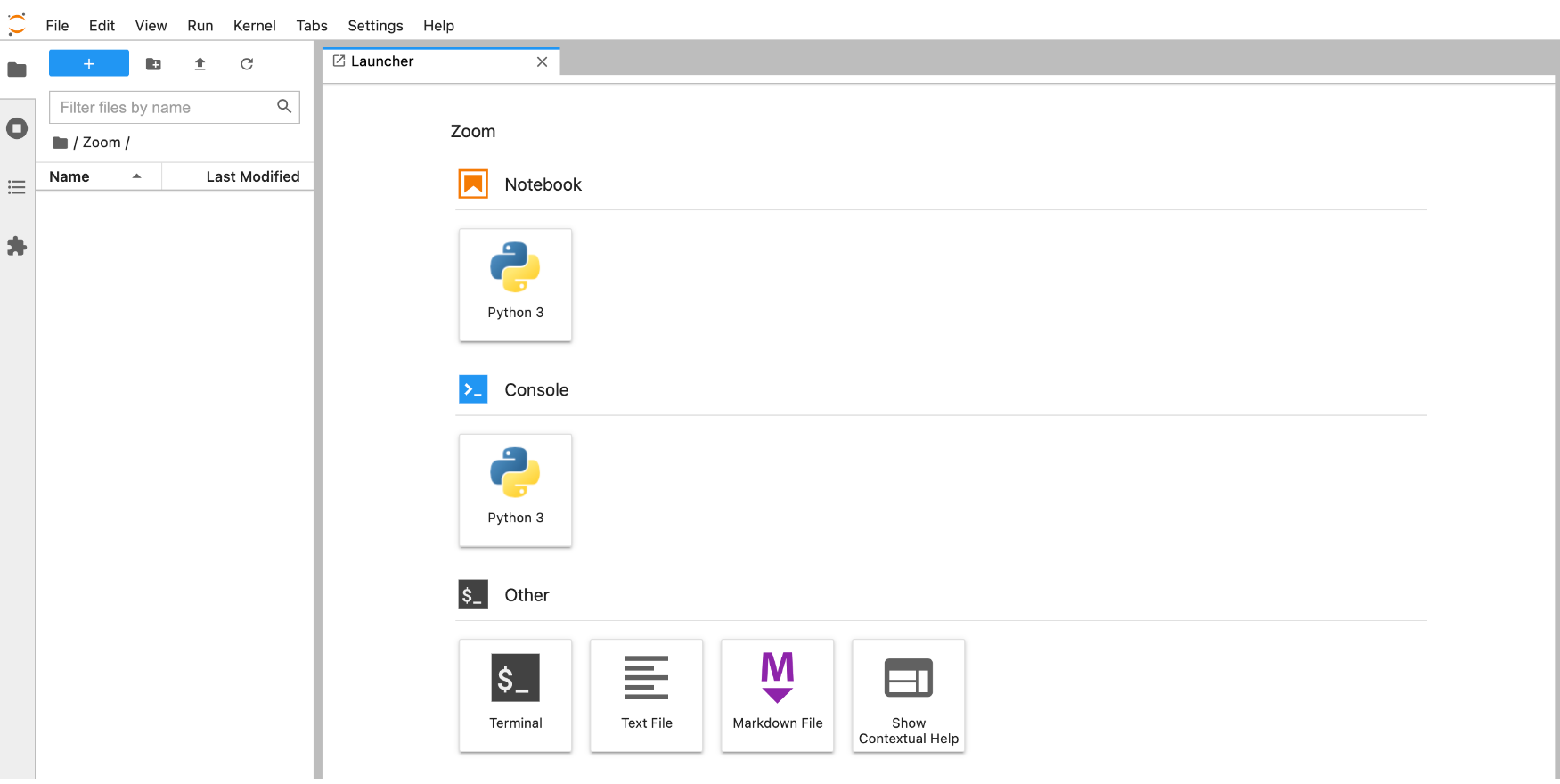
#### Jupyter Notebook

Jupyter Notebook (trước 2014 được biết với cái tên IPython Notebook) là một ứng dụng mã nguồn mở cho phép ta đưa cả mã nguồn Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức... vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác trên đó, cốt lõi của việc này chính là Markdown. Điều này giúp cho Jupyter Notebook được ưa chuộng trong việc phân tích dữ liệu, trực quan hóa dữ, xử lý và xây dựng mô hình trên dữ liệu v.v.



Jupyter Notebook

JupyterLab là môi trường phát triển tương tác dựa trên web dành cho notebook, mã và dữ liệu của Jupyter. Nó có cấu trúc mô-đun giúp ta có thể viết các plugin bổ sung các thành phần mới, tích hợp với các thành phần hiện có, và mở một số notebook hoặc tệp (ví dụ: HTML, Markdowns, v.v.) dưới dạng các tab trong cùng một cửa sổ cũng như cung cấp nhiều trải nghiệm giống như khi làm việc với các IDE. Điểm cộng của JupyterLab là sử linh hoạt, cho phép cấu hình và sắp xếp giao diện người dùng để hỗ trợ các quy trình trong khoa học dữ liệu, máy tính khoa học và máy học.



Jupyter Lab

Để có thể sử dụng Jupyter Notebook và JupyterLab, cách đơn giản nhất là cài phần mềm Anaconda. Sau khi cài đặt xong Anaconda, ta có thể cài Jupyter Notebook và Jupyter Lab trong hoặc sau khi tạo ra môi trường ảo của dự án.



Cài đặt Jupyter Notebook

#### Google Colab

Môi trường colab là 1 nền tảng được cung cấp miễn phí bởi Google và chạy trên nền Jupyter, giúp người dùng có thể dễ dàng mở và chạy Jupyter notebook từ bất kỳ đâu mà không cần cài đặt bất kỳ thứ gì.

Truy cập colab tại địa chỉ [https://colab.research.google.com/](https://colab.research.google.com/%20) và bắt đầu viết code.

Giao diện colab gần như giống y hệt với Jupyter notebook. Tuy nhiên tính năng thì vô vàn! Ví dụ khả năng kết nối tới Google Drive hay khả năng chia sẽ notebook ...

# NỘI DUNG VÀ KẾT QUẢ ĐỒ ÁN

## Các thư viện hỗ trợ

Để bắt đầu quy trình làm việc với khoa học dữ liệu nói chung và đề án xây dựng mô hình AI nói riêng bằng ngôn ngữ lập trình Python thì trước hết việc hiểu biết và cách sử dụng các thư viện hỗ trợ phân tích dữ liệu cũng hết sức quan trọng. Ở đây chúng em sẽ sử dụng một vài thư viện rất mãnh mẽ đã được tích hợp sẵn trong Python nhằm nâng cao hiệu quả và chất lượng công việc.

### Numpy

NumPy (Numerical Python) là một thư viện Python mã nguồn mở được sử dụng trong hầu hết các lĩnh vực khoa học và kỹ thuật. Đây là tiêu chuẩn chung để làm việc với dữ liệu số bằng Python và là tiêu chuẩn cốt lõi của hệ sinh thái Python và PyData. NumPy API được sử dụng rộng rãi trong Pandas, SciPy, Matplotlib, scikit-learning, scikit-image và hầu hết các gói Python khoa học và khoa học dữ liệu khác.

Thư viện NumPy chứa các cấu trúc dữ liệu ma trận và mảng đa chiều (bạn sẽ tìm thấy thêm thông tin về điều này trong các phần sau). NumPy có thể được sử dụng để thực hiện nhiều phép toán khác nhau trên mảng. Nó bổ sung các cấu trúc dữ liệu mạnh mẽ vào Python để đảm bảo tính toán hiệu quả với các mảng và ma trận, đồng thời nó cung cấp một thư viện khổng lồ các hàm toán học cấp cao hoạt động trên các mảng và ma trận này.

NumPy cung cấp cho bạn rất nhiều cách nhanh chóng và hiệu quả để tạo mảng (array) và thao tác dữ liệu số bên trong chúng. list có thể chứa các kiểu dữ liệu khác nhau, còn tất cả các phần tử trong array NumPy phải đồng nhất. Điều này cho phép các phép toán được thực hiện trên mảng sẽ cực kỳ kém hiệu quả nếu các mảng không đồng nhất. Cũng bởi vì thế, 1 array tiêu tốn ít bộ nhớ hơn, dẫn đến NumPy sử dụng ít bộ nhớ hơn nhiều để lưu trữ dữ liệu và nó cung cấp một cơ chế xác định các kiểu dữ liệu. Điều này cho phép mã được tối ưu hóa hơn nữa.

### Pandas

Pandas (viết tắt từ Panel Data - bảng dữ liệu) là thư viện mã nguồn mở phục vụ cho việc phân tích và xử lý dữ liệu trong Python, được phát triển bởi Wes McKinney trong năm 2008. Thư viện này được thiết kế để làm việc dễ dàng và trực quan với dữ liệu có cấu trúc (dạng bảng, đa chiều, ...) và dữ liệu chuỗi thời gian. Hiện nay, Pandas được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu.

Pandas trở thành thư viện yêu thích của những nhà phân tích dữ liệu bởi:

* Pandas phù hợp với nhiều loại dữ liệu khác nhau:
* Dữ liệu dạng bảng, như trong bảng SQL hoặc bảng tính Excel.
* Dữ liệu chuỗi thời gian theo thứ tự và không có thứ tự.
* Dữ liệu ma trận tùy ý với nhãn hàng và cột...
* Dễ dàng thao tác, phân tích, xử lý và trực quan hoá dữ liệu
* Khả năng xử lý dữ liệu mất mát (NaN,...), nhiễu, ...
* Khả năng thay đổi kích thước: chèn và xóa cột, dòng
* Khả năng căn chỉnh dữ liệu tự động và rõ ràng
* Khả năng phân tách, gộp nối, chuyển đổi, định hình các tập dữ liệu 1 cách linh hoạt giúp cho việc tổng hợp và phân tích dữ liệu nhanh gọn, dễ dàng hơn.
* Khả năng tải và lưu trữ dữ liệu theo nhiều format khác nhau: .csv, .txt, .excel, .pkl, .hdfs5, ...
* Khả năng xử lí dữ liệu dạng chuỗi.
* Khả năng tích hợp tốt với các thư viện khác của Python: SciPy, Matplotlib, Seaborn, Plotly, Sklearn, ...

### Matplotlib

* Matplotlib là thư viện package Python vẽ biểu đồ ở mức thấp (low level)
* Là thư viện Python trực quan hóa dữ liệu đầu tiên, cũng là thư viện được sử dụng rộng rãi nhất để vẽ biểu đồ trong cộng đồng Python.
* Matplotlib cực kỳ mạnh mẽ nhưng sức mạnh đó đi kèm với sự phức tạp
* Rất nhiều thư viện khác phát triển dựa trên Matplotlib (Pandas, Seaborn)
* Matplotlib được tạo bởi John D. Hunter năm 2003
* Là mã nguồn mở, dùng miễn phí
* Được thiết kế gần giống với MATLAB, một ngôn ngữ lập trình độc quyền phát triển vào những năm 1980
* Tham khảo: <https://matplotlib.org/stable/gallery/index.html>

### Seaborn

Seaborn là thư viện mở rộng được viết trên nền Matplotlib, từ đó kế thừa các chức năng biểu diễn dữ liệu từ Matplotlib. Vì vậy, những gì Seaborn làm được thì Matplotlib cũng làm được tuy nhiên không có chiều ngược lại, cũng có nghĩa là Seaborn không thể thay thế hoàn toàn được Matplotlib (Seaborn hiểu đơn giản là 1 đứa con được Matplotlib tạo ra).

* Seaborn là công cụ trực quan hóa dữ liệu thống kê rất mạnh
* Cung cấp nhiều style mặc định và bảng màu đẹp, giúp dữ liệu thống kê trở nên rõ ràng và cuốn hút
* Được xây dựng trên nền tảng thư viện Matplotlib
* Được tích hợp chặt chẽ với các cấu trúc dữ liệu trong thư viện Pandas
* Tham khảo: <https://seaborn.pydata.org/examples/index.html>

### Scikit-learn

* Là thư viện hỗ trợ trong quá trình phát triển các mô hình học máy, nó cung cấp một loạt các công cụ và thuật toán cho nhiều tác vụ khác nhau, từ phân loại và hồi quy tuyến tính đến clusterring.
* Các công cụ đơn giản và hiệu quả để phân tích dữ liệu dự đoán
* Mọi người đều có thể truy cập và có thể sử dụng lại trong nhiều bối cảnh khác nhau
* Được xây dựng trên NumPy, SciPy và Matplotlib
* Mã nguồn mở, có thể sử dụng về mặt thương mại - Giấy phép BSD
* Tham khảo: <https://scikit-learn.org/>

## Các bước thực hiện đề tài

### Thu thập dữ liệu (Data collection)

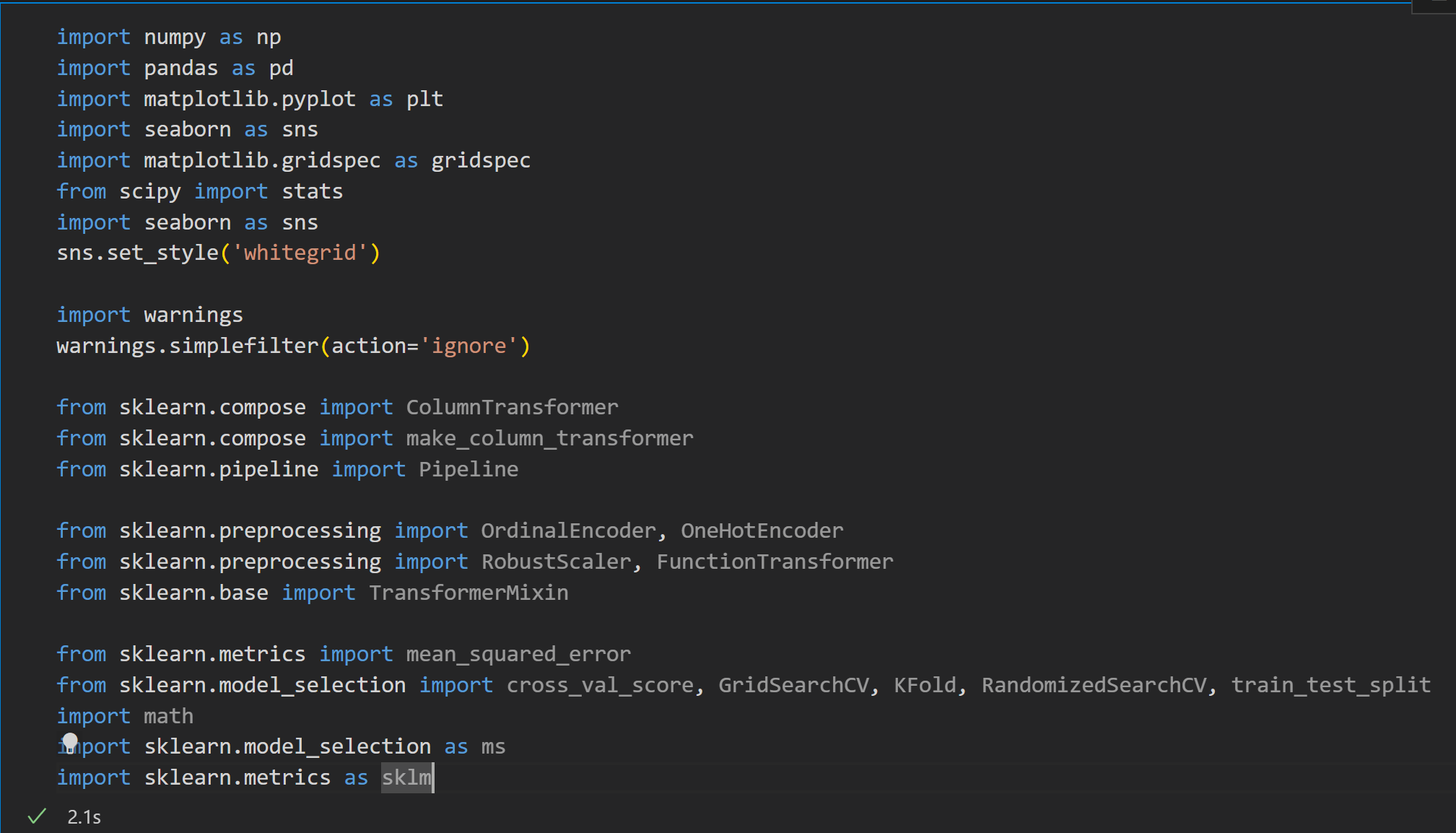
Bước đầu tiên trong quy trình Khoa học dữ liệu là thu thập dữ liệu. Dữ liệu có thể có sẵn từ trước, vừa mới được thu thập hoặc có thể dễ dàng truy cập trên Internet. Và đôi khi dữ liệu chúng ta cần không chỉ đến từ một nguồn duy nhất mà là sự kết hợp, tổng hợp từ nhiều nguồn khác nhau, ví dụ từ cơ sở dữ liệu nội bộ hoặc bên ngoài, phần mềm CRM của công ty, nhật ký máy chủ web, mạng xã hội hoặc mua dữ liệu từ các nguồn bên thứ ba đáng tin cậy. Nhiệm vụ của các nhà khoa học dữ liệu sau khi xác định được vấn đề cần giải quyết là tìm kiếm, thu thập, và trích xuất những dữ liệu phù hợp.

Có nhiều cách để chúng ta thu thập dữ liệu tuy nhiên thì trong dự án này ta sẽ lấy dữ diệu từ các nguồn dữ liệu mở cụ thể là bộ dữ liệu trên Kaggle.Thông tin chi tiết chúng em đã trình bày rất rõ ràng ở mục 1.3 và 2.1.5.

### Làm sạch và tiền xử lý dữ liệu (Data cleaning and Data processing)

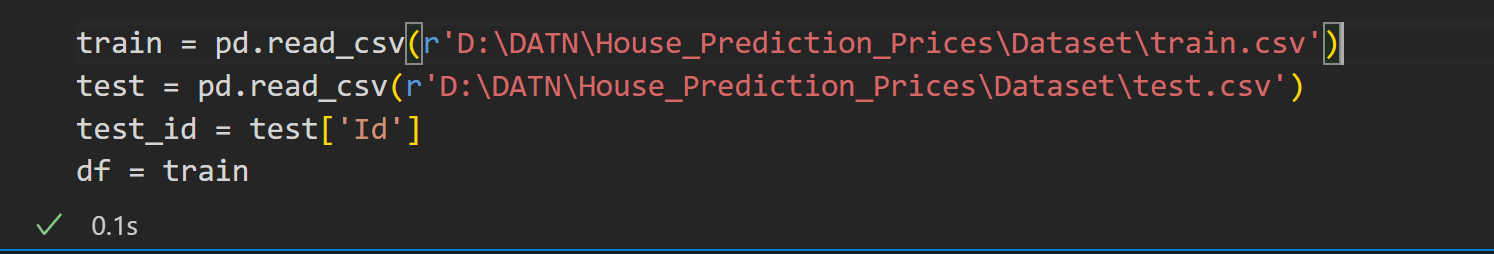
Những dữ liệu chúng ta thu thập được trong bước đầu tiên thường được gọi là dữ liệu thô. Dữ liệu thô thường không đáp ứng được các yêu cầu của các thuật toán được sử dụng trong phân tích và học máy, ví dụ như dữ liệu bị thiếu, dữ liệu bị lặp lại, dữ liệu không đúng định dạng, dữ liệu không đồng nhất, v.v. Do đó, bước tiếp theo trong quy trình ta cần thức hiện, đó là làm sạch dữ liệu. Đây là bước quan trọng nhất trong quy trình Khoa học dữ liệu giúp chúng ta chuẩn hoá dữ liệu theo một định dạng được định trước để chúng ta dễ dàng khám phá và tìm ra những hành vi ẩn thú vị có thể giúp ích cho doanh nghiệp.

