**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KỲ**

**MÔN HỌC: LẬP TRÌNH PYTHON**

**DỰ ĐOÁN GIÁ MÁY TÍNH XÁCH TAY**

**Mã lớp học phần: IPPA233277**

**Học kỳ 1 – Năm học 2024-2025**

**Giảng viên hướng dẫn: ThS. Trần Quang Khải**

**Danh sách sinh viên thực hiện:**

|  |  |
| --- | --- |
| **MSSV** | **Họ tên** |
| 23110176 | Nguyễn Hoàng Anh |
| 23110223 | Nguyễn Đình Huy |
| 23110249 | Lưu Thái Thanh Lâm |

***Thành phố Hồ Chí Minh, tháng 11 năm 2024***

*Nhận xét của giảng viên:*

*TP. Hồ Chí Minh, ngày … tháng… năm 2024*

*Giảng viên ký tên*

**MỤC LỤC**

[**LỜI CẢM ƠN** 9](#_Toc183593712)

[**PHẦN 1: MỞ ĐẦU** 11](#_Toc183593713)

[**1.** **Lý do chọn đề tài** 11](#_Toc183593714)

[**2.** **Mục tiêu đề tài** 11](#_Toc183593715)

[**3.** **Đối tượng và phạm vi nghiên cứu** 12](#_Toc183593716)

[**PHẦN 2. NỘI DUNG** 13](#_Toc183593717)

[**CHƯƠNG 1: TÓM TẮT CƠ SỞ LÝ THUYẾT** 13](#_Toc183593718)

[**1.1.** **Giới thiệu về lập trình python** 13](#_Toc183593719)

[**1.2. Giới thiệu về các thư viện thường được sử dụng trong lập trình python và môi trường lập trình** 14](#_Toc183593720)

[**CHƯƠNG 2: MÔ TẢ TẬP DỮ LIỆU GIÁ MÁY TÍNH XÁCH TAY** 17](#_Toc183593721)

[**2.1. Các tập dữ liệu cần thu thập** 17](#_Toc183593722)

[**CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU GIÁ MÁY TÍNH (LAPTOP\_PRICE)** 19](#_Toc183593723)

[**3.1. Khởi tạo dữ liệu** 19](#_Toc183593724)

[**3.2. Chuẩn hóa dữ liệu và làm sạch dữ liệu(Data normalizer and cleaning)** 19](#_Toc183593725)

[**3.2.1. Làm sạch dữ liệu(Data Cleaning)** 19](#_Toc183593726)

[**3.2.2. Chuẩn hóa dữ liệu(Data normalizer)** 22](#_Toc183593727)

[**3.3. Thực hiện các cơ chế CRUD** 24](#_Toc183593728)

[**3.3.1. Cơ chế READ** 24](#_Toc183593729)

[**3.3.2. Cơ chế CREATE** 25](#_Toc183593730)

[**3.3.3. Cơ chế UPDATE** 27](#_Toc183593731)

[**3.3.4. Cơ chế DELETE** 30](#_Toc183593732)

[**3.3.5. Cơ chế sắp xếp dữ liệu(Sort)** 32](#_Toc183593733)

[**3.3.6. Cơ chế tìm kiếm dữ liệu(Search)** 33](#_Toc183593734)

[**3.4. Phân tích dữ liệu (Data analysis)** 34](#_Toc183593735)

[**3.4.1. Các thư viện cần thiết** 34](#_Toc183593736)

[**3.4.2. Xử lý dữ liệu thiếu và trùng lặp** 35](#_Toc183593737)

[**3.4.3 Nhóm dữ liệu theo công ty và vẽ biểu đồ** 37](#_Toc183593738)

[**3.4.4. Phân phối kích thước màn hình** 39](#_Toc183593739)

[**CHƯƠNG 4: GIAO DIỆN TƯƠNG TÁC NGƯỜI DÙNG** 40](#_Toc183593740)

[**4.1. Thư viện Tkinter:** 40](#_Toc183593741)

[**4.1.1. Định nghĩa và các thành phần của thư viện Tkinter:** 40](#_Toc183593742)