* Import các thư viện cần thiết

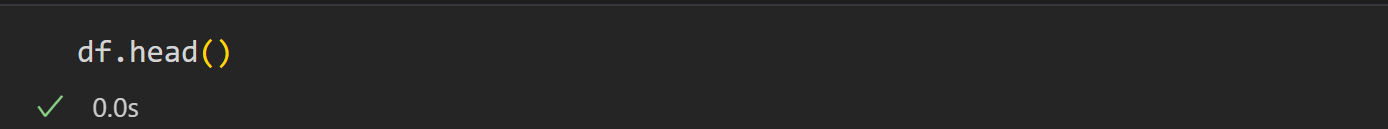


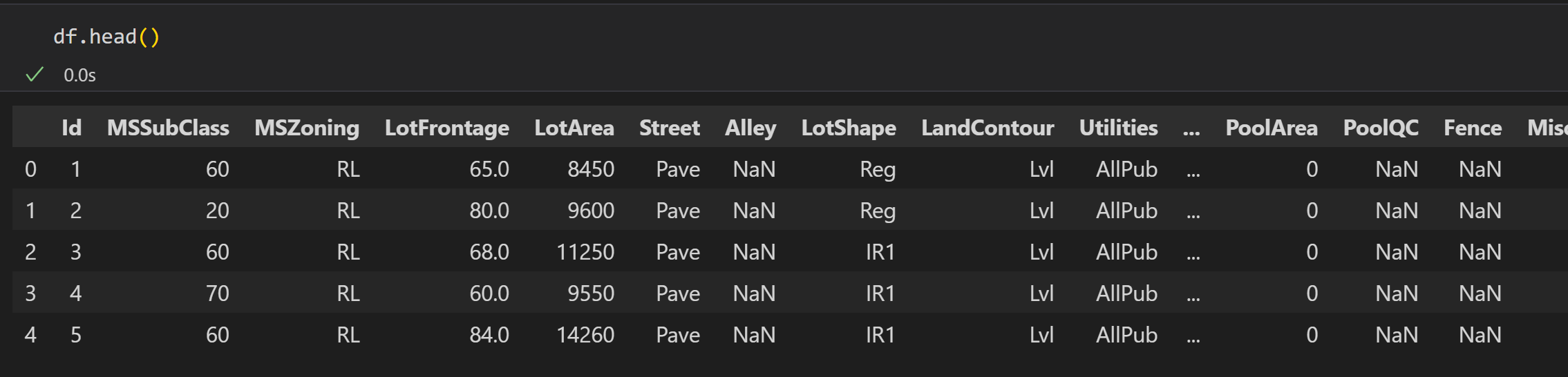
Các thư viện hỗ trợ

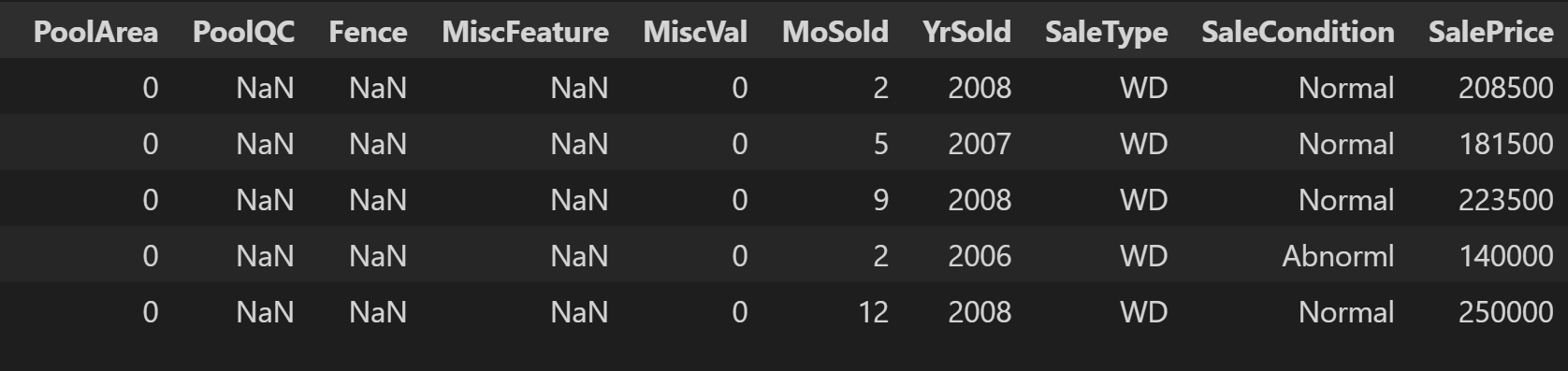
* Đọc dữ liệu từ file csv



* Lấy ra 5 dòng đầu trong DataFrame

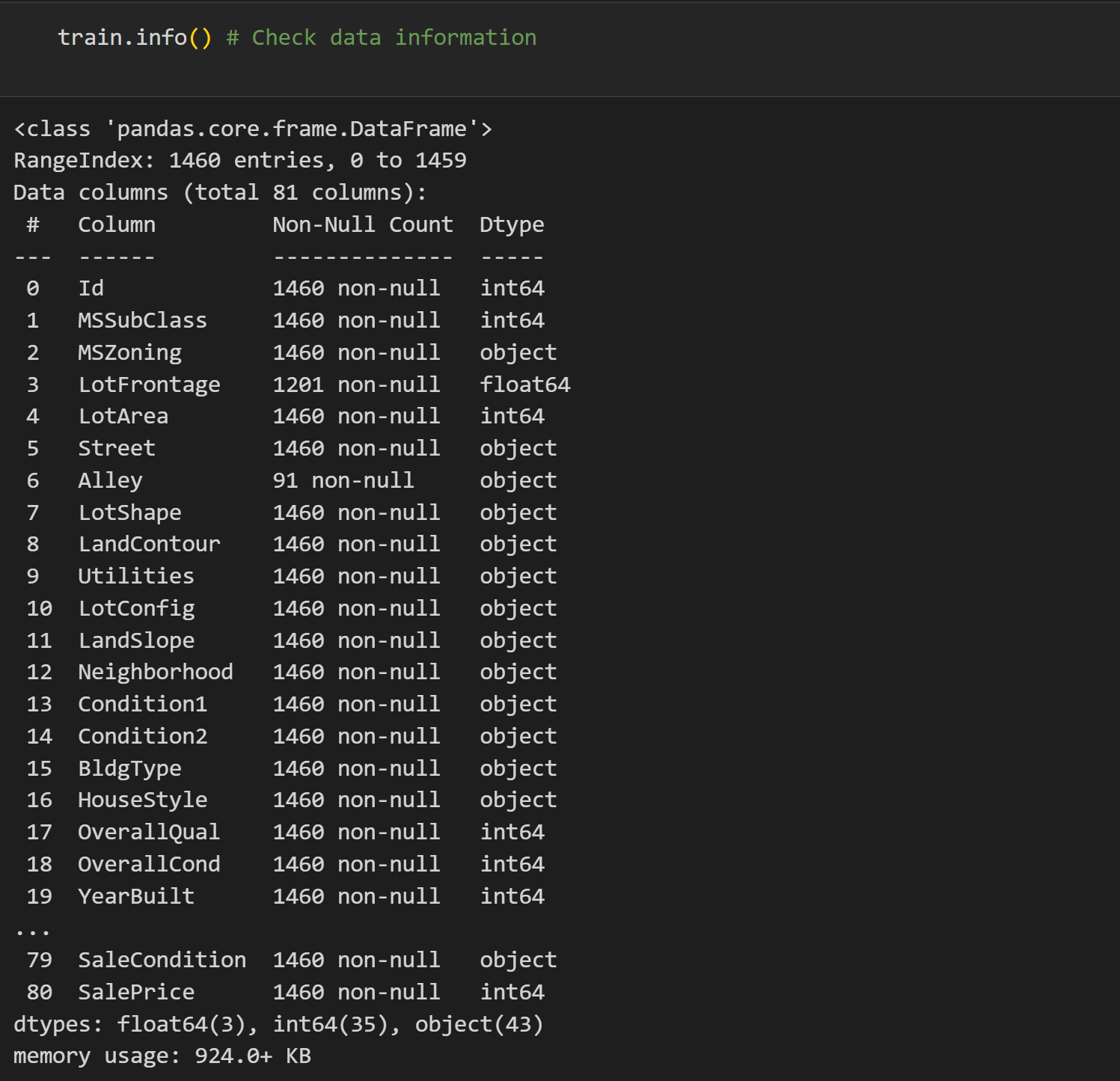






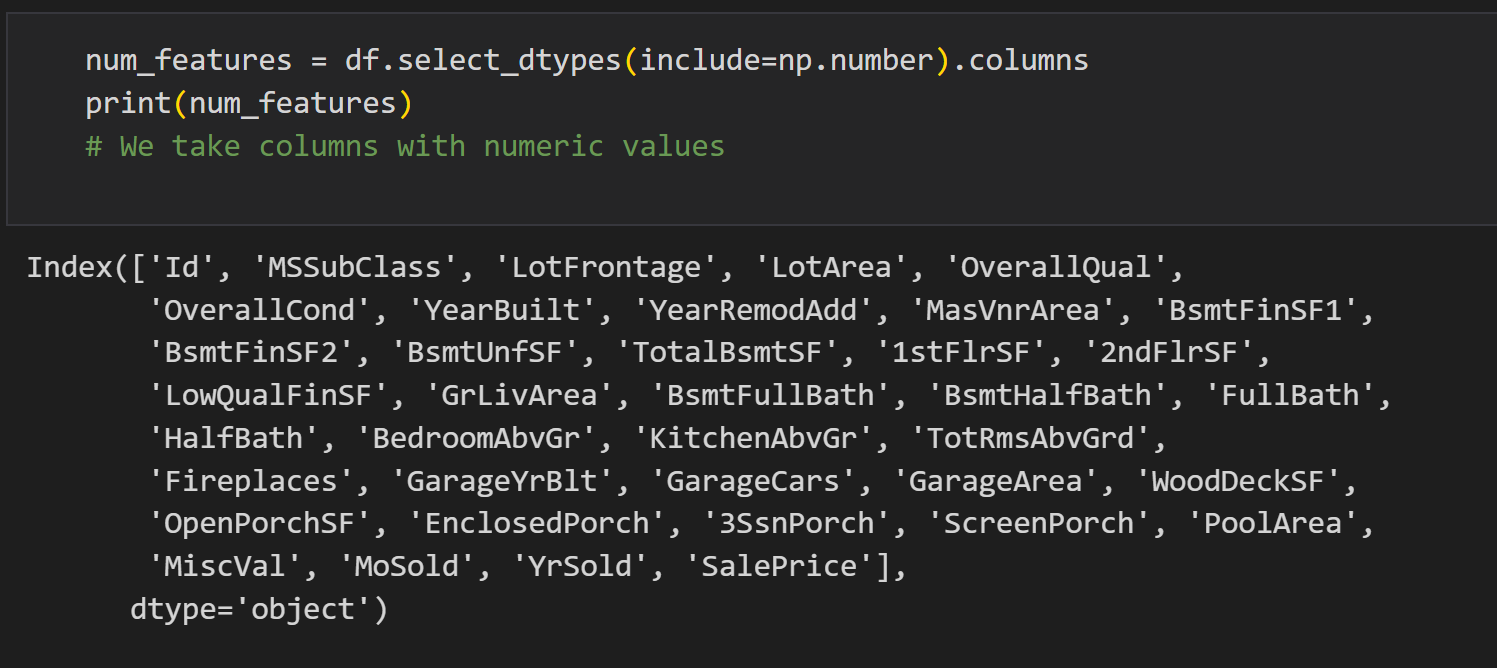
Đọc dữ liệu từ file csv

* Tiếp theo chúng ta sẽ xem thông tin của tập dữ liệu train và test

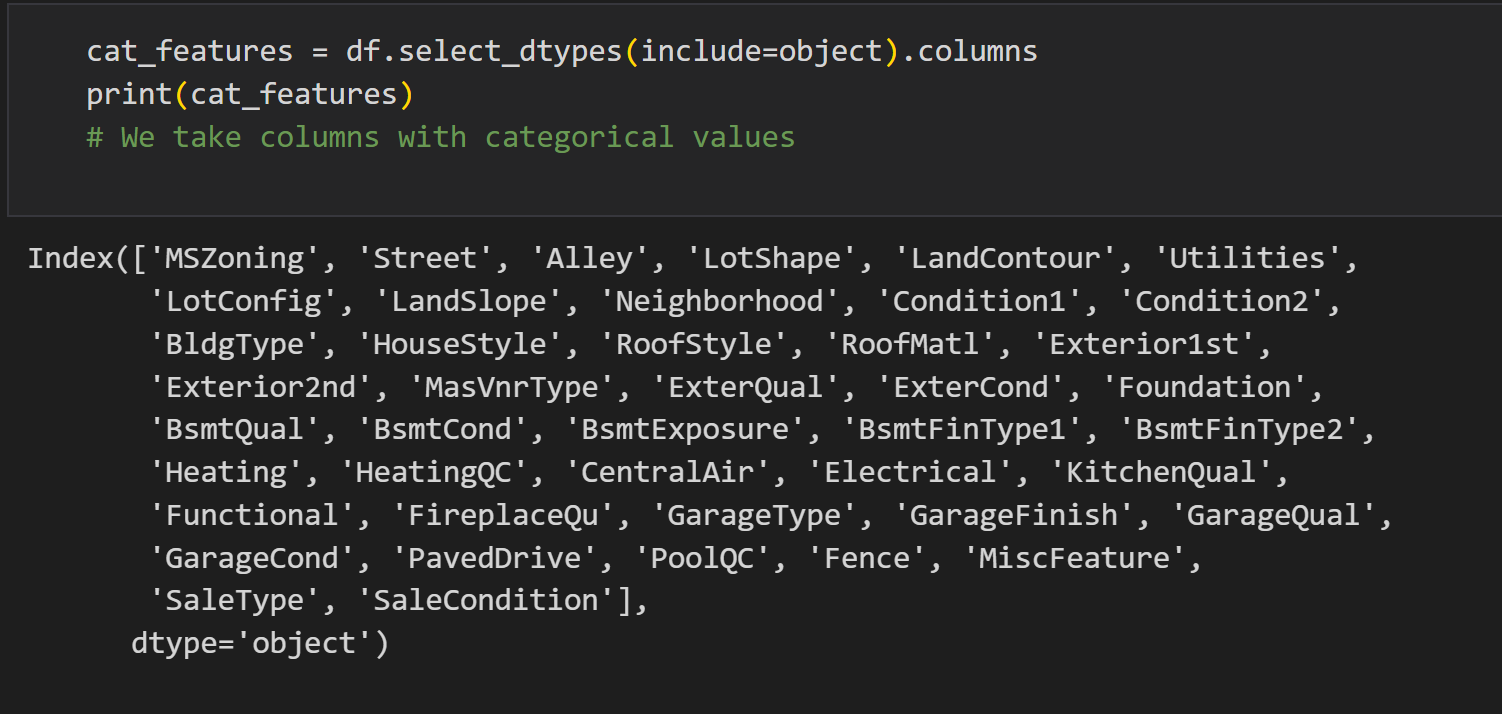


Chi tiết bộ dữ liệu huấn luyện

* Ta thấy tập dữ liệu ở đây có tổng 1460 dòng dữ liệu và 81 cột đại diện cho các giá trị cũng như các trường của dữ liệu. Tại cột Dtype cho ta biết kiểu dữ liệu của các trường, những trường có kiểu int64 hoặc float64 thuộc kiểu dữ liệu dạng số (Numerical) còn lại là dữ liệu dạng object ( Categorical).
* Chúng ta sẽ lấy ra thông tin của những trường có dữ liệu dạng numerical và categorical



Dự liệu dạng Numerical

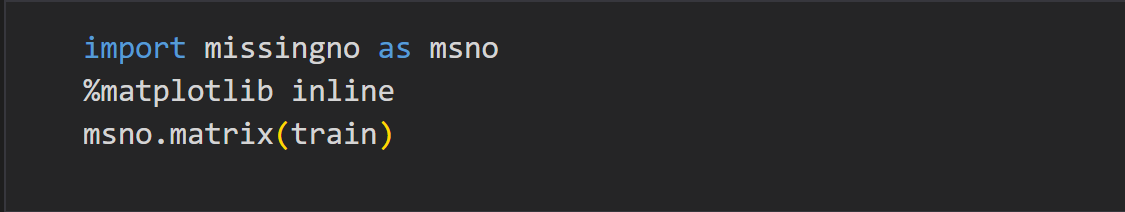


Dữ liệu dạng Categorical

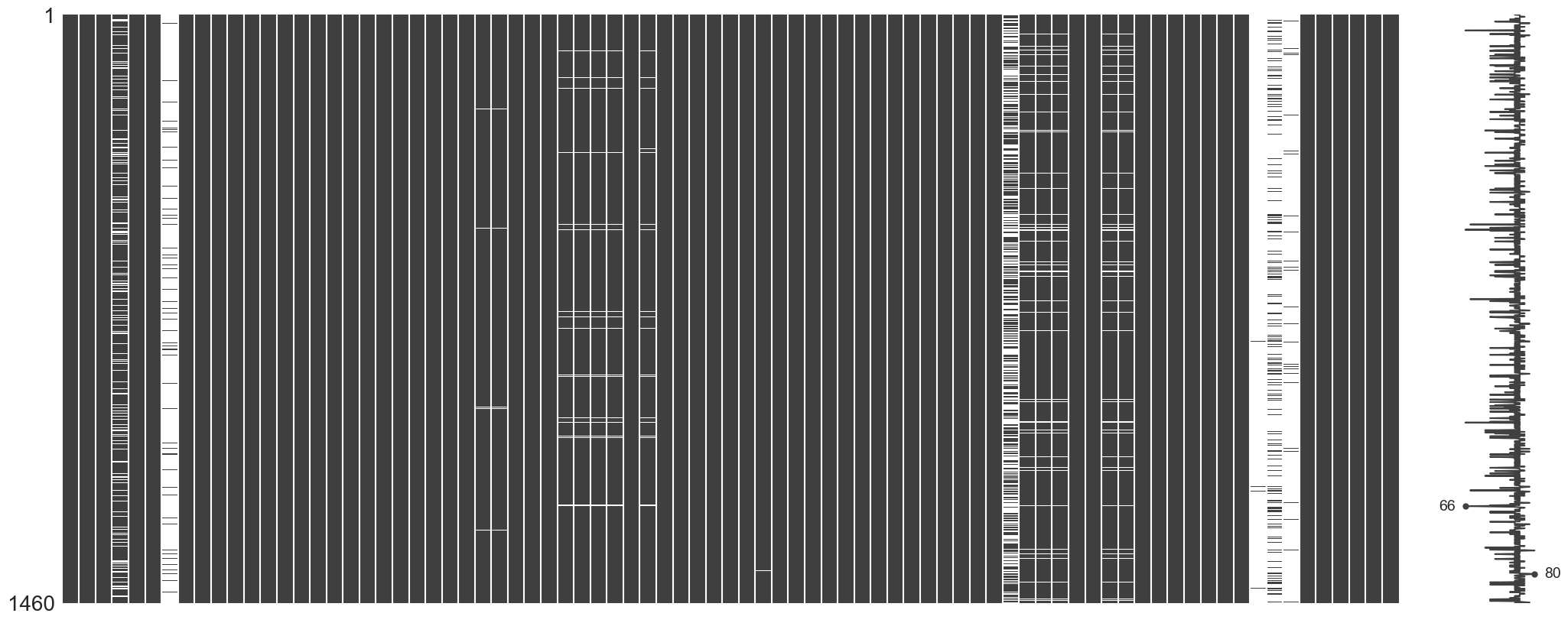
* Như chúng ta thấy dữ liệu khi thu về thường sẽ là dữ liệu thô vì thế việc thiếu sót cũng như sai xót là điều không thể tránh khỏi và việc làm của chúng ta là xử lý chúng.

Dữ liệu thiếu là tình trạng mà một hoặc nhiều giá trị trong tập dữ liệu bị khuyết hoặc không có. Việc xử lý dữ liệu thiếu là một bước quan trọng để đảm bảo rằng dữ liệu được chuẩn bị một cách đầy đủ và chính xác trước khi đưa vào mô hình học máy. Dưới đây là một số bước cụ thể để xử lý dữ liệu thiếu:

* Đánh giá tỷ lệ dữ liệu thiếu: Trước tiên, cần kiểm tra tỷ lệ dữ liệu thiếu trong từng biến. Điều này có thể được thực hiện bằng cách tính toán tỷ lệ phần trăm các giá trị thiếu trong mỗi biến.
* Chúng ta có một thư viện rất mạnh mẽ trong việc mô tả dữ liệu bị thiếu theo dạng ma trận đó là thư viện missingno

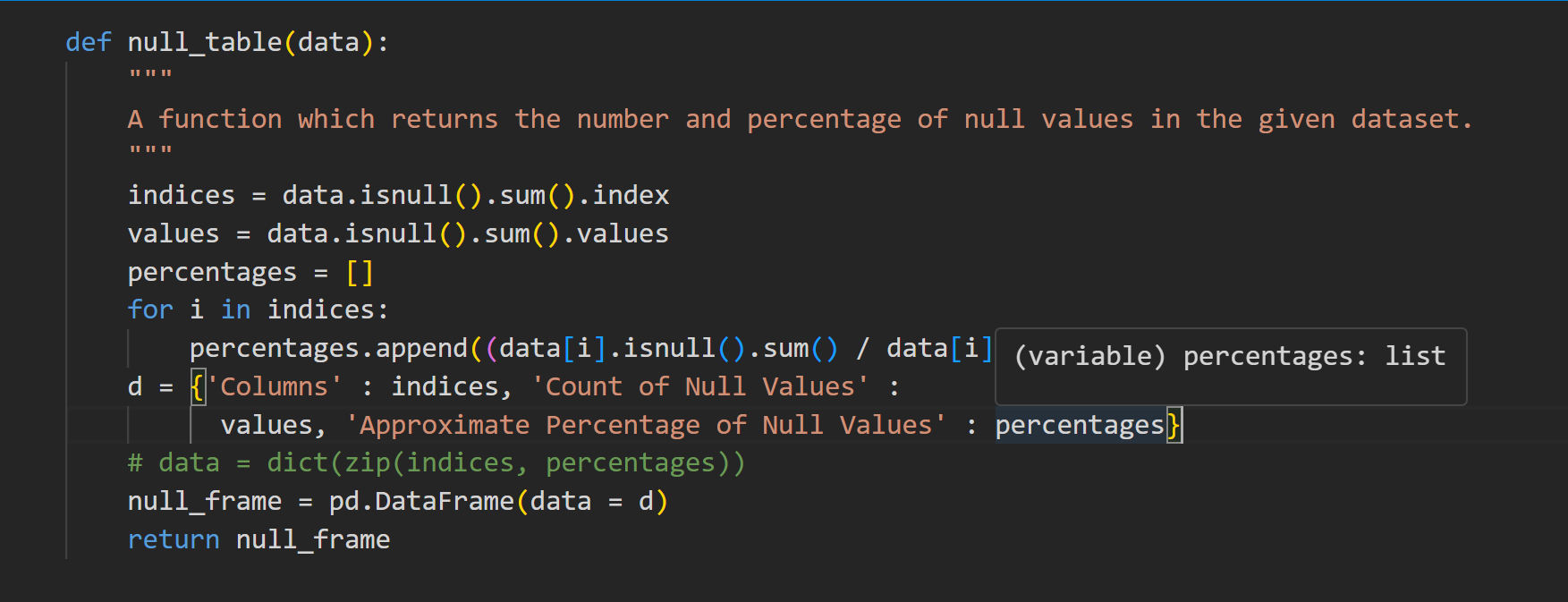


* Hãy cũng xem kết quả khi chạy câu lệnh trên

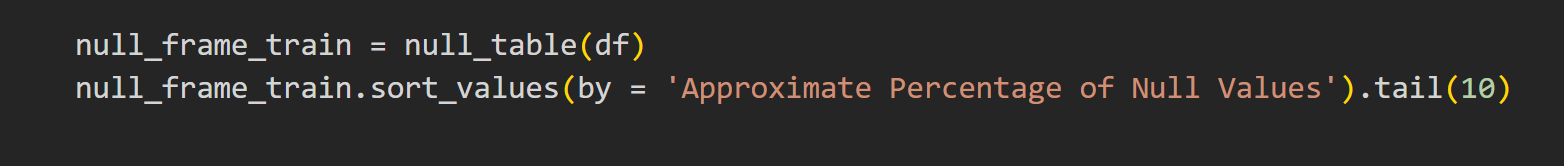


Missing value tập huấn luyện

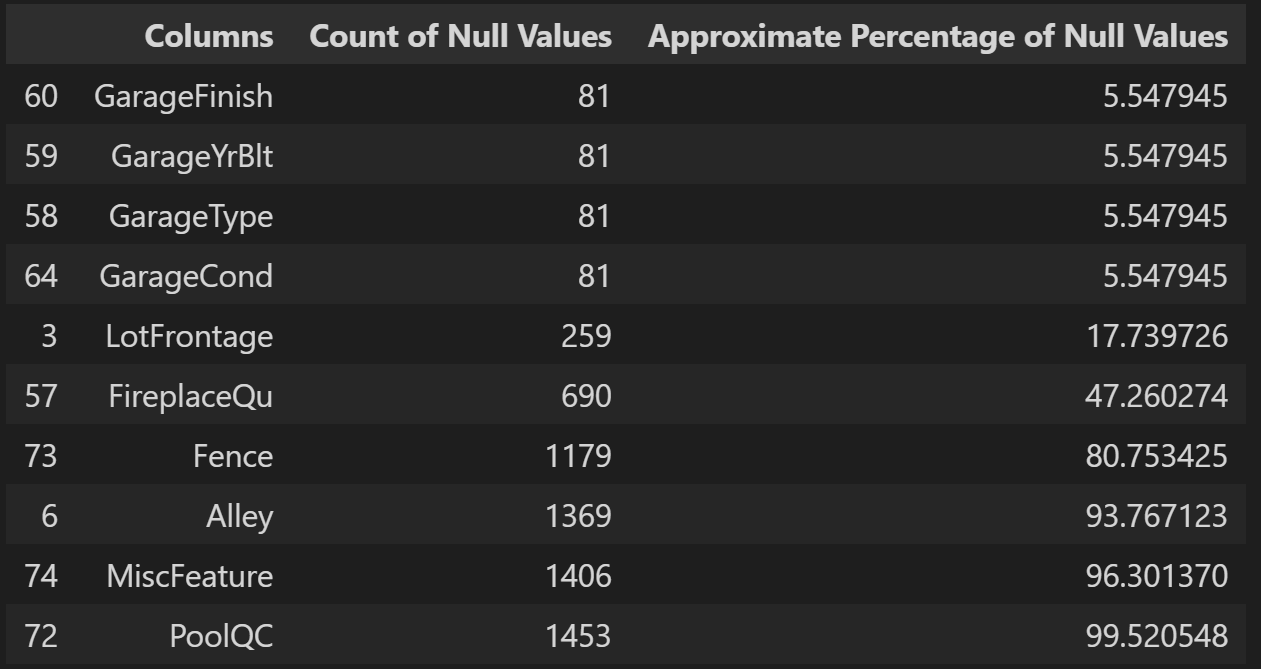
* **Nhận xét:** Ta thấy bộ dữ liệu ở tập train bị thiếu rất nhiều dữ liệu điều này cũng là điều dễ hiểu vì trong quá trình thu thập không thể chắc chắn sẽ đầy đủ thông tin. Hãy xem thống kê các trường có tỉ lệ mất mát dữ liệu lớn nhất để tìm cách xử lý



* Hàm thống kê dữ liệu bị thiếu

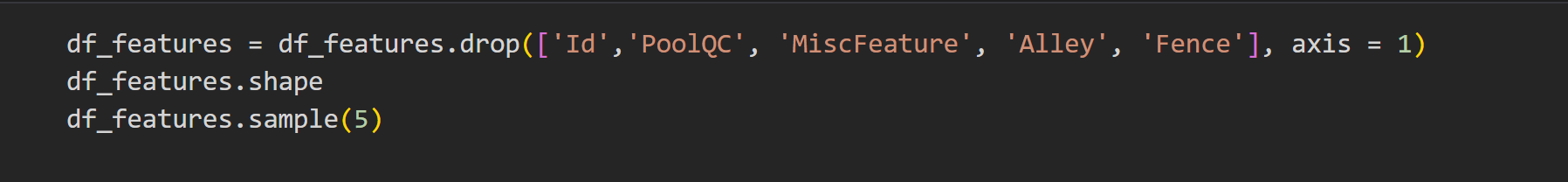


* Kết quả thống kê các số lượng và phần trăm mất mát dữ liệu của các trường trong dataset

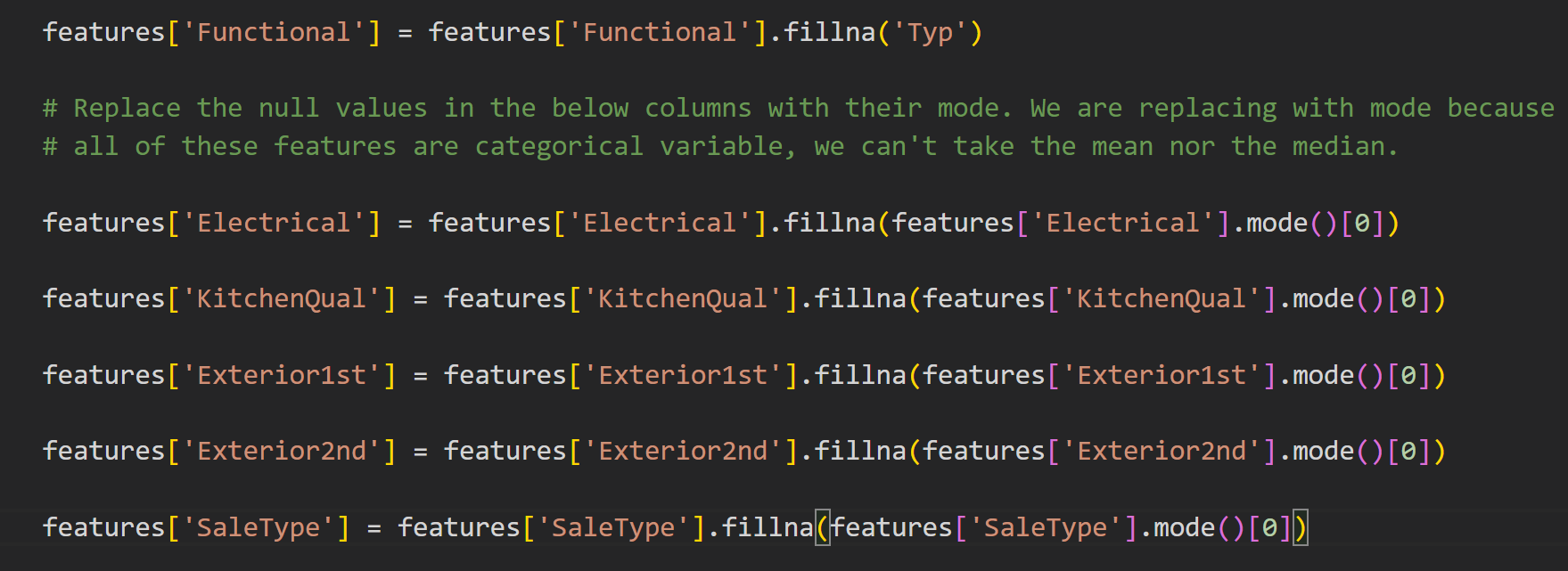


Bảng thống kê vissing value tập huấn luyện

* Ta thấy rằng 5 trường cuối là những trường có tỉ lệ mất mát dữ liệu nhiều nhất ta có thể xem xét bỏ những trường này.Những cột cồn lại có thể sử dụng một vài phương pháp như:
* Điền giá trị trung bình/median/most frequent
* Sử dụng giá trị trung bình, trung vị hoặc mode của biến để điền vào các giá trị thiếu.
* Cách xử lý ở đây chúng ta sẽ xem xét xóa bỏ 5 trường có tỉ lệ missing value cao và một vài cột không ảnh hưởng nhiều đến giá trị bài toán như cột Id,…

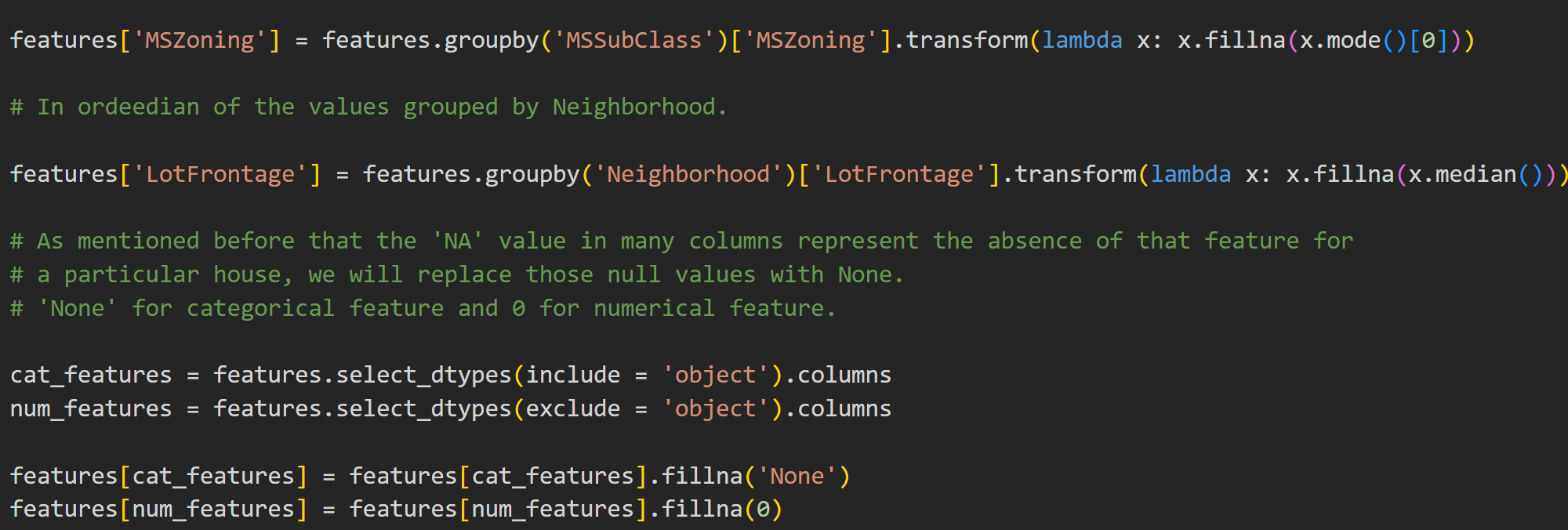


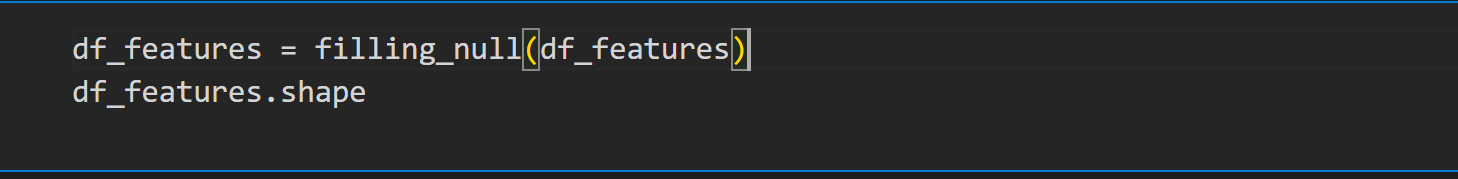
* Đối với các trường có kiểu dữ liệu dạng Categorical chúng ta sẽ lựa chọn điền tự động với phương pháp mode() tức là những giá trị nào xuất hiện nhiều nhất thì sẽ được ưu tiên



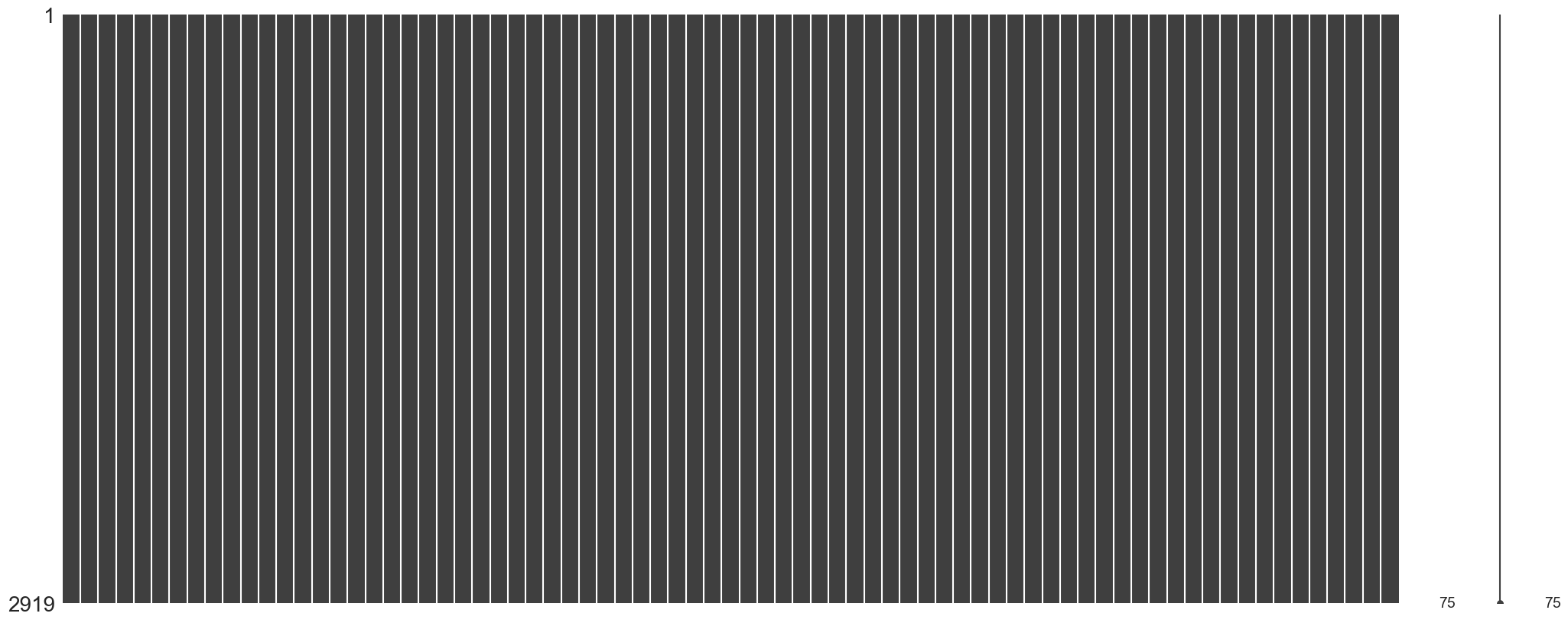


* Các trường dữ liệu dạng Numerical thì sẽ điền giá trị null dựa trên giá trị trung bình, trung vị (median)





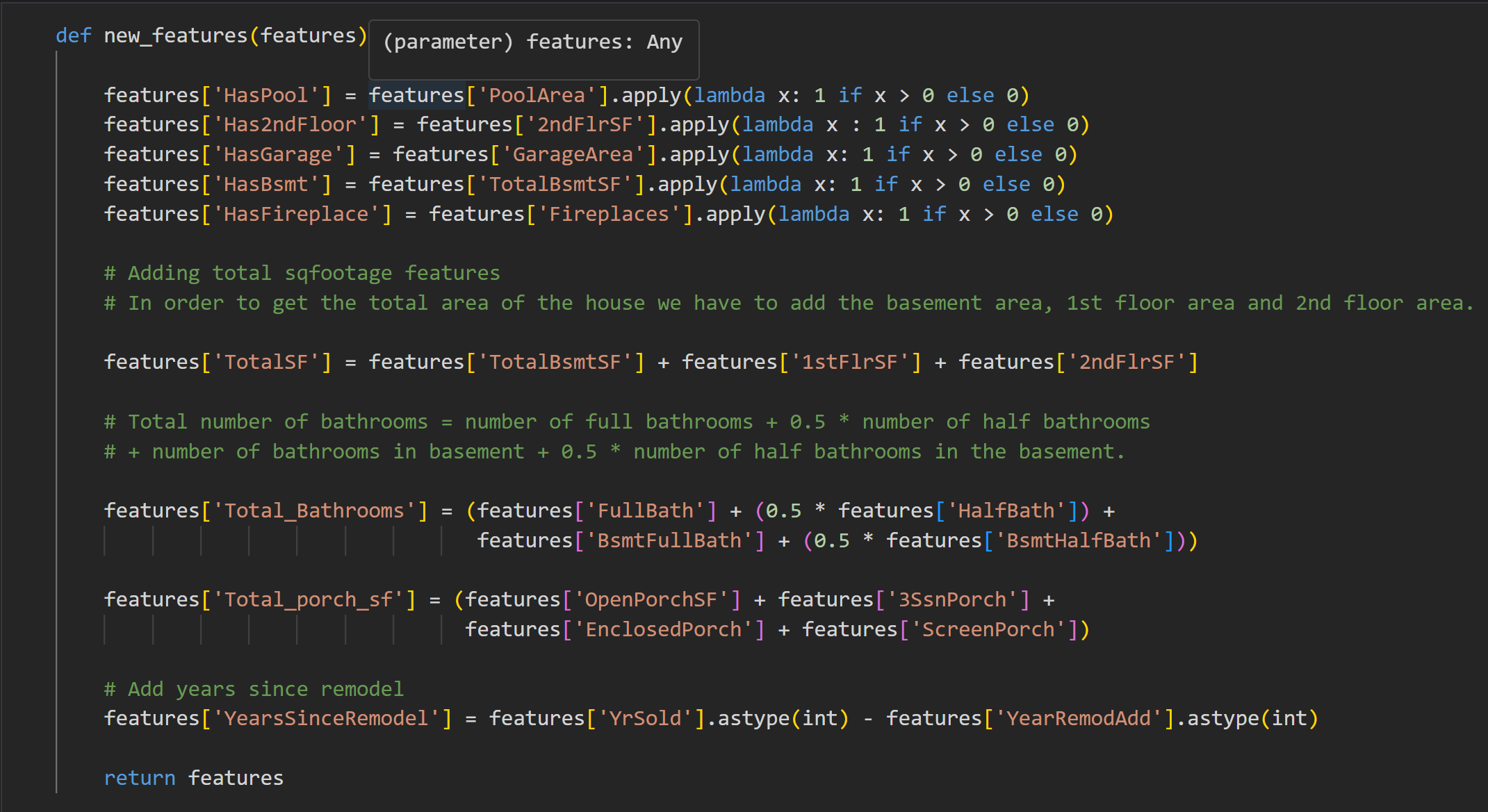
* Kết quả sau khi chúng ta xử lý dữ liệu bị thiếu, ta thấy bộ dữ liệu đã hoàn chỉnh hơn



Kết quả xử lý missing value

Tiếp theo chúng ta sẽ xem xét đến việc chuyển đổi dữ liệu , việc này nhằm mục đích nâng cao chất lượng của mô hình máy học khi chúng ta huấn luyện và cũng là công việc rất cần thiết.

* Dưới đây là hàm dùng để chuyển đổi dữ liệu

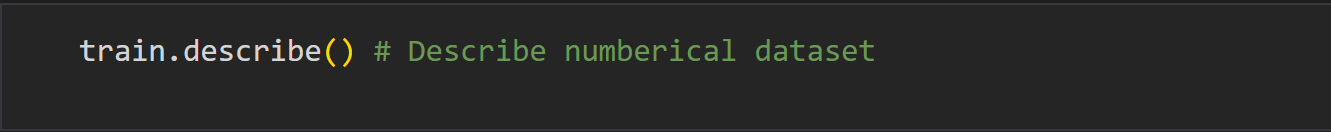


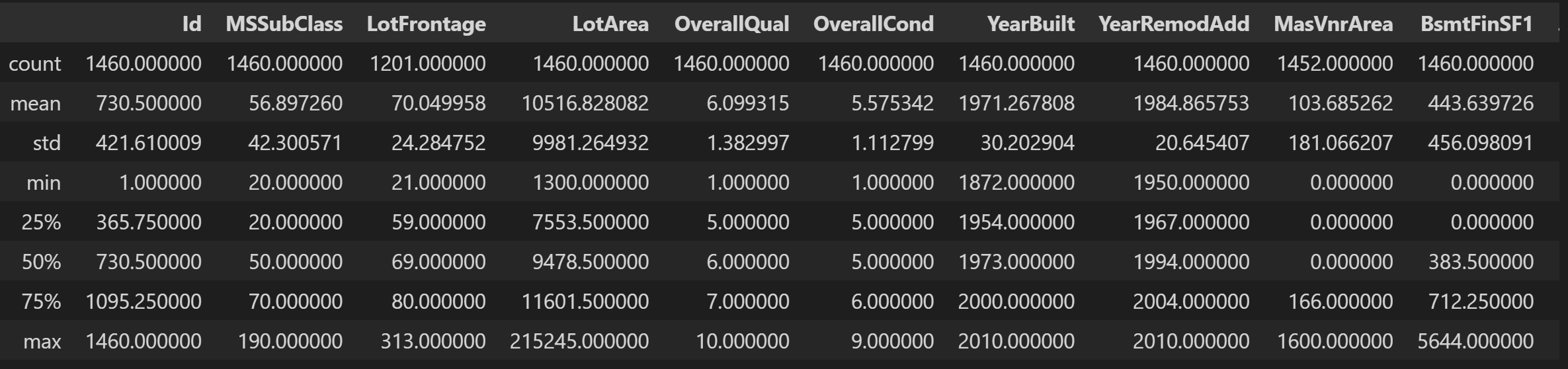
* Trên đây là những phương pháp để làm sạch và tiền sử lý dữ liệu trước khi xây dựng một mô hình AI. Tuy nhiên quá trình này không chỉ có vậy mà sẽ còn rất nhiều việc mà một nhà khoa học dữ liệu phải làm , tuy vào bộ dữ liệu mà chúng ta sẽ phải đưa ra nhiều phương pháp và cách xử lý khác nhau.

### Khai phá và trực quan hóa dữ liệu ( Data mining và Data visualization )

Bước tiếp theo trong quy trình Khoa học dữ liệu là khám phá dữ liệu. Đây là thao tác phân tích sơ bộ dữ liệu giúp chúng ta hiểu sâu sắc hơn về dữ liệu ta đã thu thập được, những mẫu dữ liệu nào thú vị có thể được nghiên cứu hoặc tận dụng, những thông tin nào có thể giúp ích cho doanh nghiệp, những thông tin nào có thể được sử dụng để xây dựng mô hình học máy, v.v. Từ đó, nhà Khoa học dữ liệu có thể lập kế hoạch kỹ hơn cho các chiến lược mô hình hóa dữ liệu bước tiếp theo. Các công cụ như Pandas, NumPy để phân tích dữ liệu, và Matplotlib, Seaborn, Plotly, v.v. để trực quan hóa và khám phá dữ liệu thường được sử dụng trong giai đoạn này. Tương tự như bước làm sạch dữ liệu, chúng ta sẽ đi sâu vào các phương pháp khám phá dữ liệu hiệu quả thông các công cụ phân tích và trực quan hóa dữ liệu sử dụng Python trong các phần sau.