[**4.1.2. Ưu điểm, nhược điểm và ứng dụng của thư viện Tkinter:** 44](#_Toc183593743)

[**4.2. Giao diện tương tác người dung:** 46](#_Toc183593744)

[**4.2.1. Mục đích của giao diện:** 46](#_Toc183593745)

[**4.2.2. Thiết kế giao diện chính:** 46](#_Toc183593746)

[**4.2.3. Thiết kế giao diện phần Feature** 53](#_Toc183593747)

[**4.2.4. Thiết kế giao diện phần** **Data\_cleaning\_normalization:** 65](#_Toc183593748)

[**PHẦN KẾT LUẬN** 67](#_Toc183593749)

[**PHẦN TÀI LIỆU THAM KHẢO** 69](#_Toc183593750)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Mô tả vài cột trong dữ liệu. 17](#_Toc183546311)

[Hình 2:Code import các thư viện cần thiết cho việc phân tích và xây dự mô hình 19](#_Toc183546312)

[Hình 3: Code hàm để tải dử liệu 19](#_Toc183546313)

[Hình 4: Code xóa các hang bị thiếu dữ liệu 20](#_Toc183546314)

[Hình 5: Code hàm cần điền dữ liệu mặc định 20](#_Toc183546315)

[Hình 6: Code hàm chuyển đổi kiểu dữ liệu 20](#_Toc183546316)

[Hình 7: Code hàm làm sạch cột dữ liệu 21](#_Toc183546317)

[Hình 8: Code hàm thay thế giá trị không hợp lệ 21](#_Toc183546318)

[Hình 9: Code hàm xóa dữ liệu trùng lắp 21](#_Toc183546319)

[Hình 10: Code hàm lưu dữ liệu làm sạch 22](#_Toc183546320)

[Hình 11: Code tải dữ liệu 22](#_Toc183546321)

[Hình 12: Code chuẩn hóa tên văn bản 23](#_Toc183546322)

[Hình 13: Code chuẩn hóa tên cột 23](#_Toc183546323)

[Hình 14: Code chuẩn hóa dữ liệu 23](#_Toc183546324)

[Hình 15: Code chuẩn hóa toàn bộ dữ liệu 23](#_Toc183546325)

[Hình 16: Code hàm chính để làm sạch dữ liệu và lưu vào kết quả đã cho 24](#_Toc183546326)

[Hình 17: Code thực hiện chức năng Read 25](#_Toc183546327)

[Hình 18: Code tạo dữ liệu mới 26](#_Toc183546328)

[Hình 19: Code kiểm tra và cập nhật File CSV 26](#_Toc183546329)

[Hình 20: Tổng quan code thực hiện cư chế UPDATE 28](#_Toc183546330)

[Hình 21: Code kiểm tra tệp CSV 29](#_Toc183546331)

[Hình 22: Code tìm kiếm model\_name cần cập nhật 29](#_Toc183546332)

[Hình 23: Code hiển thị giao diện nhập liệu 29](#_Toc183546333)

[Hình 24: Code lưu thay đổi 30](#_Toc183546334)

[Hình 25 :Tổng quan code thực hiện cơ chế Delete 31](#_Toc183546335)

[Hình 26: Code kiểm tra tệp CSV 31](#_Toc183546336)

[Hình 27: Code kiểm tra thuộc tính 32](#_Toc183546337)

[Hình 28: Code hàm sắp xếp dữ liệu 33](#_Toc183546338)

[Hình 29: Code hàm tìm kiếm dữ liệu 34](#_Toc183546339)

[Hình 30: Code các thư viện cần thiết cho việc phân tích dữ liệu 34](#_Toc183546340)

[Hình 31 :Code đọc dữ liệu file và hiển thị thông tin cơ bản 35](#_Toc183546341)

[Hình 32: Hiển thị danh sách 35](#_Toc183546342)

[Hình 33: Code kiểm tra giá trị bị thiếu trong từng cột và kiểm tra các bản ghi trùng lặp 36](#_Toc183546343)

[Hình 34: Xuất ra các giá trị bị thiếu trong từng cột và các bản ghi trùng lặp 36](#_Toc183546344)