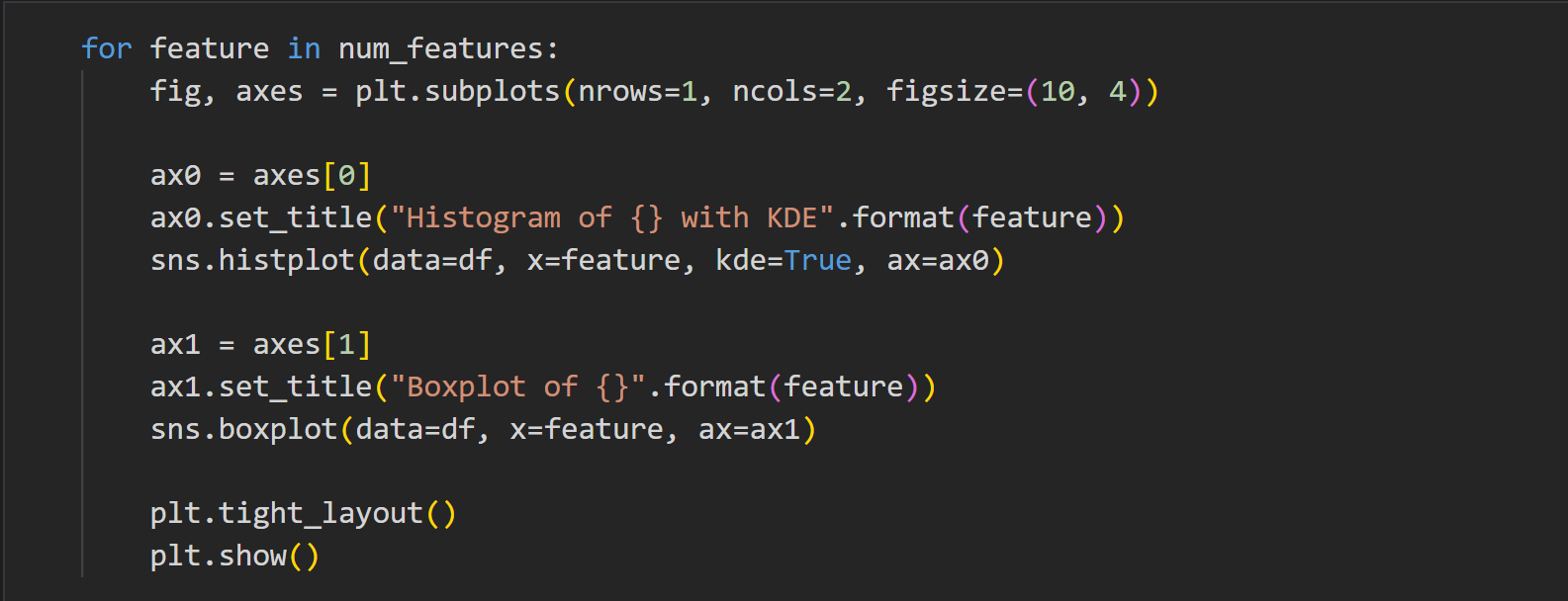
* Để bắt đầu khai phá dữ liệu chúng ta hãy đi qua thống kê mô tả của bộ dữ liệu qua hàm describe() của Pandas. Đây là hàm thống kê các trường dữ liệu dạng số , nó cho ta biết các thông tin về số lượng, giá trị trung bình, min , max, độ lệch chuẩn ,…



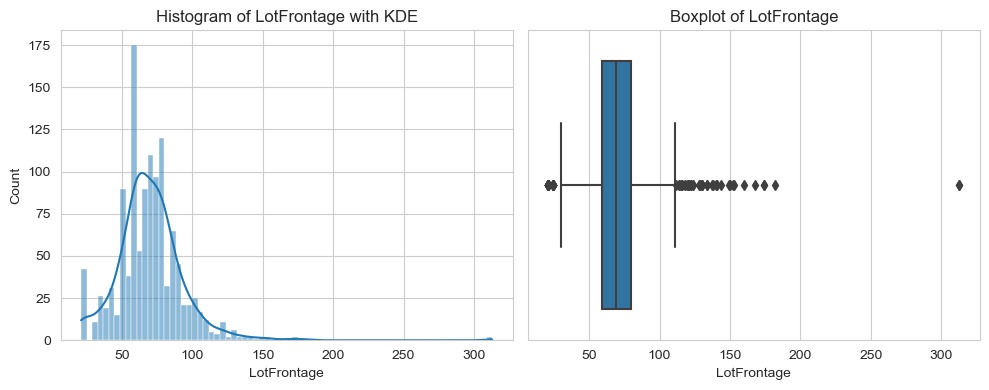


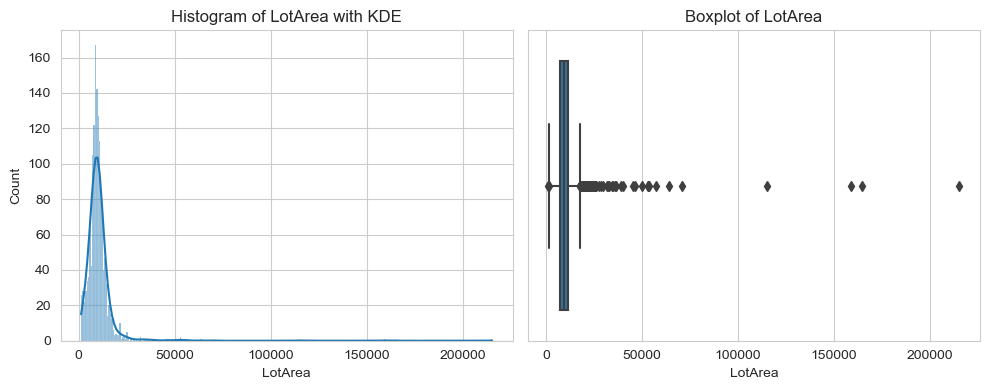
Mô tả thống kê

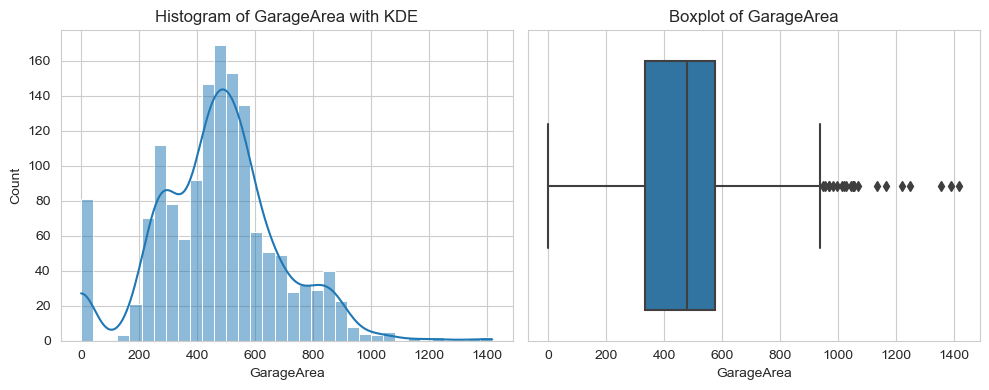
* Tiếp tục chúng ta sẽ xem sự phân phối của các trường dữ liệu để có thể hiểu hơn và có thể đưa ra đánh giá cũng như hướng giải quyết

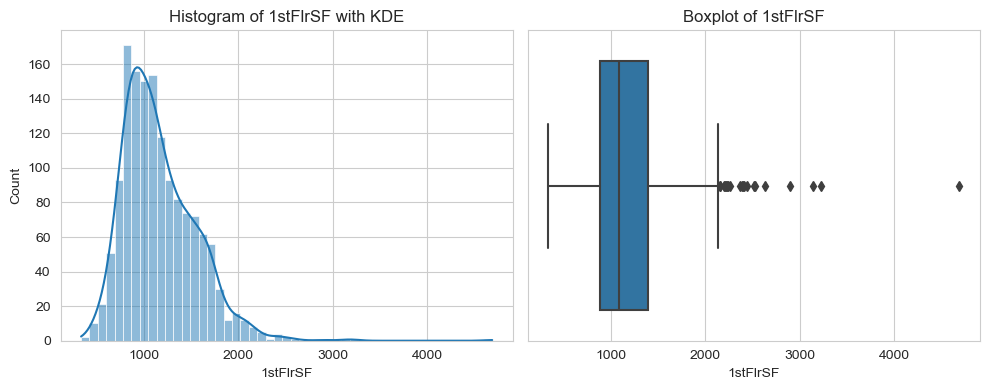


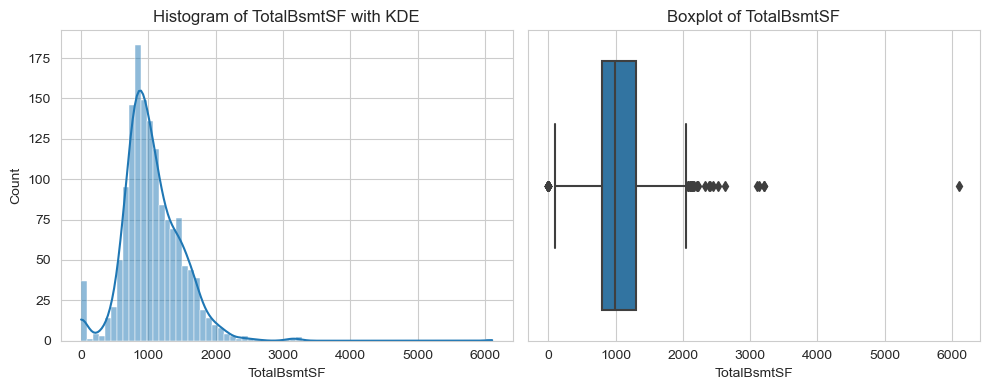
* Kết quả sau khi chạy câu lệnh

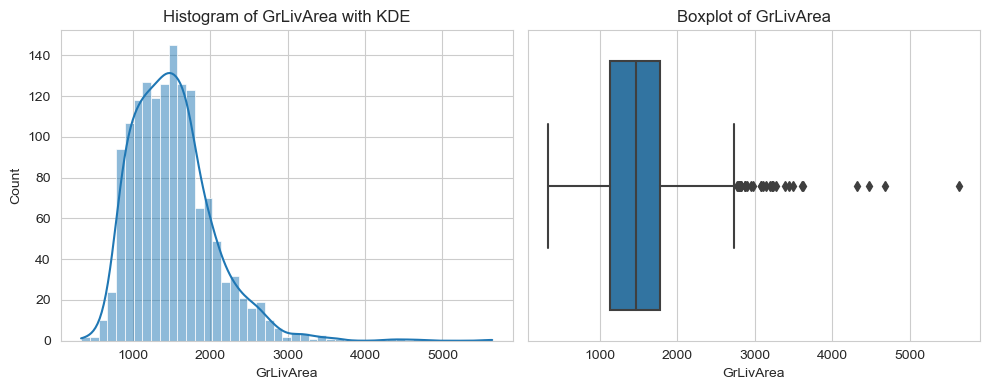




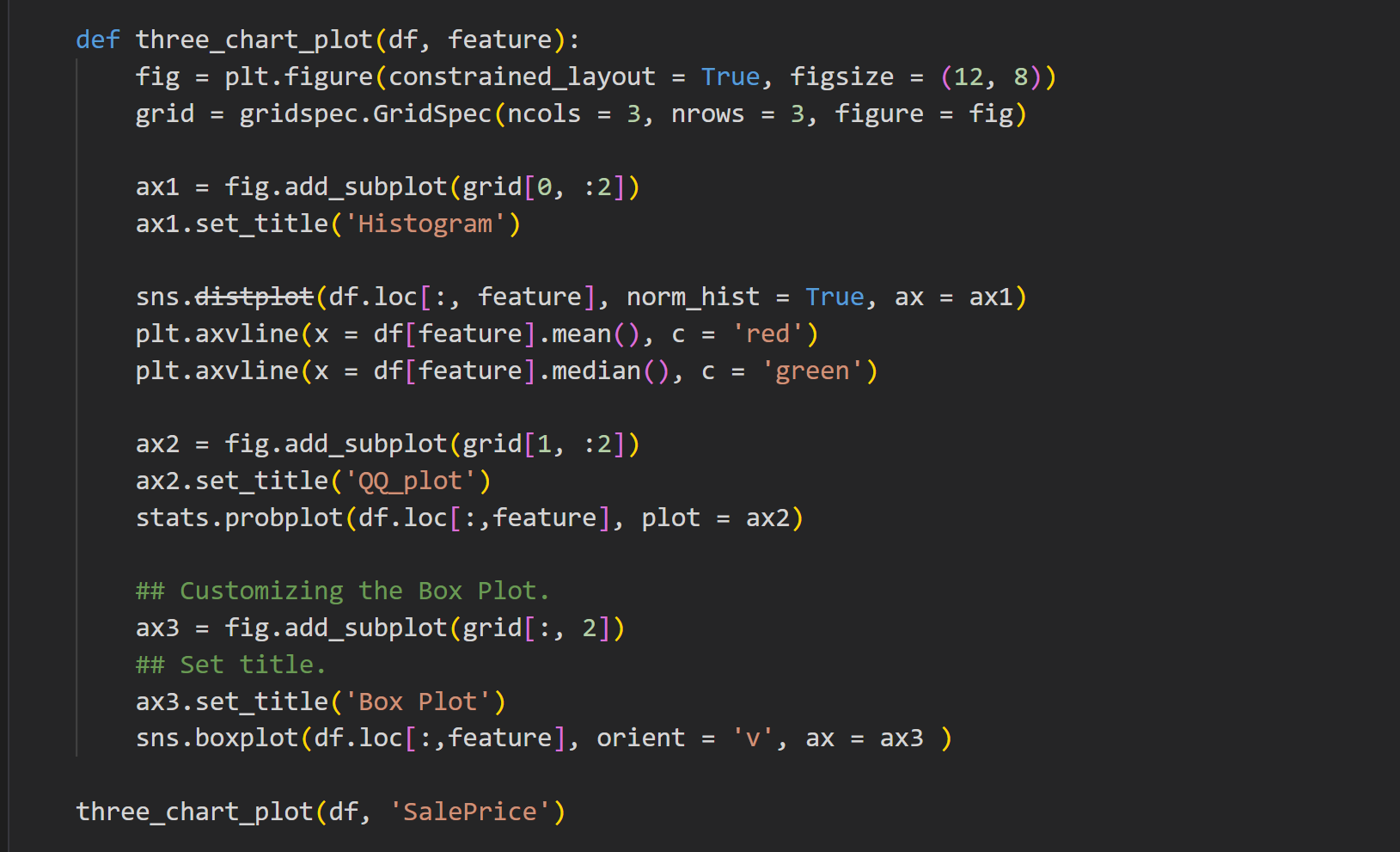




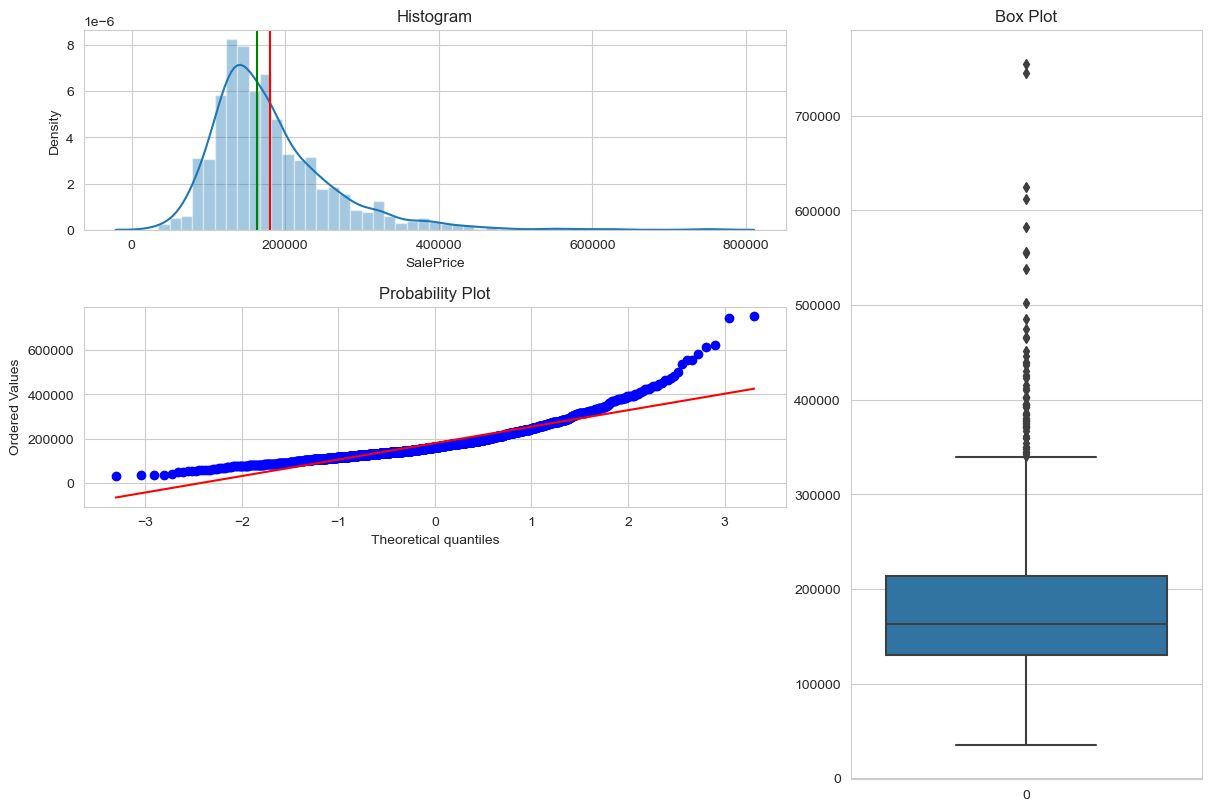




* Ở đây chúng ta thấy có vẻ như dữ liệu phân bố bị lệch về phía trái và có rất nhiều ngoại lai (outliers) và việc của chúng ta là giải quyết chúng.
* Chúng ta sẽ quan tâm biến mục tiêu của bài toán cụ thể là trường “SalePrice”. Giờ hãy xem sự phân bố dữ liệu của trường này:

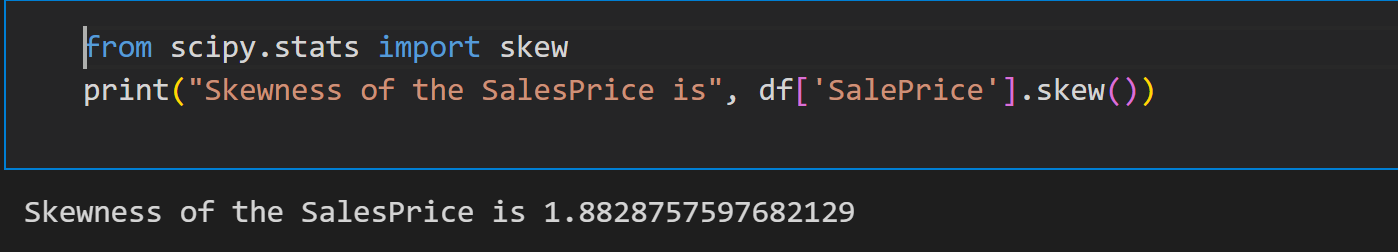


* Kết quả sau khi chạy hàm sau

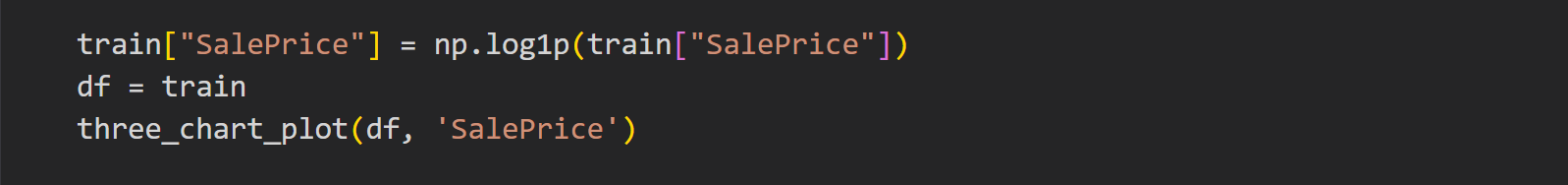


Phân phối dữ liệu của biến mục tiêu “Saleprice”

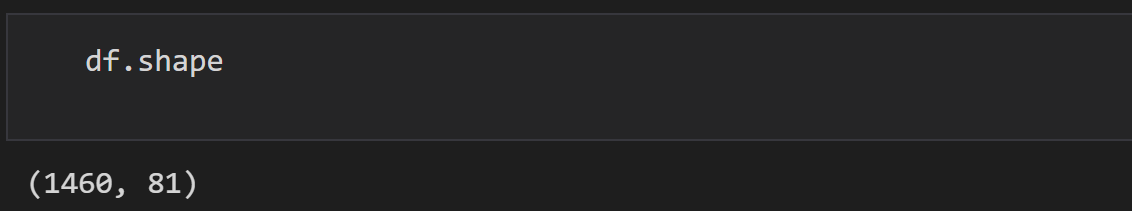
* Ta thấy sự phân bố dữ liệu của trường *“Saleprice”* không cân đối bị lệch phải qua biểu đồ histogram. Cột màu đỏ thể hiện giá trị trung bình và cột màu xanh cho thấy giá trị trung vị.
* Trong biểu đồ xác xuất probability thấy rằng đường màu đỏ là tiêu chuẩn của dữ liệu tuy nhiên đường màu xanh đại diện cho giá trị thực. Trong biểu đồ Boxplot ta cũng thấy có rất nhiều điểm nằm ngoài mức Q3 đó là những giá trị ngoại lai, chúng làm cho bộ dữ liệu của ta trở nên mất cân đối và nếu không xử lý sẽ ảnh hưởng đến độ chính xác của mô hình.
* Để thấy rõ hơn ta sẽ sử dụng hàm skew() trong thư viện Scipy của Python thống kê độ lệch của bộ dữ liệu



* Rõ ràng độ lệch là 1,88 là rất cao và bộ dữ liệu của chúng ta lệch phải rất nhiều
* Đưa dữ liệu về dạng phân phối chuẩn

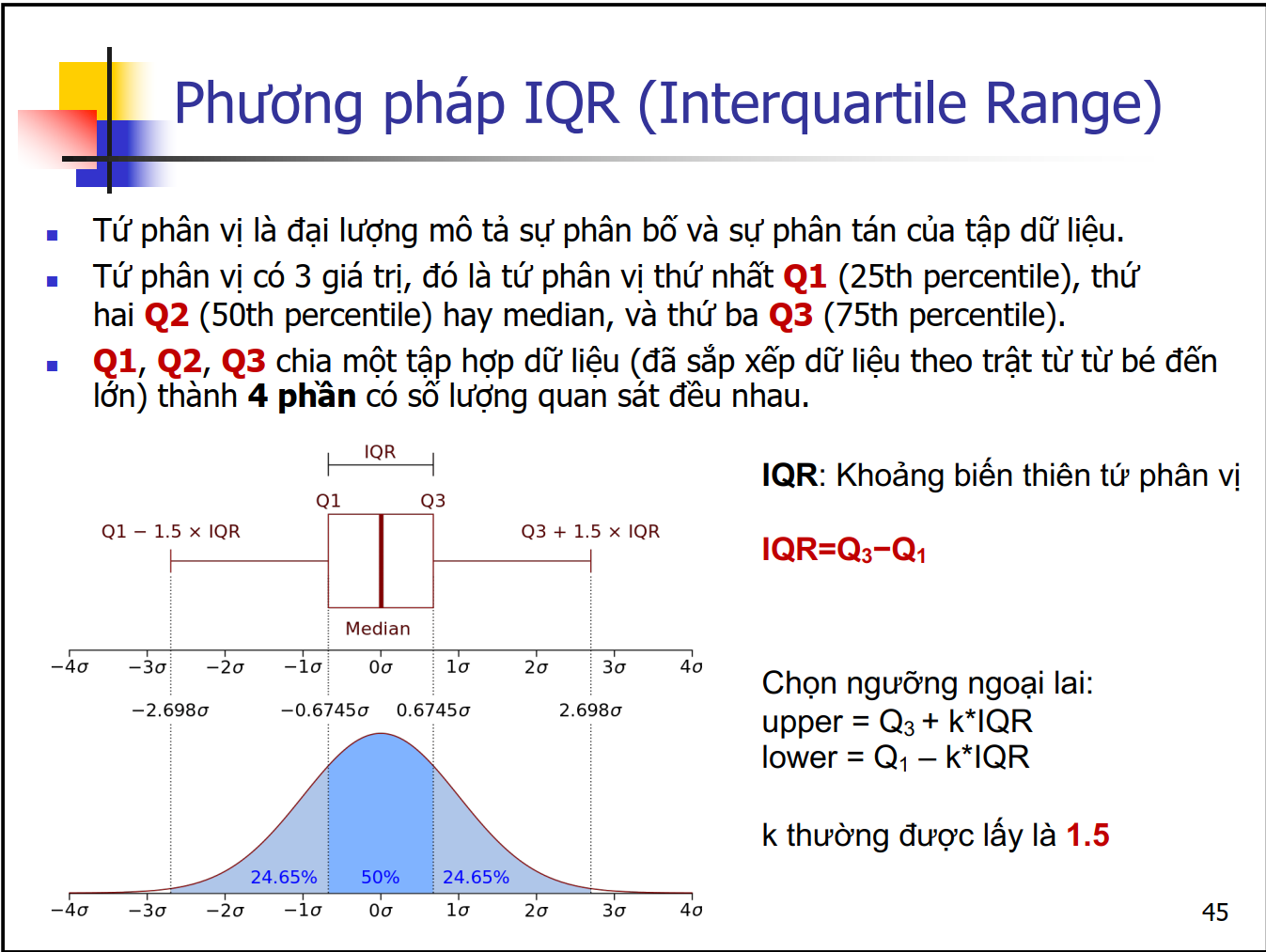


* Việc tiếp theo rất quan trọng đó là chúng ta cần xóa bỏ những ngoại lai làm ảnh hưởng tới kết quả

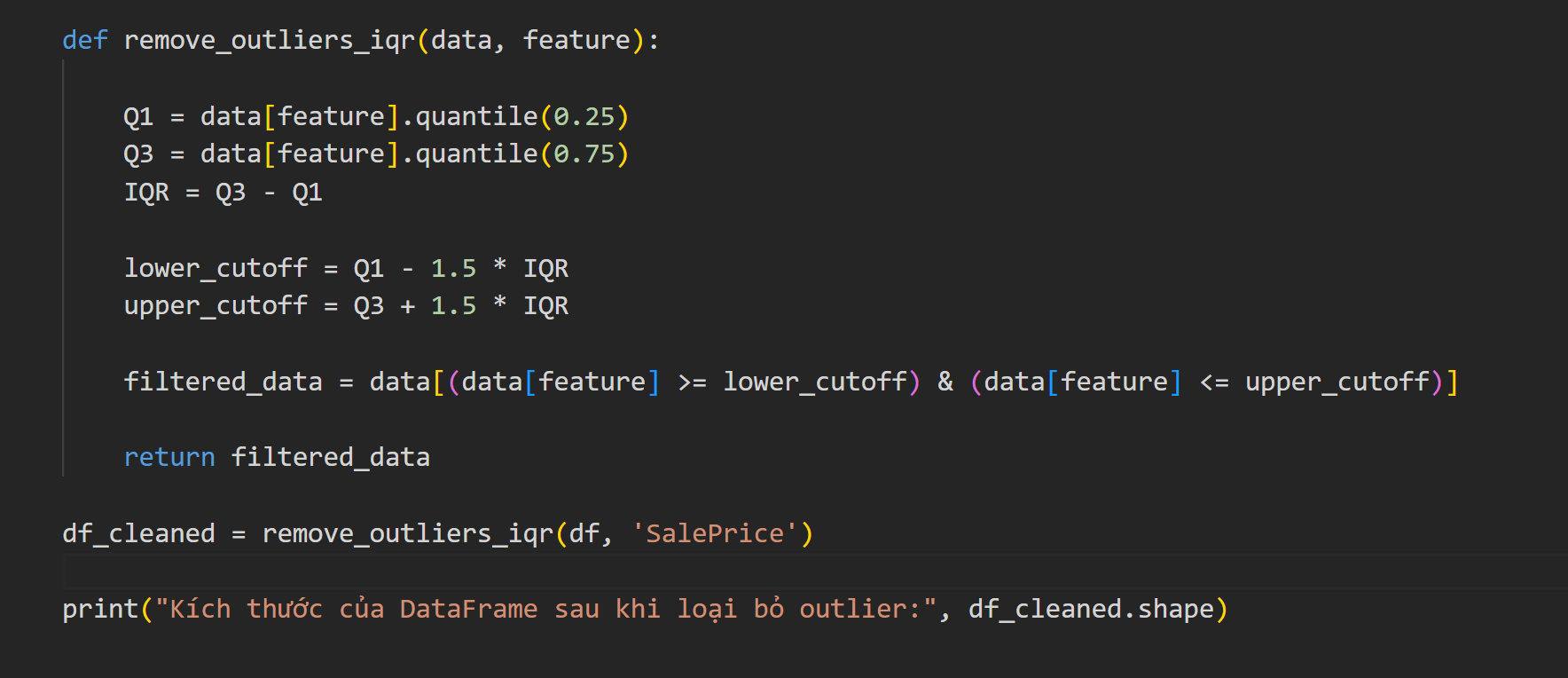


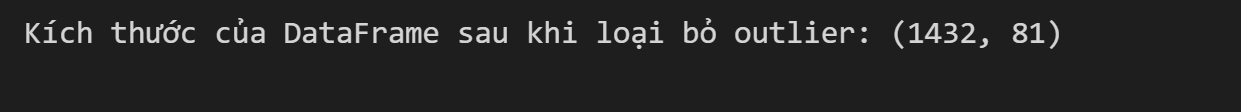
Có nhiều phương pháp xử lý ngoại lai (outliers) như:

* Phương pháp bách phân vị (Percentile)
* Phương pháp độ lệch chuẩn (std – standard deviation)
* Phương pháp Z-score
* Phương pháp tứ phân vị (IQR – interquartile range)
* Mô hình Isolation Forest
* Ở đây chúng ta sẽ lựa chọn phương pháp tứ phân vị (IQR)



Phương pháp IQR – xử lý Outliers

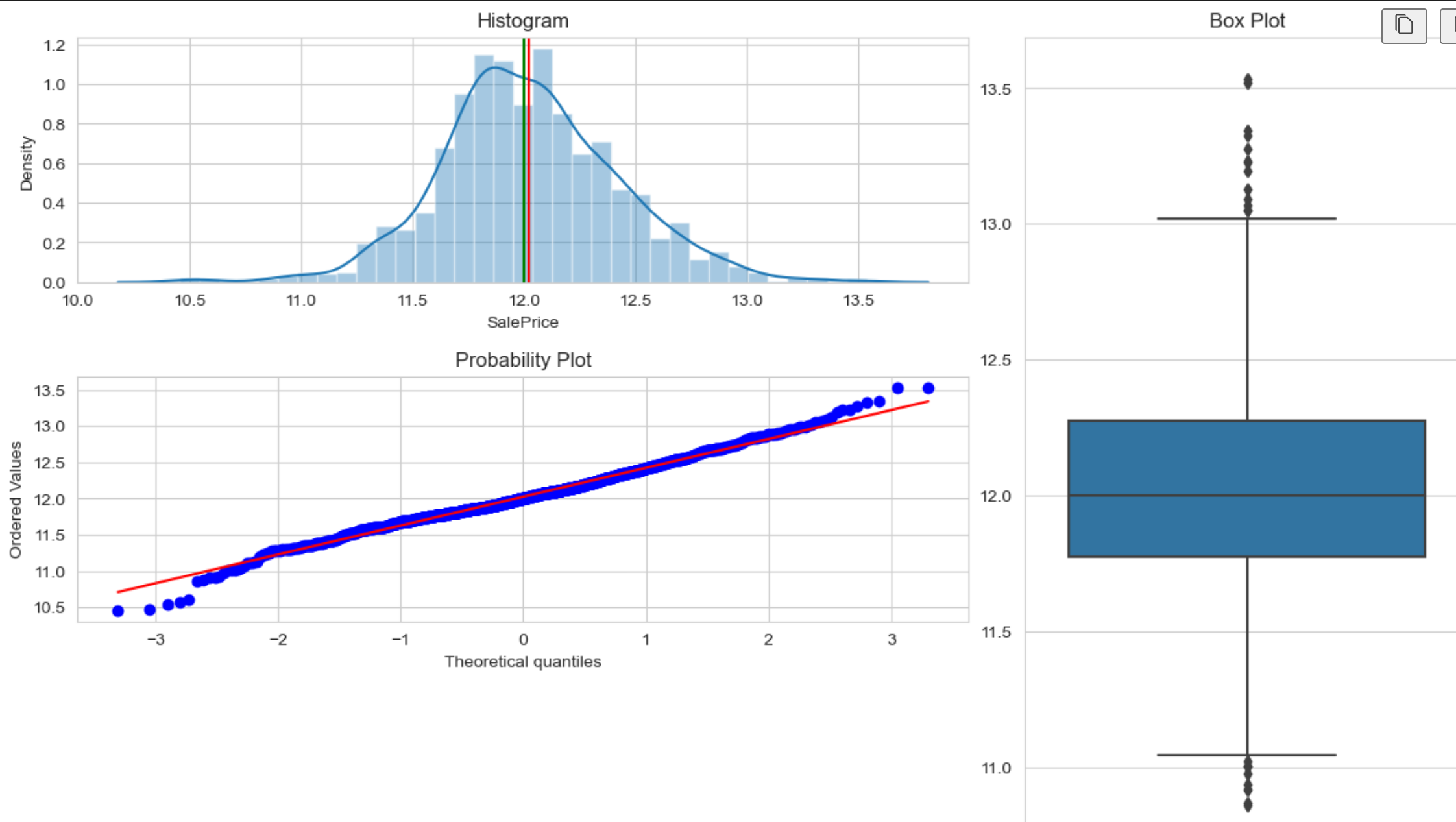




* Như vậy sau khi xử lý chúng ta đã xóa được 28 dòng có dữ liệu ngoại lai
* Tiếp theo chúng ta cần viết một hàm để xử lý độ lệch để đưa về dạng phân phối chuẩn

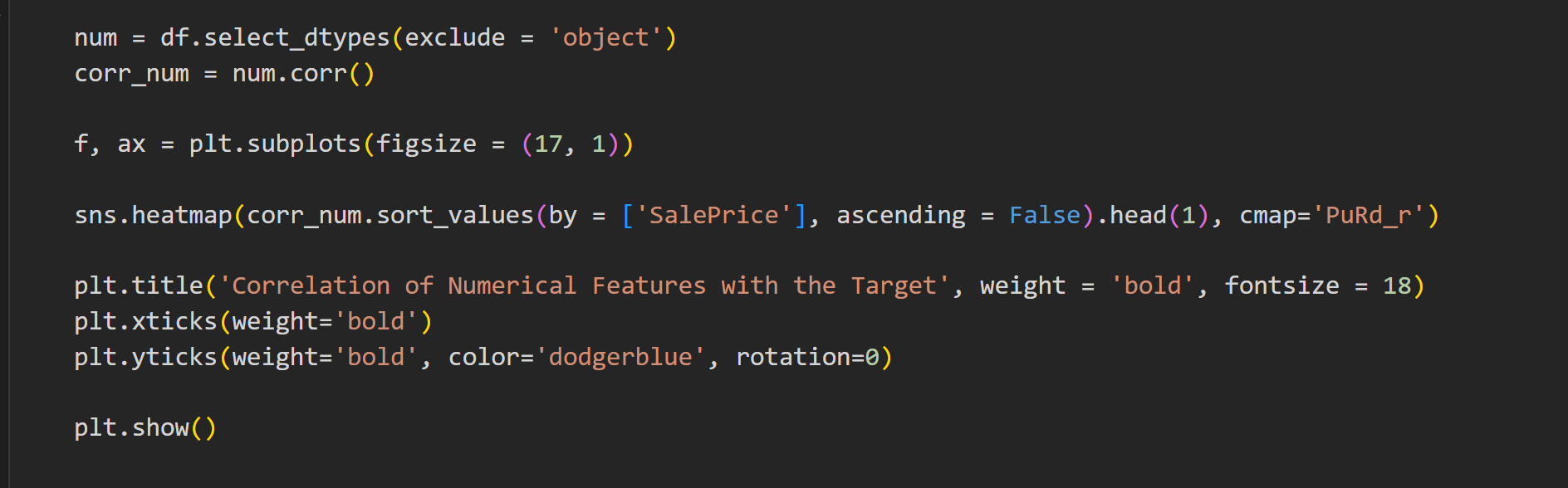


* Kết quả sau khi xử lý độ lệch

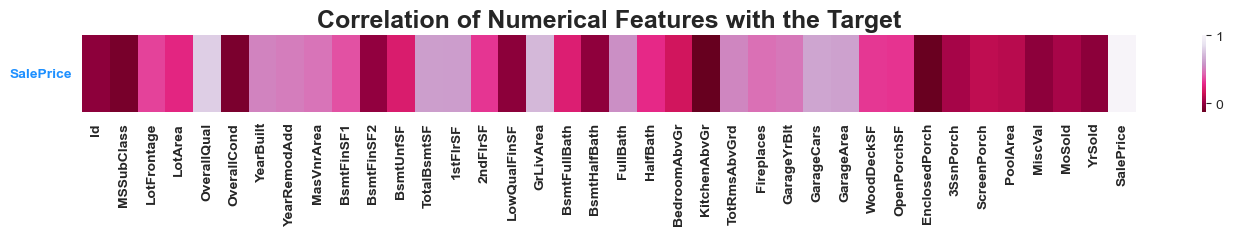


Kết quả xử lí độ lệch của dữ liệu

* Như vậy dữ liệu của chúng ta đã về dạng phân phối chuẩn.
* Chúng ta cần khám phá thêm một vài mối quan hệ giữa các trường dữ liệu để tìm hiểu mối tương quan điều này sẽ giúp chúng ta hiểu hơn về dữ liệu từ đó có thể rút ra những tri thức nào đó

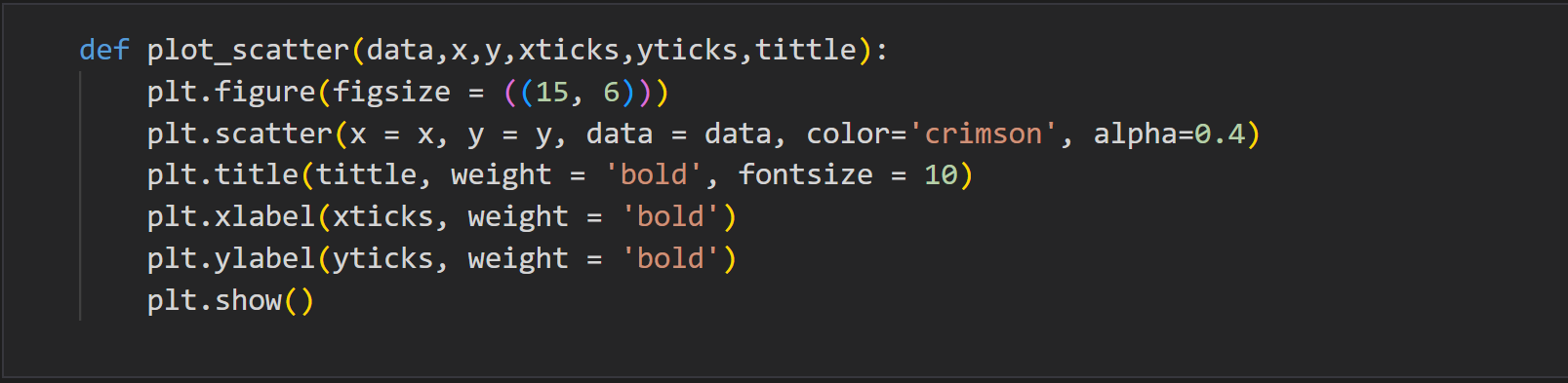


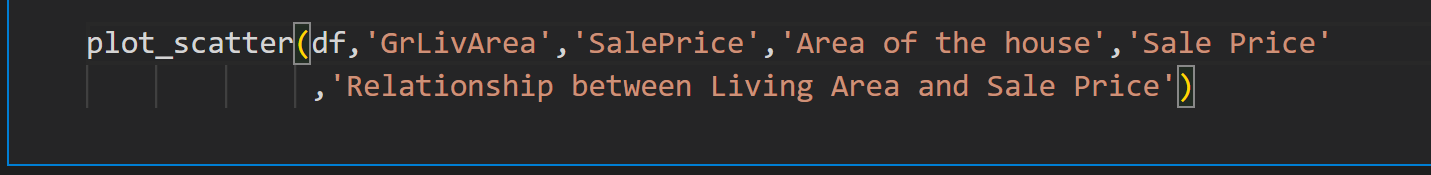
* Mối tương quan giữa biến mục tiêu và các trường còn lại



Mối tương quan của “SalePrice” và các trường khác

* Mối tương quan càng cao cho thấy chúng càng có quan hệ với nhau hơn (tương quan dương – tương quan thuận) và ngược lại. Chất lượng tổng thể ngôi nhà càng tốt thì giá nhà cũng sẽ cao hơn , điều này phù hợp trong thực tế.
* Hàm vẽ biểu đồ phân tán các mối quan hệ các trường với biến mục tiêu “SalePrice”



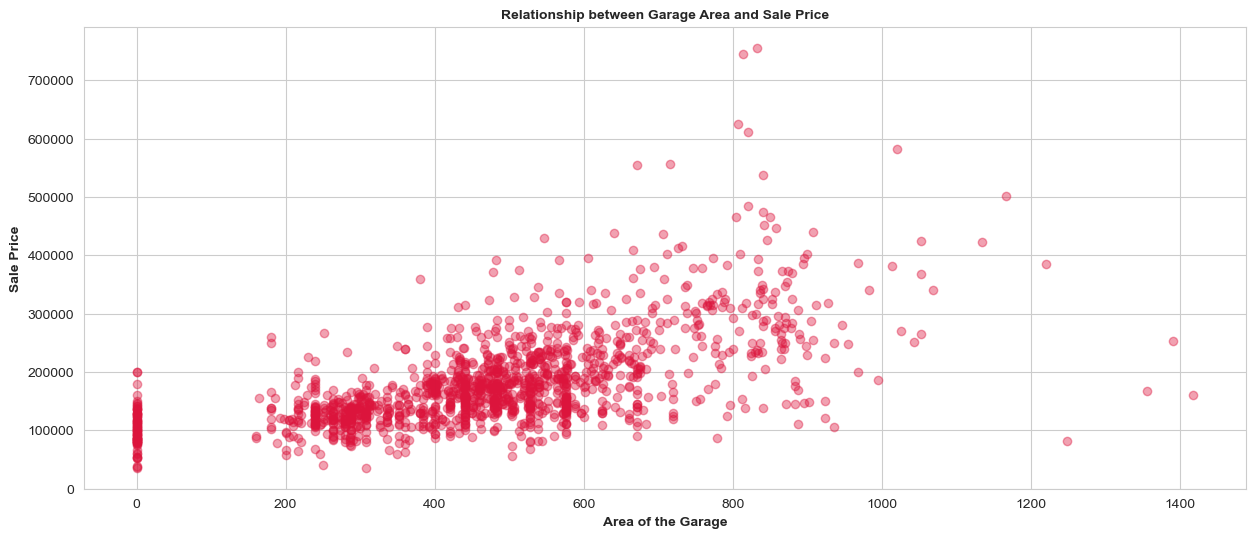


* “GrLivArea” là trường đại diện cho diện tích sinh hoạt ( feet vuông)



Biểu đồ phân rã mối quan hệ giữa “GrLivArea” và “SalePrice”

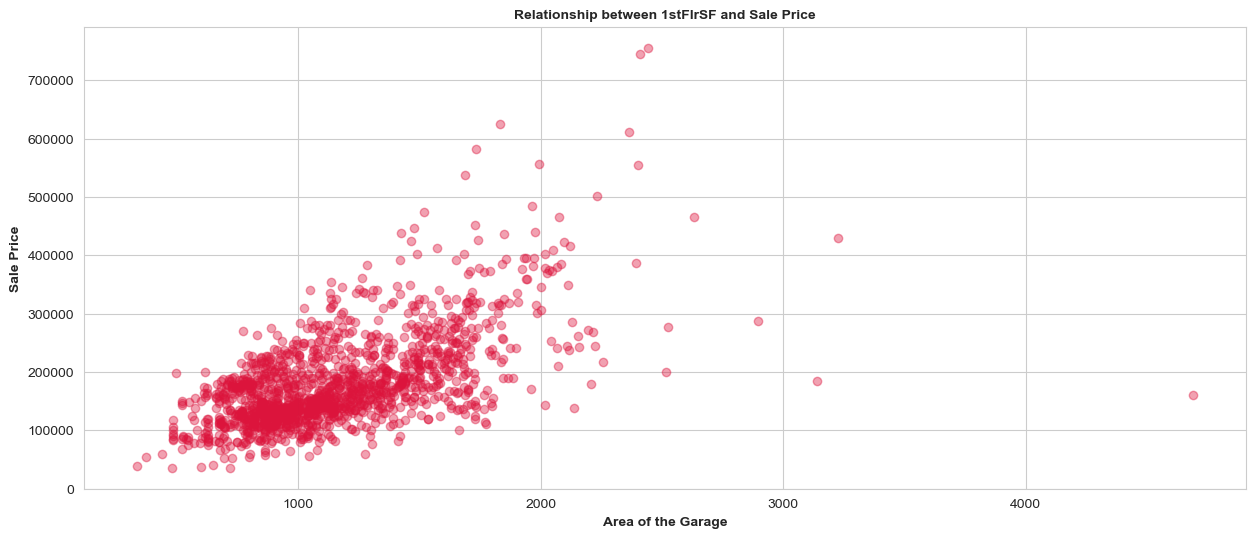
* Ta thấy rằng có mối quan hệ tuyến tính giữa 2 trường này có nghĩa là nếu diện tích sinh hoạt tăng thì giá nhà cũng sẽ tăng
* Chúng ta sẽ tham khảo thêm một vài trường dữ liệu khác mà sẽ ảnh hưởng trực tiếp tới giá nhà



Biểu đồ phân rã mối quan hệ giữa “GarageArea” và “SalePrice”