[Hình 35: Code biểu đồ sản phẩm và giá cao nhất theo công ty 37](#_Toc183546345)

[Hình 36: Biểu đồ sản phẩm và giá cao nhất theo công ty 38](#_Toc183546346)

[Hình 37: Code biểu đồ kích thước màn hình trung bình theo công ty 38](#_Toc183546347)

[Hình 38: Biểu đồ kích thước màn hình trung bình theo công ty 39](#_Toc183546348)

[Hình 39: Code thể hiện biểu đồ kích thước màn hình 39](#_Toc183546349)

[Hình 40: Code tạo cửa sổ chính, frame chính và tiêu đề 47](#_Toc183546350)

[Hình 41: Code treeview hiển thị dữ liệu 48](#_Toc183546351)

[Hình 42: Biến toàn cục cho phân trang 49](#_Toc183546352)

[Hình 43: Hàm hiển thị dữ liệu theo trang 49](#_Toc183546353)

[Hình 44: Hàm đọc dữ liệu và thiết lập phân trang 50](#_Toc183546354)

[Hình 45: Hàm chuyển trang trước 50](#_Toc183546355)

[Hình 46: Hàm chuyển trang sau 51](#_Toc183546356)

[Hình 47: Hàm cập nhật trạng thái phân trang 51](#_Toc183546357)

[Hình 48: Khung chứa nút phân trang 51](#_Toc183546358)

[Hình 49: Nhãn hiển thị số trang và nút điều khiển 52](#_Toc183546359)

[Hình 50: Hàm hiển thị kết quả tìm kiếm 53](#_Toc183546360)

[Hình 51: Khu vực tìm kiếm 54](#_Toc183546361)

[Hình 52: Hàm mở cửa sổ tạo bản ghi mới 55](#_Toc183546362)

[Hình 53: Hàm xử lý việc them bản ghi 56](#_Toc183546363)

[Hình 54: Tạo cửa sổ nhập liệu 57](#_Toc183546364)

[Hình 55: Tạo các trường nhập liệu 57](#_Toc183546365)

[Hình 56: Nút thêm bản ghi 58](#_Toc183546366)

[Hình 57: Nút tạo Laptop mới 58](#_Toc183546367)

[Hình 58: Nàm mở giao diện cập nhật 59](#_Toc183546368)

[Hình 59: Tạo nút cập nhật 59](#_Toc183546369)

[Hình 60: Hàm xử lí xóa laptop theo tên product 60](#_Toc183546370)

[Hình 61: Khung chứa các thành phần liên quan đến xóa 61](#_Toc183546371)

[Hình 62: Label hường dẫn người dung 61](#_Toc183546372)

[Hình 63: Entry để nhập tên sản phẩm 61](#_Toc183546373)

[Hình 64: Nút xóa 62](#_Toc183546374)

[Hình 65: Hàm hiển thị dữ liệu sau khi sắp xếp 62](#_Toc183546375)

[Hình 66: khu vực sắp xếp 63](#_Toc183546376)

[Hình 67: Cửa sổ biểu đồ 64](#_Toc183546377)

[Hình 68: Các nút chọn biểu đồ 64](#_Toc183546378)

[Hình 69: Hàm làm sạch 64](#_Toc183546379)

[Hình 70: Hàm chuẩn hóa 64](#_Toc183546380)

[Hình 71: Khung chứa nút xử lý dữ liệu 65](#_Toc183546381)

[Hình 72: Nút làm sạch dữ liệu 65](#_Toc183546382)

[Hình 73: Nút chuẩn hóa dữ liệu 65](#_Toc183546383)