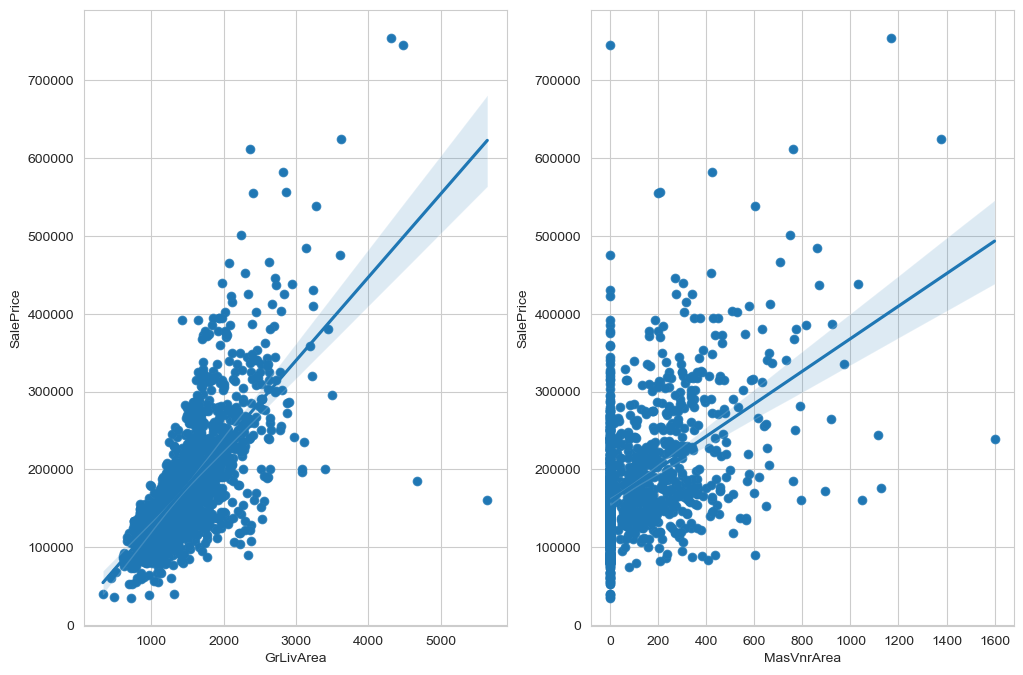


Biểu đồ phân rã mối quan hệ giữa “TotalBsmtSF” và “SalePrice”



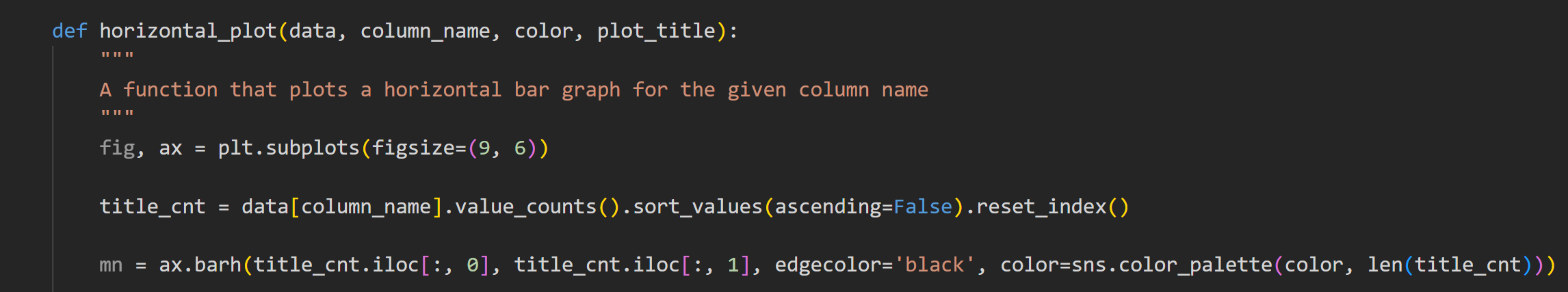
Biểu đồ phân rã mối quan hệ giữa “1stFlrSF” và “SalePrice”

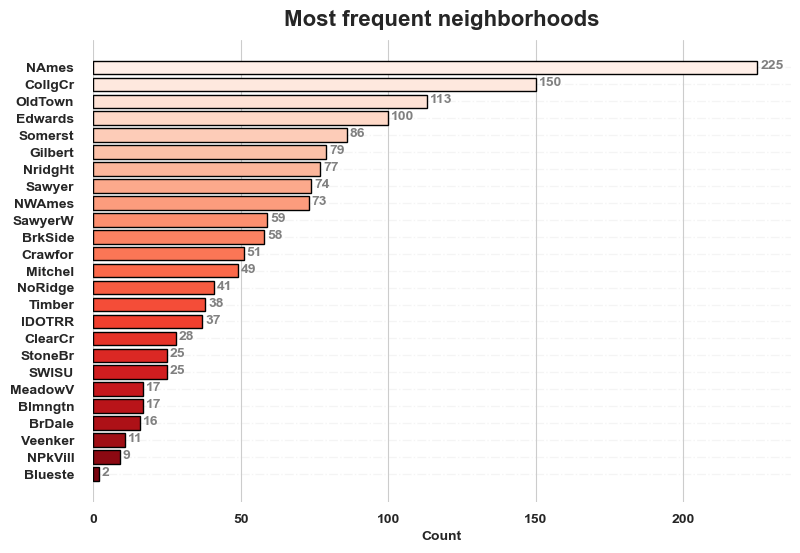
* Để thấy rõ hơn mối quan hệ tuyến tính chúng ta sẽ trực quan bằng thư viện seaborn của Python



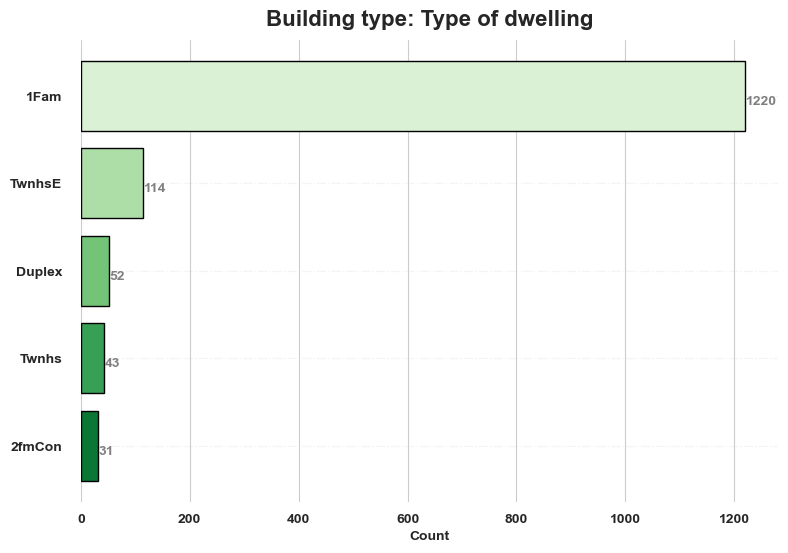
Biểu đồ phân rã tuyến tính giữa các trường so với biến mục tiêu

* Chúng ta có thể nhận thấy rằng mối quan hệ giữa Giá bán và GrLivArea tuyến tính hơn nhiều so với mối quan hệ giữa Giá bán và MasVnrArea.
* Giá trị dư là thước đo mức độ đường hồi quy bỏ sót một điểm dữ liệu theo chiều dọc. Các đường hồi quy phù hợp nhất với một tập hợp dữ liệu. Bạn có thể coi các đường này là mức trung bình; một số điểm dữ liệu sẽ vừa với đường thẳng và những điểm khác sẽ bị thiếu. Biểu đồ dư có các Giá trị dư trên trục tung; trục hoành hiển thị biến độc lập.
* Hãy khám phá thêm về các biến dạng Categorical và xem sự phân bố cũng như thống kê giá trị của một vài trường.
* Hàm vẽ biểu đồ cột ngang thể hiện sự phân phối dữ liệu

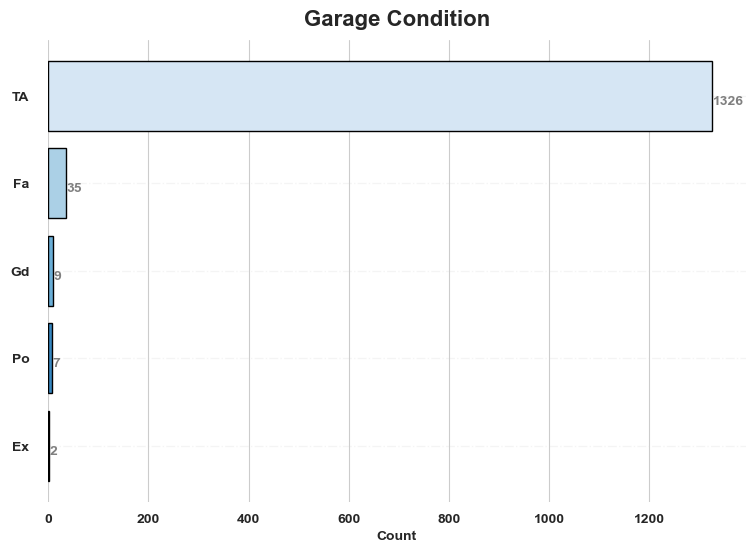




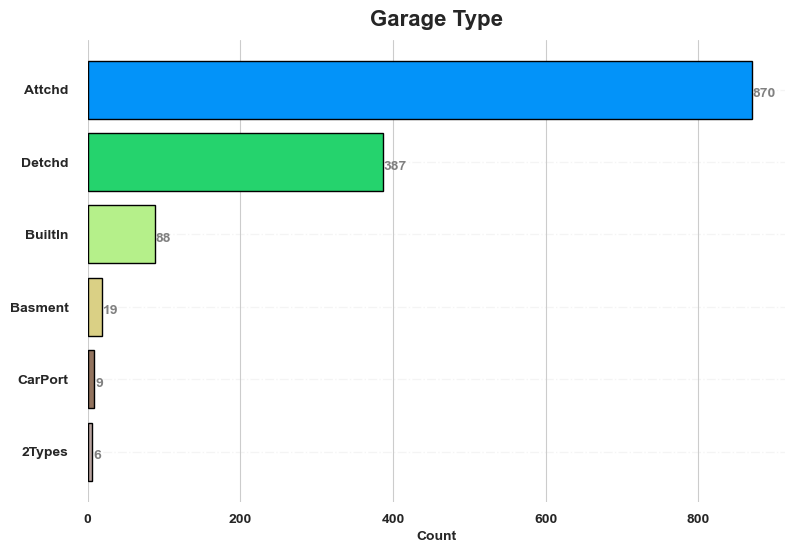
Thống kê số lượng trường “Neighborhood”



Thống kê số lượng trường “BldgType”



Thống kê số lượng trường “GarageCond”



Thống kê số lượng trường “GarageType”

Trong quá trình khám phá và trực quan hóa dữ liệu, chúng em đã tiến hành các hoạt động phân tích để hiểu sâu hơn về bộ dữ liệu về giá nhà bất động sản từ Kaggle. Dưới đây là những kết luận và nhận xét quan trọng từ quá trình này:

* **Cấu trúc dữ liệu:** Chúng tôi đã xác định cấu trúc tổng quan của dữ liệu bao gồm các thuộc tính như diện tích, số phòng ngủ, vị trí địa lý, tiện nghi và giá bán của các căn nhà.
* **Phân bố dữ liệu:** Nhờ vào việc trực quan hóa, chúng tôi nhận thấy phân bố của các thuộc tính khác nhau trong tập dữ liệu, bao gồm các giá trị trung bình, phương sai, và các giá trị ngoại lệ.
* **Tương quan giữa các thuộc tính:** Chúng tôi đã phân tích các mối tương quan giữa các thuộc tính để hiểu rõ hơn về sự phụ thuộc và ảnh hưởng của chúng lên giá nhà.
* **Xử lý dữ liệu thiếu:** Chúng tôi đã xác định và xử lý các giá trị thiếu trong dữ liệu để đảm bảo tính toàn vẹn của tập dữ liệu.
* **Định hướng cho bước tiếp theo:** Dựa trên các phân tích và nhận xét từ bước này, chúng tôi sẵn sàng chuyển sang bước tiền xử lý dữ liệu và xây dựng mô hình dự đoán giá nhà.

### Chuẩn bị dữ liệu cho mô hình

#### Tiêu chuẩn hóa dữ liệu

Tiêu chuẩn hóa dữ liệu là một bước quan trọng trong quá trình chuẩn bị dữ liệu cho mô hình học máy. Mục tiêu của tiêu chuẩn hóa là để đảm bảo rằng tất cả các biến có cùng phạm vi giá trị, giúp mô hình học máy hội tụ nhanh hơn và đạt hiệu suất tốt hơn. Dưới đây là các bước cụ thể để thực hiện tiêu chuẩn hóa dữ liệu:

Chọn phương pháp tiêu chuẩn hóa: Có hai phương pháp tiêu chuẩn hóa phổ biến là chia tỷ lệ (Min-Max Scaling) và chuẩn hóa Z-score (Standardization).

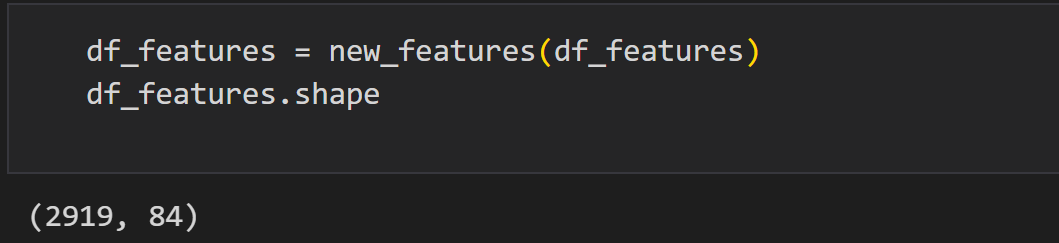
Chia tỷ lệ chuyển đổi giá trị của các biến sao cho chúng nằm trong một phạm vi cố định, thường là [0, 1].

Chuẩn hóa Z-score chuyển đổi giá trị của các biến sao cho chúng có giá trị trung bình gần bằng 0 và độ lệch chuẩn gần bằng 1.

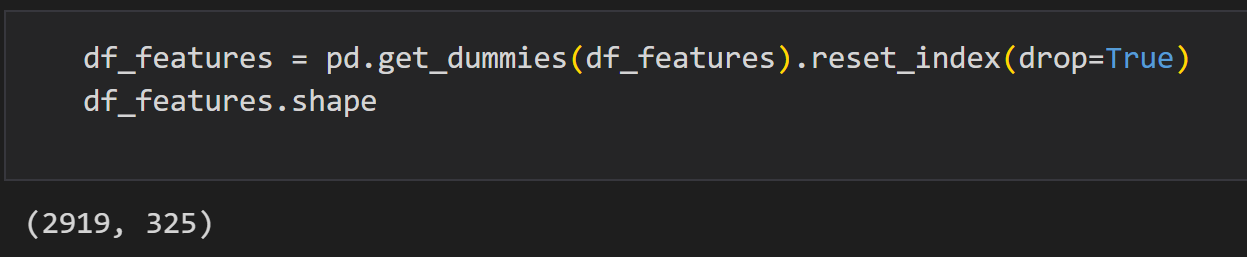
Áp dụng phương pháp tiêu chuẩn hóa: Thực hiện tiêu chuẩn hóa cho tất cả các biến dạng số trong tập dữ liệu. Đảm bảo rằng tiêu chuẩn hóa được thực hiện trên cả tập huấn luyện và tập kiểm tra để đảm bảo tính nhất quán và hiệu quả của mô hình.

Đối với các biến dạng categorical, không cần tiêu chuẩn hóa vì chúng không có giá trị liên tục. Thay vào đó, các biến này cần được chuyển đổi thành dạng số trước khi áp dụng tiêu chuẩn hóa.

* Thực hiện chuẩn hóa dữ liệu đối với dữ liệu dạng Categorical



* Trước khi chuẩn hóa ta thấy bộ dữ liệu có 84 columns
* Hàm get\_dummies() trong thư viện pandas được sử dụng để chuyển đổi các biến phân loại (categorical variables) thành các biến giả (dummy variables) trong phân tích dữ liệu.
* Khi sử dụng get\_dummies(), mỗi giá trị trong biến phân loại sẽ được biến đổi thành một cột riêng (biến giả) trong DataFrame, với các giá trị nhị phân (0 hoặc 1) chỉ ra sự có mặt hoặc vắng mặt của giá trị đó trong mỗi quan sát.



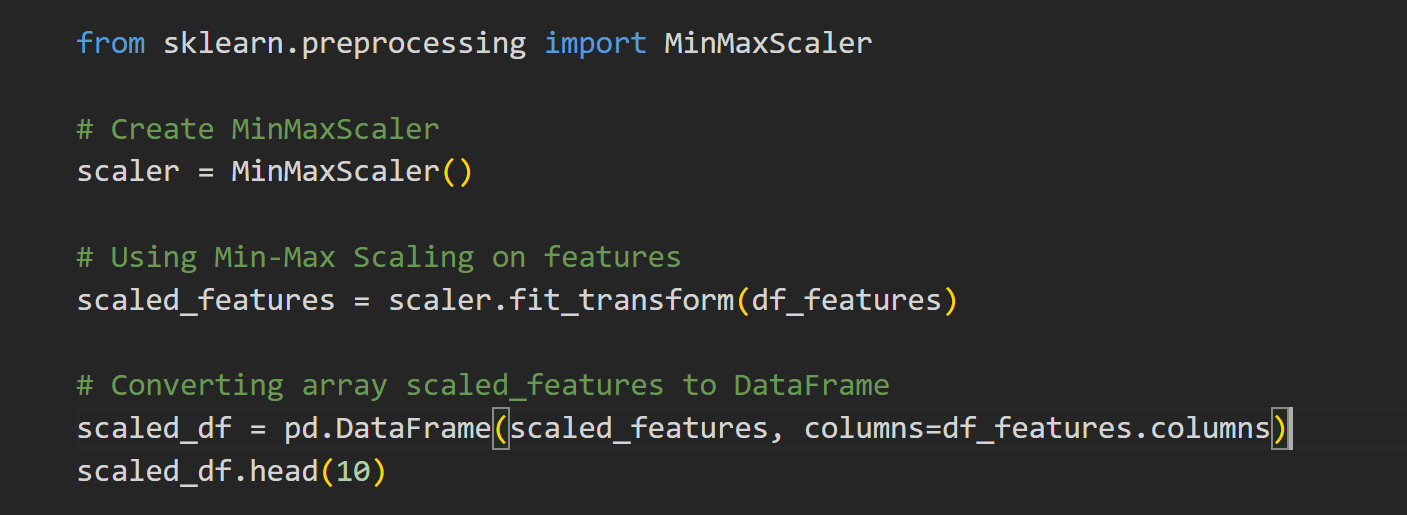
* Sau khi chuẩn hóa dữ liệu ta thấy số lượng columns là 325



Kết quả chuẩn hóa dữ liệu dạng Categorical

Thực hiện co dãn dữ liệu (Scale Data):

* Ở đây chúng ta sẽ lựa chọn phương pháp Min\_Max Scaler đưa bộ dữ liệu về cùng hệ quy chiếu
* Trong thư viện Sklearn của Python cũng đã hỗ trỡ sẵn chúng ta phương thức MinMaxScaler





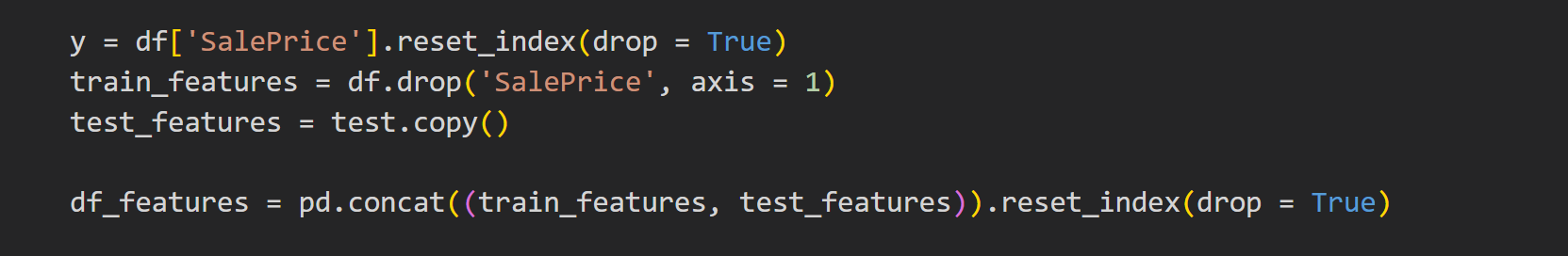
Kết quả sau khi co dãn dữ liệu

Việc tiêu chuẩn hóa dữ liệu sẽ giúp cải thiện hiệu suất của mô hình học máy bằng cách làm cho quá trình hội tụ nhanh hơn và giảm thiểu ảnh hưởng của tỷ lệ tỷ lệ thay đổi của biến đầu vào.

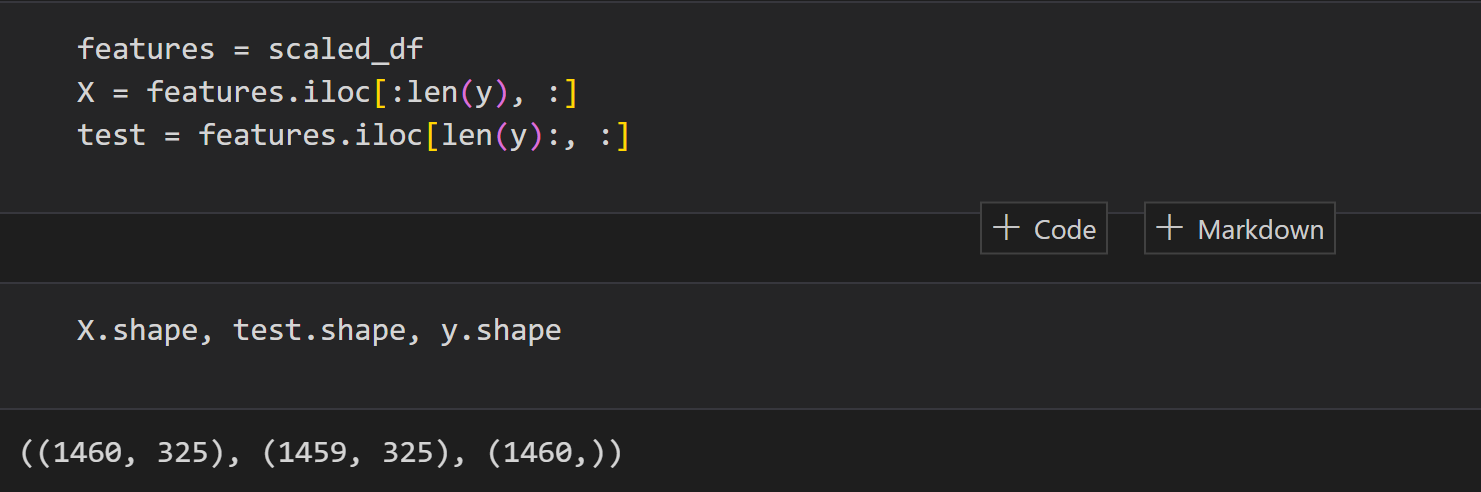
#### Phân chia tập dữ liệu thành tập huấn luyện và tạp kiểm tra

Phân chia dữ liệu thành tập huấn luyện và tập kiểm tra là một trong những bước quan trọng nhất trong quá trình xây dựng mô hình học máy. Quá trình này giúp đảm bảo rằng mô hình của chúng ta được đào tạo trên một phần của dữ liệu và sau đó được kiểm tra trên dữ liệu mà mô hình chưa nhìn thấy trước đó. Dưới đây là một phân tích chi tiết hơn về cách thực hiện phân chia dữ liệu:

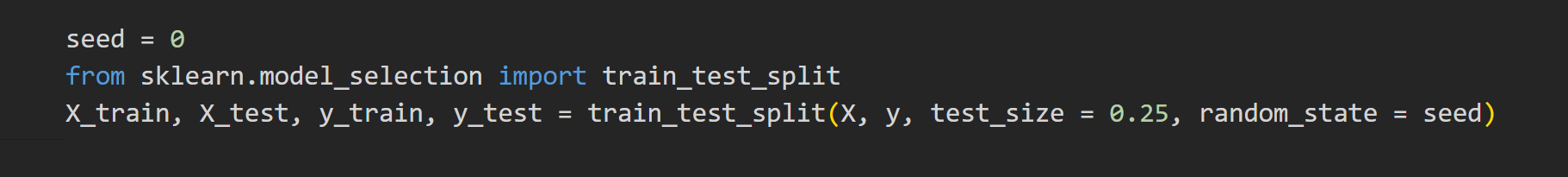
Trước tiên, cần xác định tỷ lệ phân chia giữa tập huấn luyện và tập kiểm tra. Tỷ lệ này thường phụ thuộc vào kích thước của tập dữ liệu và yêu cầu cụ thể của mô hình. Một phần lớn dữ liệu thường được sử dụng cho tập huấn luyện, trong khi một phần nhỏ được dành cho tập kiểm tra. Tỷ lệ phổ biến là 70-80% cho tập huấn luyện và 20-30% cho tập kiểm tra.