**KẾ HOẠCH PHÂN CÔNG NHIỆM VỤ THỰC HIỆN ĐỀ TÀI**

**CUỐI KỲ MÔN LẬP TRÌNH PYTHON**

**HỌC KỲ I NĂM HỌC 2023-2024**

**1. Mã lớp môn học: IPPA233277**

**2. Giảng viên hướng dẫn: ThS. Trần Quang Khải**

**3. Tên đề tài: Dự đoán giá máy tính xách tay**

**4. Bảng phân công nhiệm vụ:**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **Nội dung thực hiện** |
| Nguyễn Hoàng Anh | * Code xây dựng mô hình dữ liệu (interface). * Code tìm kiếm dữ liệu (Search). * Xây dựng báo cáo Phần Nội Dung(chương 4) và Phần Kết Luận |
| Nguyễn Đình Huy | * Tìm hiểu và code làm sạch, chuẩn hóa dữ liệu. * Code sort dữ liệu. * Xây dựng báo cáo Phần Mở Đầu, Phần Nội Dung (chương 1,2) * Tổng hợp và hoàn thiện báo cáo. |
| Lưu Thái Thanh Lâm | * Code xử lí dữ liệu và vẽ biểu đồ (CRUD,vẽ biểu đồ ). * Xây dưng báo cáo Phần nội dung (chương 3). * Soạn slide thuyết trình báo cáo đồ án |

# **LỜI CẢM ƠN**

Để hoàn thành đề tài này, trước tiên chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Trần Quang Khải - nghiên cứu viên môn Lập trình Python của lớp IPPA233277. Thầy không chỉ là người truyền đạt kiến ​​thức một cách tận tâm mà còn luôn theo sát, hướng dẫn chúng tôi trong suốt quá trình học tập và thực hiện bản đồ cuối cùng. Những lời khuyên chỉ bảo, năng động và định hướng của thầy chính là nguồn động lực lớn giúp chúng em vượt qua những khó khăn, thử thách để hoàn thiện đề tài

Trong suốt quá trình học tập và nghiên cứu, thầy đã dành nhiều thời gian quý báu để giải đáp những thắc mắc, cung cấp những tài liệu hữu ích và chia sẻ kinh nghiệm thực tế. Không chỉ giúp chúng em phát triển kỹ năng lập trình, thầy còn truyền cảm hứng về tinh thần học hỏi không ngừng, giúp chúng em nhìn nhận vấn đề về một cách sáng tạo và toàn diện hơn.

Chúng em cũng xin cảm ơn thầy trong công việc tạo điều kiện để họ có môi trường học tập và thực hành hiệu quả. Từ những bài giải thưởng hướng dẫn các buổi học, thầy luôn khuyến khích chúng em tìm kiếm, sáng tạo và tự mình giải quyết các vấn đề. Điều này không chỉ giúp chúng em nâng cao kỹ năng chuyên môn mà còn rèn luyện tư duy logic, tính dũng chiến và trách nhiệm trong công việc.

Mặc dù chúng em đã nỗ lực hết sức, song với kiến ​​thức và kinh nghiệm còn hạn chế, bản báo cáo của chúng em khó tránh khỏi những sai sót và thiếu hụt. Vì vậy, chúng em rất mong nhận được những nhận xét, lời khuyên và chỉ dẫn từ thầy để có thể hoàn thiện hơn không chỉ trong môn học này mà còn trong những món ăn đường tiếp theo trên con đường học vấn và sự thật

Một lần nữa, chúng em xin gửi lời cảm ơn sâu sắc đến thầy vì sự quan tâm, chỉ dạy và hỗ trợ hết mình trong suốt quá trình hành động vừa phải. Xin cảm ơn thầy, chúng em đã học được rất nhiều bài học quý giá và tin rằng đây sẽ là nền tảng vững chắc để chúng tôi tiến xa hơn trên con đường tương lai.

Trân trọng cảm ơn.

# **PHẦN 1: MỞ ĐẦU**

1. **Lý do chọn đề tài**

Trong thời đại công nghệ 4.0, laptop đã trở thành công cụ thiết yếu trong học tập, làm việc và giải trí. Tuy nhiên, thị trường máy tính xách tay ngày càng đa dạng về mẫu mã, giá cả và cấu hình, gây khó khăn cho người tiêu dùng khi lựa chọn sản phẩm phù hợp với nhu cầu và ngân sách. Việc phân tích các yếu tố ảnh hưởng đến giá laptop và xây dựng công cụ hỗ trợ quyết định việc mua sắm trở nên cần thiết và có ý kiến

Bên cạnh đó, sự phát triển của các kỹ thuật phân tích dữ liệu và học máy mang lại cơ sở ứng dụng công nghệ để giải quyết các vấn đề thực tiễn. Đề tài này không chỉ giúp khai thác dữ liệu để hiểu rõ các yếu tố ảnh hưởng đến giá máy tính xách tay mà còn tạo cơ hội cho nhóm họ áp dụng kiến ​​thức vào thực tế, rèn luyện kỹ năng phân tích, lập trình và làm việc nhóm

Chọn đề tài này, chúng em mong muốn cung cấp thông tin hữu ích cho người tiêu dùng, đồng thời nâng cao kỹ năng chuyên môn để chuẩn bị tốt hơn cho các công thức học tập và nghề nghiệp trong tương lai.