* Ở đây chúng ta tách biến mục tiêu “SalePrice” của bộ dữ liệu huấn luyện gán vào biến y vào sau đó gộp tập train và test lại với nhau để thành tập huấn luyện



* Phân chia dữ liệu thành 2 phần:
* “**X**”: là tập dữ liệu chứa các đặc trưng (features) mà bạn muốn sử dụng để huấn luyện mô hình
* “**y**”: là tập dữ liệu chức các nhãn (labels) tương ứng, trong trường hợp này thì là “SalePrice”
* Sau đó tách dữ liệu thành 4 biến sau:



* Sau khi thực hiện ta được 4 biến:
* **‘X\_train`**: Tập dữ liệu đặc trưng cho việc huấn luyện mô hình.
* **`X\_test`**: Tập dữ liệu đặc trưng cho việc kiểm thử mô hình.
* **`y\_train`**: Tập dữ liệu nhãn cho việc huấn luyện mô hình
* **`y\_test`**: Tập dữ liệu nhãn cho việc kiểm thử mô hình

Tập huấn luyện được sử dụng để huấn luyện mô hình, trong khi tập kiểm thử sử dụng để đánh giá hiệu suất của mô hình dựa trên dữ liệu mới mà nó chưa thấy. Việc thực hiện các bước trên sẽ đảm bảo rằng chúng ta có hai tập dữ liệu độc lập, một để đào tạo mô hình và một để kiểm tra mô hình, giúp đánh giá và cải thiện hiệu suất của mô hình một cách chính xác và khách quan.

## Lựa chọn và huấn luyện mô hình

#### Lựa chọn mô hình phù hợp

Trong quá trình dự đoán giá nhà, việc lựa chọn mô hình phù hợp là vô cùng quan trọng để đảm bảo hiệu suất và độ chính xác của dự đoán. Dưới đây là một phân tích chi tiết về ba mô hình được chọn: LASSO Regression, Ridge Regression và XGBoost Regressor:

* **LASSO Regression**

LASSO Regression là một biến thể của hồi quy tuyến tính, được sử dụng để giảm thiểu số lượng biến độc lập không quan trọng trong mô hình. Phương pháp này thực hiện việc đồng thời giảm thiểu hệ số của các biến không quan trọng đến gần với 0 và duy trì các biến quan trọng.

**Ưu điểm:**

LASSO Regression thường là lựa chọn tốt khi chúng ta đang làm việc với dữ liệu có số lượng biến lớn và có sự tương quan giữa các biến. Nó cũng có thể giúp loại bỏ các biến không quan trọng, làm cho mô hình trở nên đơn giản và dễ hiểu hơn.

**Nhược điểm:**

Tuy nhiên, LASSO có thể gặp vấn đề khi các biến có tương quan mạnh với nhau, gây ra hiện tượng lựa chọn biến không ổn định và có thể dẫn đến mất mát thông tin quan trọng.

* **Ridge Regression**

Ridge Regression là một phương pháp hồi quy tuyến tính cũng được sử dụng để giảm thiểu sự phụ thuộc của mô hình vào các biến độc lập. Tuy nhiên, nó thêm một điều kiện vào hàm mất mát để giảm thiểu trọng số của các biến, nhưng không đưa chúng đến 0 như LASSO.

**Ưu điểm:**

Ridge Regression thích hợp cho các tình huống có sự tương quan mạnh giữa các biến độc lập. Nó có thể giúp giảm thiểu hiện tượng overfitting và cải thiện độ ổn định của mô hình.

**Nhược điểm**:

Ridge Regression không thực sự loại bỏ các biến không quan trọng, chỉ làm giảm trọng số của chúng. Điều này có thể là một vấn đề nếu chúng ta muốn loại bỏ các biến không cần thiết hoàn toàn.

* **XGBoost Regressor**

XGBoost là một thuật toán dựa trên cây quyết định, nổi tiếng với hiệu suất cao và khả năng tự điều chỉnh tốt. Nó sử dụng một loạt các cây quyết định để học từ dữ liệu và tối ưu hóa hàm mất mát qua các vòng lặp.

**Ưu điểm:**

XGBoost có khả năng xử lý tốt với dữ liệu lớn, đa biến và không yêu cầu nhiều tiền xử lý. Nó cũng có khả năng xử lý được các biến phức tạp và tương tác giữa các biến.

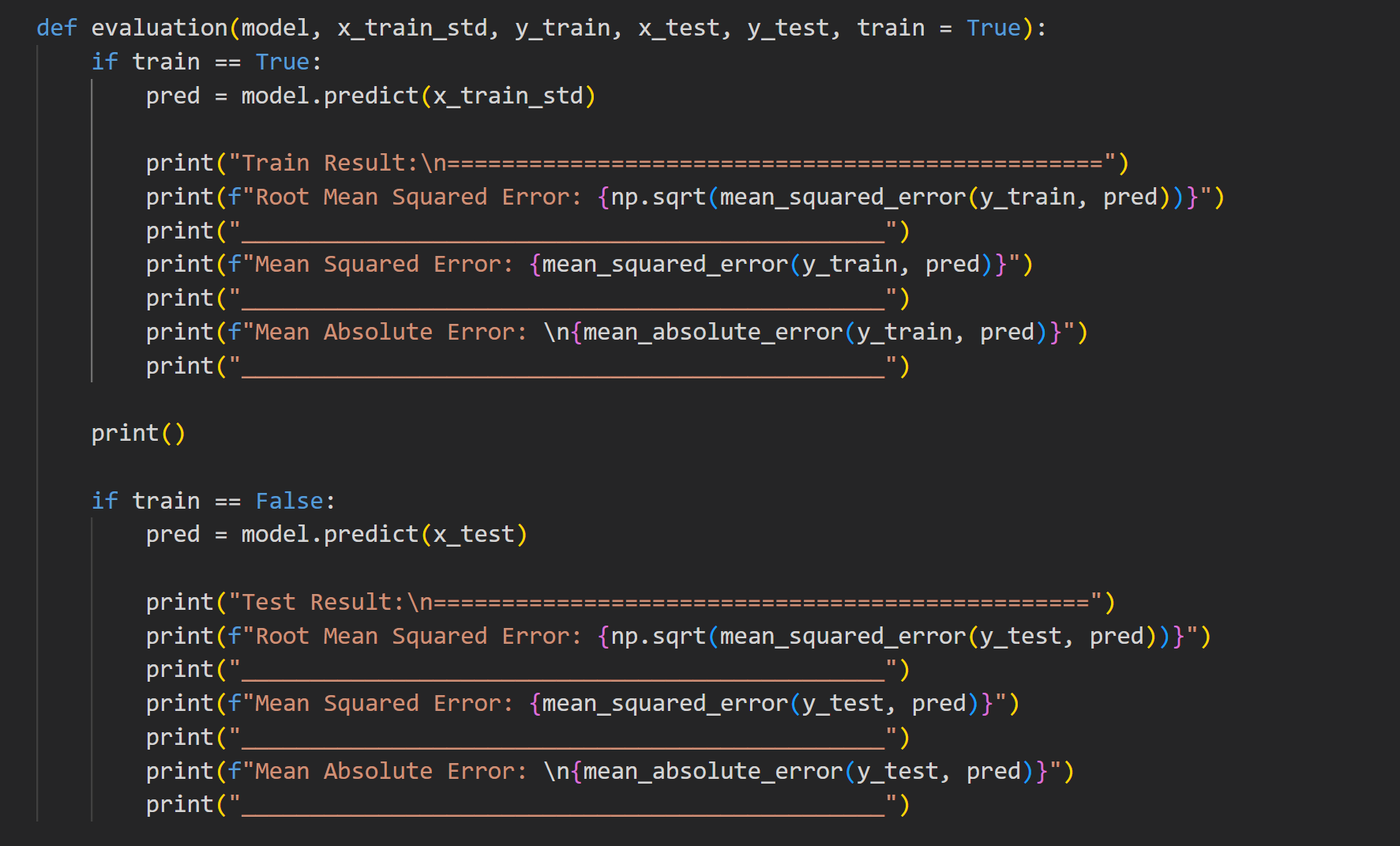
**Nhược điểm:**

XGBoost có thể dễ dàng bị overfitting nếu không được cấu hình đúng, và việc điều chỉnh siêu tham số cũng có thể tốn nhiều thời gian và công sức.

Tùy thuộc vào đặc điểm của tập dữ liệu cụ thể và yêu cầu của bài toán, chúng ta sẽ lựa chọn mô hình phù hợp nhất để xây dựng và huấn luyện trên tập dữ liệu huấn luyện, và sau đó đánh giá hiệu suất của mỗi mô hình trên tập dữ liệu kiểm tra để chọn ra mô hình tốt nhất cho dự đoán giá nhà.

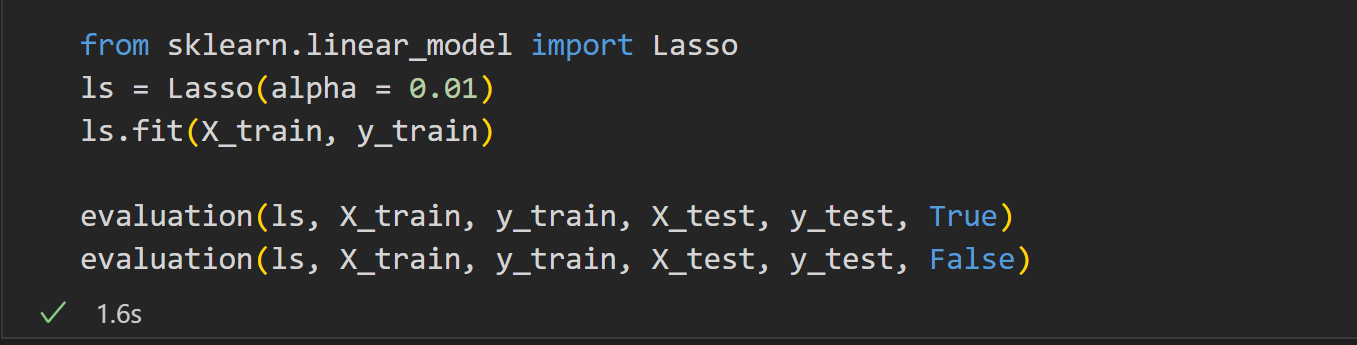
#### Huấn luyện mô hình

* Để bắt đầu huấn luyện chúng ta sẽ khởi tạo hàm evalution dùng để dự đoán cũng như đánh giá hiệu suất của mô hình hồi quy

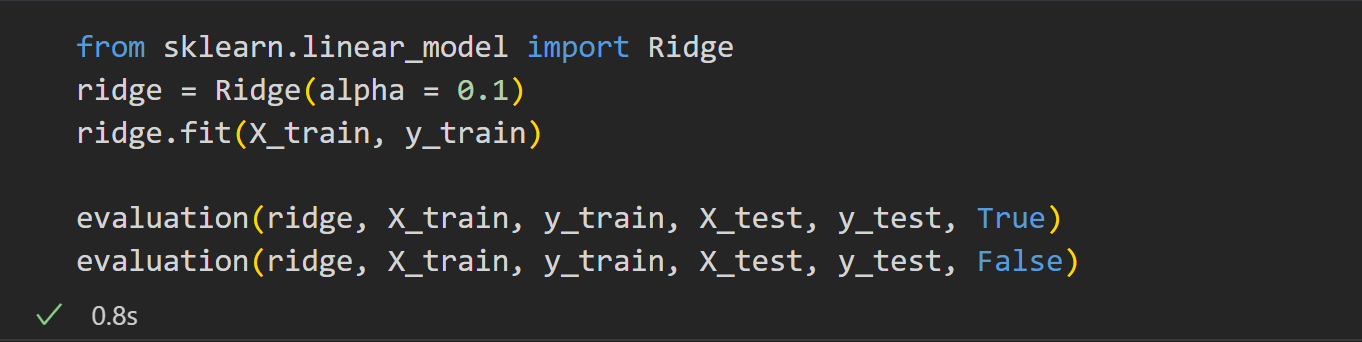


Hàm dùng để huấn luyện và đánh giá hiệu suất

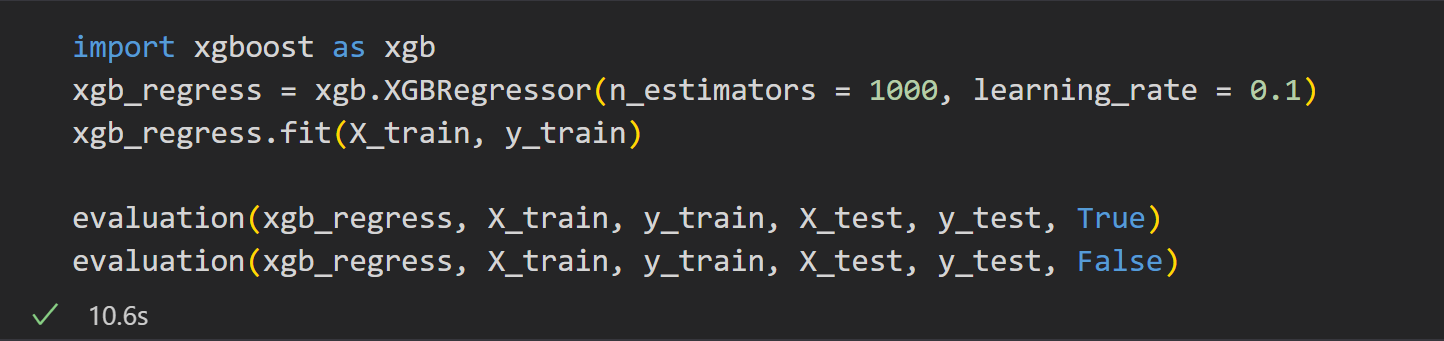
* LASSO Regression
* Huấn luyện dữ liệu sử dụng mô hình hồi quy Lasso



* Ridge Regression
* Huấn luyện dữ liệu sử dụng mô hình hồi quy Ridge



* XGBoost Regressor
* Huấn luyện dữ liệu sử dụng mô hình hồi quy XGBoost



## Đánh giá kết quả mô hình

#### Đánh giá hiệu suất mô hình trên tập kiểm tra

Sau khi đã huấn luyện mô hình trên tập dữ liệu huấn luyện, bước quan trọng tiếp theo là đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra. Việc này giúp đánh giá khả năng tổng quát hóa của mô hình trên dữ liệu mới mà nó chưa từng thấy. Dưới đây là các bước cụ thể để đánh giá hiệu suất của mô hình trên tập kiểm tra:

Sử dụng mô hình đã được huấn luyện để dự đoán giá trị của biến phụ thuộc trên tập dữ liệu kiểm tra. Điều này tạo ra một tập dự đoán cho mô hình trên dữ liệu kiểm tra.

Sử dụng các độ đo hiệu suất như Mean Squared Error (MSE), Mean Absolute Error (MAE), hoặc R-squared để đánh giá độ chính xác của mô hình trên tập dữ liệu kiểm tra. So sánh kết quả này với hiệu suất của mô hình trên tập huấn luyện để đánh giá khả năng tổng quát hóa của mô hình.

Kiểm tra overfitting: Đối chiếu hiệu suất của mô hình trên tập huấn luyện và tập kiểm tra để xem xét có dấu hiệu của overfitting hay không. Nếu hiệu suất trên tập kiểm tra kém hơn so với tập huấn luyện, có thể mô hình đang bị overfitting.

Tối ưu hóa mô hình: Dựa trên kết quả của việc đánh giá hiệu suất trên tập kiểm tra, có thể cần điều chỉnh các tham số của mô hình hoặc thực hiện các kỹ thuật tinh chỉnh mô hình để cải thiện hiệu suất trên tập kiểm tra.

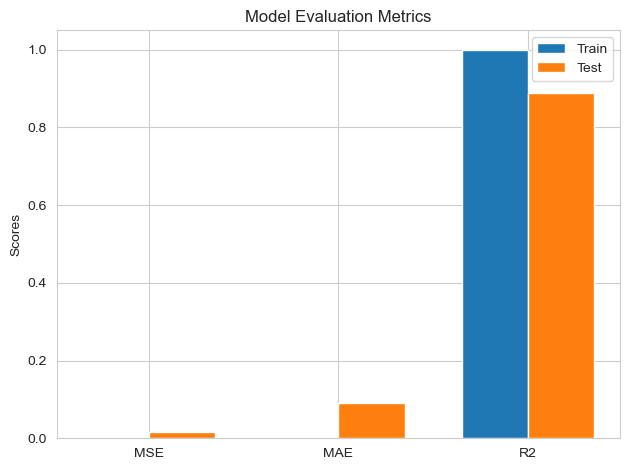
* Chúng ta hãy so sánh hiệu xuất của các mô hình thông qua việc trực quan hóa



Biểu đồ đánh giá hiệu suất mô hình Lasso Regression



Biểu đồ đánh giá hiệu suất mô hình Ridge Regression



Biểu đồ đánh giá hiệu suất mô hình XGBoost

* Mean Squared Error (MSE) là độ đo đánh giá độ lỗi trung bình. Giá trị càng thấp thì mô hình càng tốt
* Nhìn vào biểu đồ trên ta thấy:
* Mô hình Ridge có MSE thấp nhất trên tập huấn luyện và gần nhất trên tập kiểm tra, cho thấy mô hình Ridge có xu hướng hoạt động tốt trên cả hai tập dữ liệu.
* Mô hình XGBoost có MSE thấp nhất trên tập kiểm tra, tuy nhiên có MSE cao hơn trên tập huấn luyện so với mô hình Ridge, có thể là dấu hiệu của overfitting trên tập huấn luyện.
* Mean Absolute Error (MAE) là sự khác biệt trung bình giữa giá trị dự đoán và giá trị thực tế. Giá trị MAE càng thấp thì mô hình càng tốt.
* Mô hình XGBoost có MAE thấp nhất trên cả tập huấn luyện và tập kiểm tra, cho thấy mô hình XGBoost có khả năng dự đoán chính xác hơn về mặt giá trị trung bình.
* R-squared (R2 Score) Score là độ đo đánh giá khả năng giải thích của mô hình, từ 0 đến 1. Giá trị R2 càng gần 1 thì mô hình càng tốt.
* Mô hình XGBoost có R2 Score cao nhất trên cả tập huấn luyện và tập kiểm tra, cho thấy mô hình XGBoost có khả năng giải thích tốt nhất đối với biến độc lập.
* Nhìn chung mô hình XGBoost có xu hướng hoạt động tốt nhất dựa trên các chỉ số MSE, MAE và R2 Score trên cả tập huấn luyện và tập kiểm tra. Mô hình Ridge cũng cho kết quả khá tốt với MSE thấp và R2 Score gần 1 trên tập kiểm tra, tuy nhiên, có sự khác biệt về MAE so với mô hình XGBoost. Mô hình Lasso cho kết quả chưa tốt bằng hai mô hình kia trên các chỉ số đánh giá.
* Việc đánh giá hiệu suất của mô hình trên tập kiểm tra là bước quan trọng để đảm bảo rằng mô hình đã được huấn luyện tốt và có khả năng tổng quát hóa tốt trên dữ liệu mới.

#### Tối ưu hóa siêu tham số mô hình

Trong quá trình xây dựng mô hình học máy, việc tinh chỉnh các siêu tham số là một bước quan trọng để đạt được hiệu suất tốt nhất của mô hình. Siêu tham số là các tham số không được học từ dữ liệu mà cần phải được xác định trước, ví dụ như tốc độ học (learning rate), số lượng cây (n\_estimators) trong các mô hình cây quyết định, hệ số điều chuẩn (alpha) trong các mô hình hồi quy, vv…

Dưới đây là các bước để tối ưu hóa siêu tham số của mô hình:

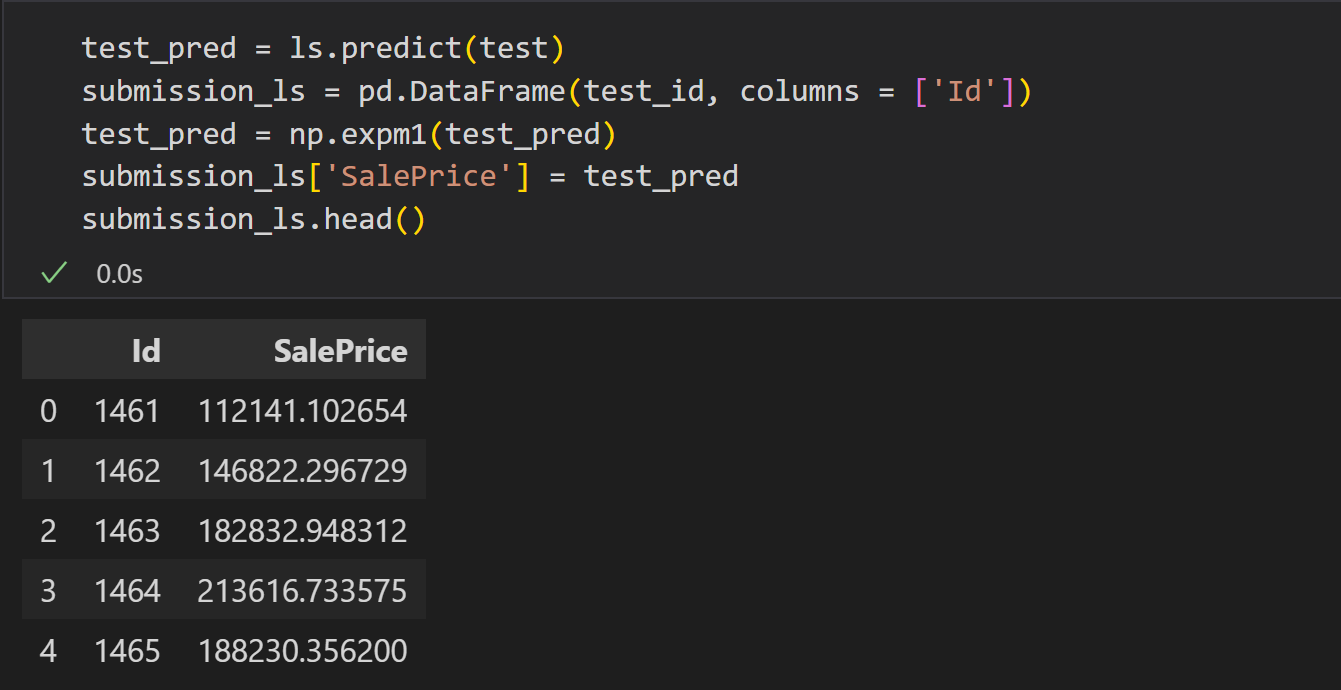
* **Xác định siêu tham số cần tối ưu:** Xác định danh sách các siêu tham số cần được tinh chỉnh cho mô hình cụ thể mà bạn đang sử dụng. Điều này có thể là một bước phức tạp và đòi hỏi hiểu biết sâu rộng về cách hoạt động của mô hình.
* **Chọn phương pháp tối ưu hóa:** Có một số phương pháp để tối ưu hóa siêu tham số, bao gồm tìm kiếm lưới (grid search), tìm kiếm ngẫu nhiên (random search), tối ưu hóa bayesian, tối ưu hóa đạo hàm, vv. Mỗi phương pháp có ưu nhược điểm riêng và phù hợp với các tình huống khác nhau.
* **Thực hiện tối ưu hóa:** Áp dụng phương pháp tối ưu hóa được chọn để tìm ra bộ siêu tham số tốt nhất cho mô hình của bạn. Quá trình này thường đòi hỏi huấn luyện và đánh giá mô hình với nhiều bộ tham số khác nhau trên tập dữ liệu kiểm tra hoặc sử dụng kỹ thuật chia tập dữ liệu chéo (cross-validation) để đảm bảo tính đáng tin cậy của kết quả.
* **Đánh giá hiệu suất:** Sau khi đã tinh chỉnh xong, đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra để đảm bảo rằng mô hình đã được tối ưu hóa đúng cách và có khả năng tổng quát hóa tốt trên dữ liệu mới.
* **Lưu trữ và triển khai:** Lưu trữ bộ siêu tham số tốt nhất và sử dụng chúng để huấn luyện mô hình trên toàn bộ tập dữ liệu huấn luyện. Mô hình đã được tối ưu hóa có thể được triển khai để dự đoán trên dữ liệu thực tế.

Quá trình tối ưu hóa siêu tham số giúp cải thiện hiệu suất của mô hình và là một bước quan trọng trong quá trình xây dựng một hệ thống học máy có khả năng dự đoán chính xác và đáng tin cậy.

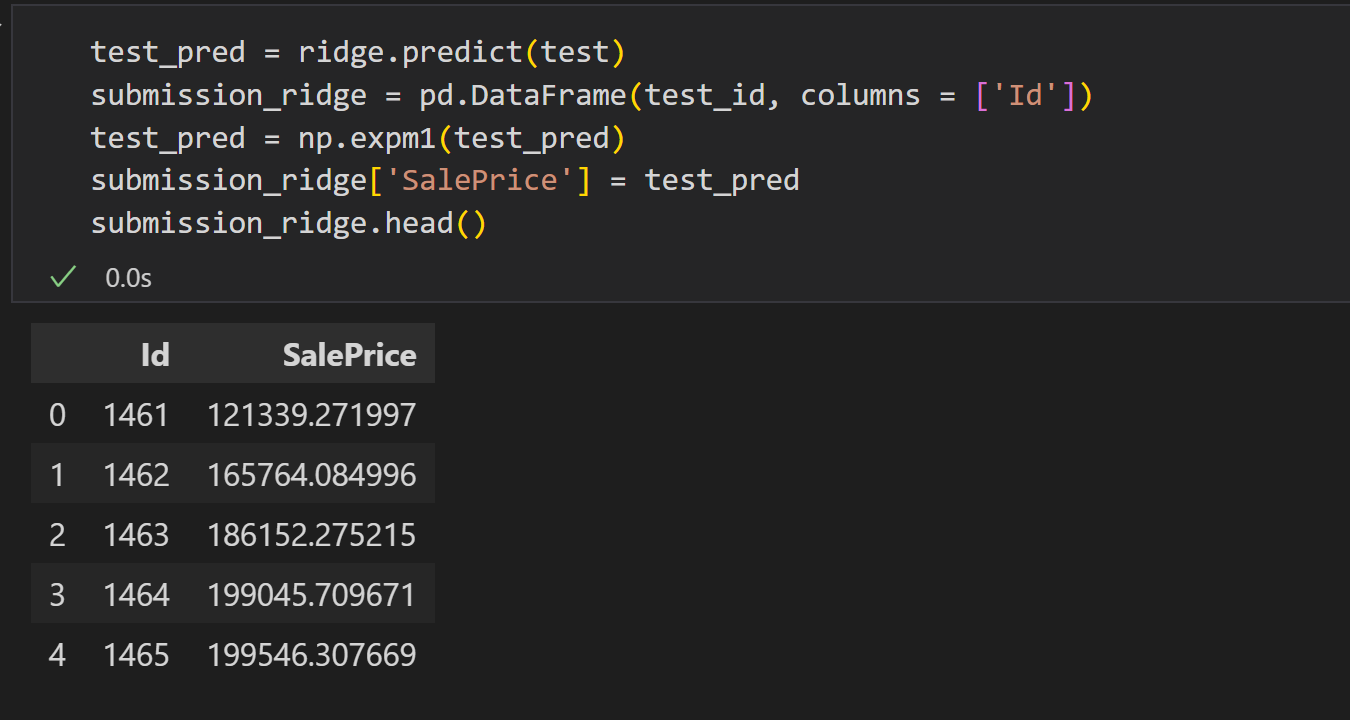
#### So sánh kết quả của các mô hình

Sau khi đã huấn luyện và tinh chỉnh các mô hình khác nhau, bước tiếp theo là so sánh kết quả của chúng để xác định mô hình nào hoạt động tốt nhất cho bài toán dự đoán giá nhà.

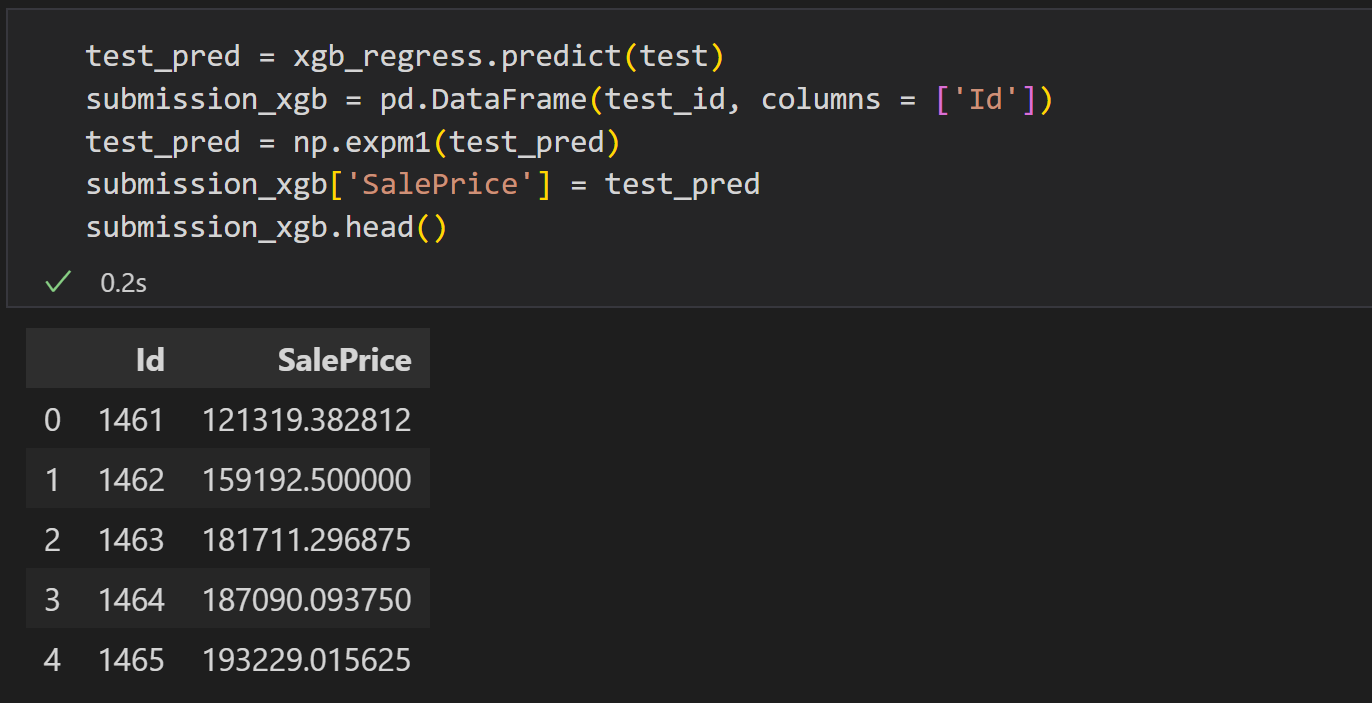
* Kết quả dự đoán giá nhà của các mô hình



Kết quả dự đoán của mô hình Lasso Regression

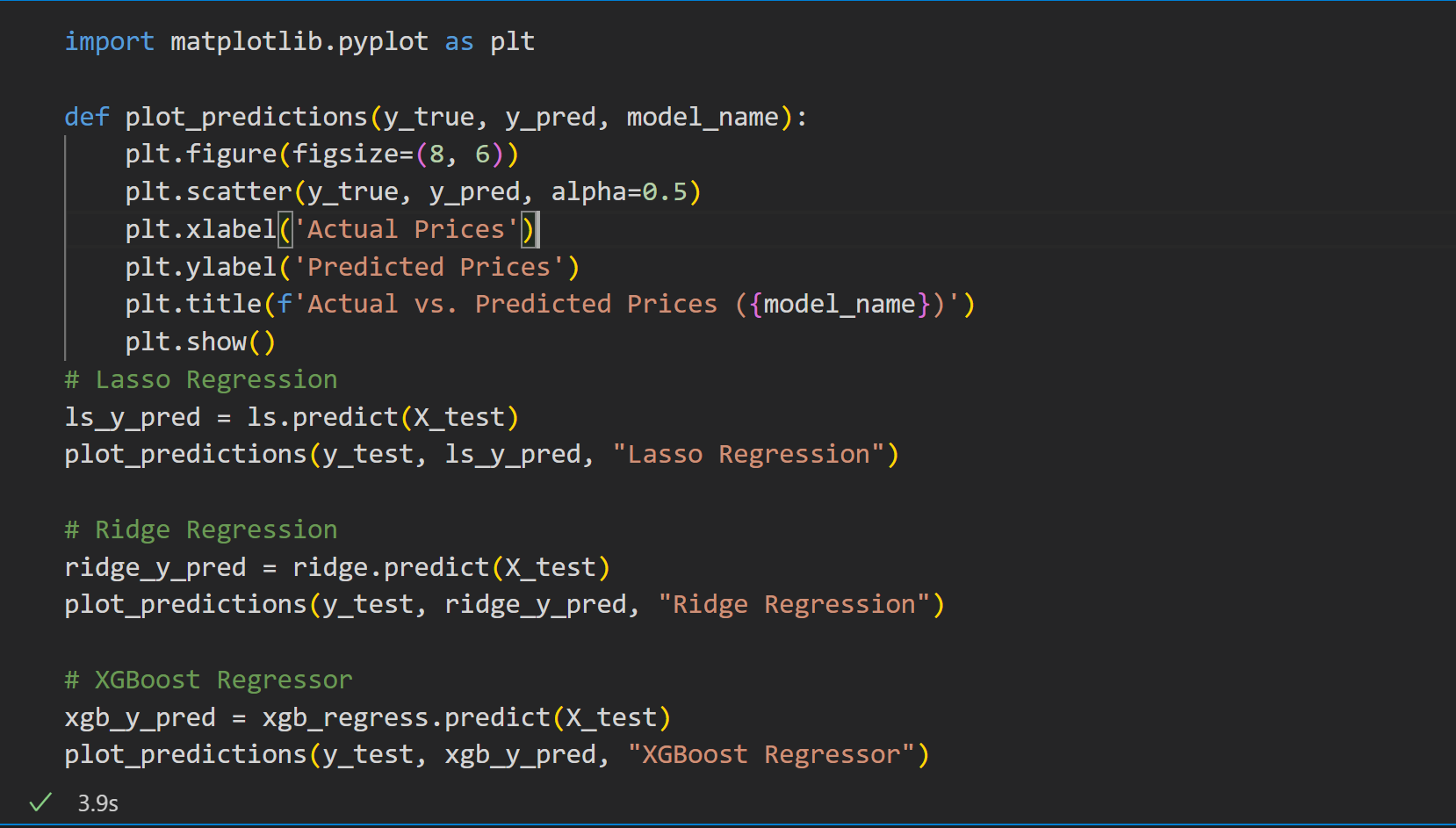


Kết quả dự đoán của mô hình Ridge Regression

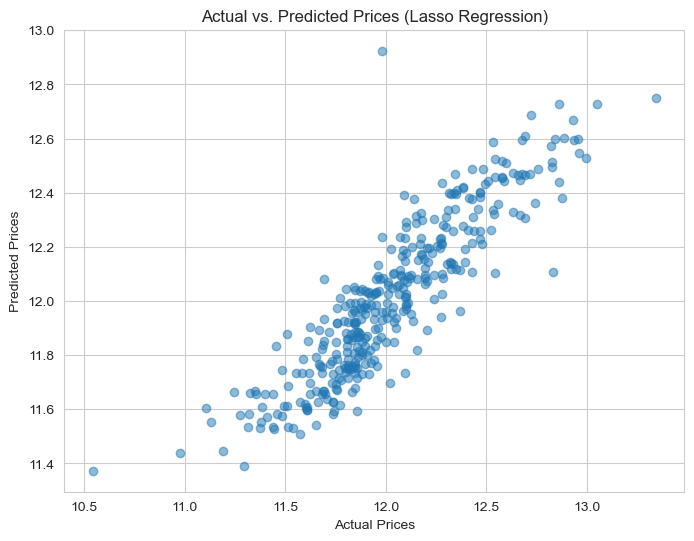


Kết quả dự đoán của mô hình XGBoost Regression

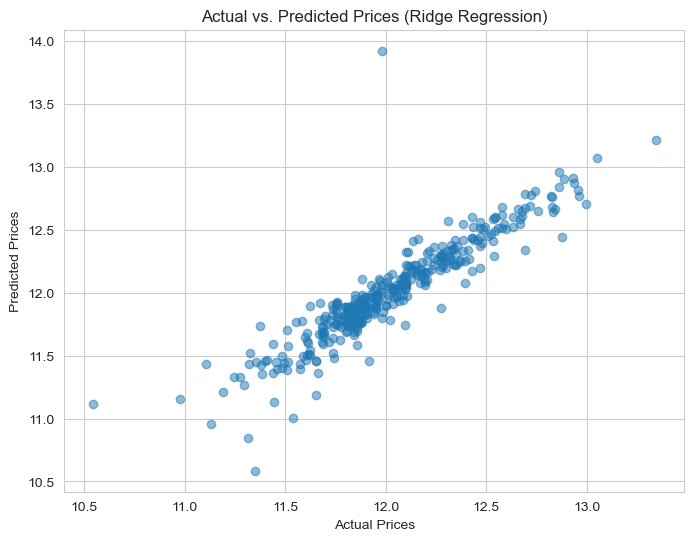
* Để thấy rõ hơn sự khác biệt chúng ta sẽ trực quan giá trị thực tế và giá trị dự đoán của mô hình trên biểu đồ phân tán mà thư viện Matplolib của Python cung cấp



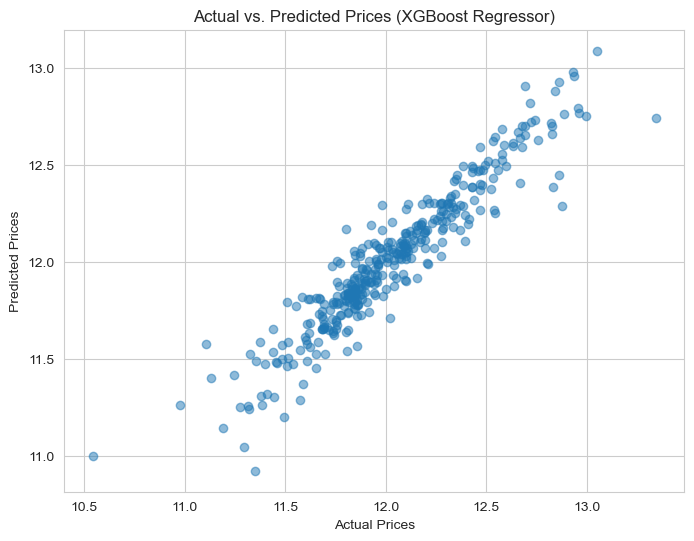
* Xem kết quả của từng biểu đồ



Biểu đồ so sánh giá thực tế và giá dự đoán của mô hình Lasso Regression



Biểu đồ so sánh giá thực tế và giá dự đoán của mô hình Ridge Regression



Biểu đồ so sánh giá thực tế và giá dự đoán của mô hình XGBoost Regression

* Như đã đánh giá ở trên có thể thấy mô hình XGBoost có vẻ là mô hình có hiệu suất cao nhất, như vậy chúng ta có thể dễ dàng đưa ra lựa chọn cho việc chọn lựa mô hình để từ đó có thể tích hợp vào các hệ thống khác nhằm giải quyết những vấn đề trong tương lai.

KẾT LUẬN

Trong quá trình dự án, chúng em đã tiến hành một loạt các bước để xây dựng và đánh giá các mô hình dự đoán giá nhà. Dưới đây là tóm tắt chi tiết về kết quả của quá trình này:

* **Tiền xử lý dữ liệu:** Trước khi xây dựng mô hình, chúng em đã thực hiện tiền xử lý dữ liệu để xử lý các giá trị thiếu, chuyển đổi biến dạng categorical thành dạng số, và tiêu chuẩn hóa dữ liệu số. Quá trình này giúp chúng tôi chuẩn bị dữ liệu để huấn luyện và kiểm tra mô hình một cách hiệu quả.
* **Phân tích và khai phá dữ liệu:** Quá trình này mục đích giúp chúng em có thể đưa ra những đánh giá để hiểu hơn về dữ liệu từ đó có những phương hướng và xử lý bài toán sao cho phù hợp với yêu cầu.
* **Xây dựng và huấn luyện mô hình:** Chúng em đã lựa chọn ba mô hình khác nhau để huấn luyện trên tập dữ liệu huấn luyện:
* LASSO Regression: Một phương pháp regression sử dụng regularization để giảm overfitting và chọn ra các biến quan trọng.
* Ridge Regression: Một phương pháp regression khác sử dụng regularization để giảm overfitting bằng cách giảm trọng số của các biến.
* XGBoost Regressor: Một mô hình ensemble learning sử dụng cây quyết định để tạo ra các dự đoán và kết hợp chúng lại với nhau.
* **Đánh giá kết quả:** Chúng em đã đánh giá hiệu suất của các mô hình bằng cách sử dụng các độ đo như Mean Squared Error (MSE), Mean Absolute Error (MAE), và R-squared trên tập dữ liệu kiểm tra. Các mô hình đã đạt được hiệu suất tốt, với các độ đo lỗi thấp và R-squared cao, cho thấy chúng có khả năng dự đoán chính xác giá nhà.

Do thời gian triển khai ngắn, chúng em thừa nhận rằng vẫn còn nhiều sơ sót trong đồ án. Kiến thức của chúng em về mặt toán học và khả năng lập trình còn hạn chế. Tuy nhiên, chúng em sẽ tiếp tục nỗ lực hơn và không ngừng học hỏi để cải thiện đồ án của mình trong thời gian tới

**Hướng Phát Triển:**

Một cách để cải thiện hiệu suất của mô hình là mở rộng dữ liệu bằng cách bổ sung các biến mới hoặc thu thập thêm dữ liệu từ các nguồn khác. Điều này có thể giúp mô hình hiểu rõ hơn về các yếu tố ảnh hưởng đến giá nhà và cung cấp dự đoán chính xác hơn.

Tinh chỉnh siêu tham số: Mặc dù chúng tôi đã tinh chỉnh siêu tham số cho các mô hình của mình, nhưng việc tiếp tục tinh chỉnh có thể cải thiện hiệu suất của chúng. Sử dụng các kỹ thuật tinh chỉnh siêu tham số như Grid Search hoặc Random Search có thể giúp chúng em tìm ra các siêu tham số tối ưu hóa hơn.

Sử dụng các phương pháp ensemble learning: Việc kết hợp các mô hình khác nhau thành một mô hình ensemble có thể cải thiện hiệu suất dự đoán. Chúng em có thể thử nghiệm các phương pháp như Bagging, Boosting, hoặc Stacking để tạo ra một mô hình mạnh mẽ hơn từ các mô hình cá nhân.

Xem xét các mô hình học sâu: Các mô hình học sâu như Neural Networks có thể mang lại hiệu suất tốt cho bài toán dự đoán giá nhà. Chúng tôi có thể nghiên cứu và thử nghiệm các mạng neural có cấu trúc phù hợp để xem xét khả năng dự đoán của chúng.

Đánh giá sự ảnh hưởng của các biến khác: Ngoài các biến đã sử dụng trong dự án này, chúng tôi có thể xem xét sự ảnh hưởng của các biến khác như đặc điểm khu vực, tiện ích xung quanh, hoặc xu hướng thị trường bất động sản. Điều này có thể cung cấp cái nhìn tổng quan và chính xác hơn về giá nhà.

Nghiên cứu về tính phổ quát của mô hình: Để đảm bảo tính tổng quát của mô hình, chúng tôi có thể nghiên cứu về tính phổ quát của mô hình trên các tập dữ liệu khác nhau hoặc trong các vùng địa lý khác nhau. Điều này giúp chúng tôi đảm bảo rằng mô hình có thể được áp dụng trong nhiều tình huống khác nhau một cách hiệu quả.

TÀI LIỆU THAM KHẢO

1. Grus, Joel. Data science from scratch: first principles with python. O'Reilly Media, Inc.", 2015.
2. Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.
3. Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. TheElements of Statistical Learning: Data Mining, Inference, andPrediction (2nd Edition). Springer series in statistics, 2009.
4. <https://drive.google.com/drive/folders/1EikCoBH717lrlrr9oOy5S9WaNYsEDYcT>
5. [https://zootopi.dev/tutorial/data\_science](duDoanGiaBatDongSan)
6. [https://docs.python.org/3/](duDoanGiaBatDongSan)
7. [https://www.kaggle.com](duDoanGiaBatDongSan)
8. [https://scikit-learn.org/](duDoanGiaBatDongSan)
9. [https://machinelearningcoban.com/](duDoanGiaBatDongSan)