1. **Mục tiêu đề tài**

* **Phân tích dữ liệu**: Tìm hiểu các yếu tố ảnh hưởng đến giá laptop, bao gồm cấu hình, thương hiệu, kích thước màn hình, và các yếu tố khác trong bộ dữ liệu đã cung cấp.
* **Xây dựng mô hình dự đoán**: Ứng dụng các kỹ thuật lập trình để xây dựng mô hình dự đoán giá laptop dựa trên các thông số kỹ thuật.
* **Hỗ trợ quyết định mua sắm**: Cung cấp các phân tích và gợi ý hữu ích, giúp người dùng chọn được laptop phù hợp với nhu cầu và ngân sách.
* **Kỹ năng thực hành**: Áp dụng các kỹ thuật lập trình, phân tích dữ liệu trong thực tế, đồng thời nâng cao khả năng làm việc nhóm và giải quyết vấn đề.

1. **Đối tượng và phạm vi nghiên cứu**

* **Đối tượng nghiên cứu**: Các yếu tố ảnh hưởng đến giá laptop và mối tương quan giữa các thông số kỹ thuật của laptop với giá bán.
* **Phạm vi nghiên cứu**:

+ Bộ dữ liệu được sử dụng trong đề tài là tập dữ liệu đã cung cấp (laptop\_price - dataset.csv).

+ Đề tài tập trung phân tích các thông số như thương hiệu, bộ vi xử lý, RAM, kích thước màn hình, ổ cứng, và hệ điều hành.

+ Các kỹ thuật phân tích dữ liệu và dự đoán được thực hiện bằng các ngôn ngữ lập trình và thư viện như Python, Pandas, Scikit-learn và Matplotlib.

+ Phạm vi nghiên cứu chủ yếu nhắm đến thị trường laptop trong thời gian cụ thể mà dữ liệu đã cung cấp.

# **PHẦN 2. NỘI DUNG**

# **CHƯƠNG 1: TÓM TẮT CƠ SỞ LÝ THUYẾT**

* 1. **Giới thiệu về lập trình python**

Python là một ngôn ngữ lập trình bậc cao, được thiết kế để dễ đọc, dễ học và mạnh mẽ trong ứng dụng thực tế. Dưới đây là những định nghĩa khái quát về Python:

**"Python is an interpreted, high-level, general-purpose programming language. Its design philosophy emphasizes code readability with its notable use of significant indentation."**  
(Python là một ngôn ngữ lập trình thông dịch, bậc cao và đa năng. Triết lý thiết kế của Python nhấn mạnh sự dễ đọc của mã nguồn nhờ việc sử dụng thụt lề quan trọng.)  
— Guido van Rossum, 1991

"Python is the Swiss Army knife of programming languages, versatile enough for anything from web development to machine learning."

(Python là “dao đa năng Thụy Sĩ” trong các ngôn ngữ lập trình, đủ linh hoạt để sử dụng từ phát triển web đến học máy.)

— Lập trình viên cộng đồng

Python đã và đang trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới nhờ tính linh hoạt, cú pháp dễ học và sự hỗ trợ mạnh mẽ từ cộng đồng lập trình viên trên toàn cầu.

## **1.2. Giới thiệu về các thư viện thường được sử dụng trong lập trình python và môi trường lập trình**

Python sở hữu một hệ sinh thái thư viện phong phú, hỗ trợ nhiều lĩnh vực khác nhau từ phân tích dữ liệu, học máy, đến phát triển web và xử lý ảnh. Dưới đây là một số thư viện phổ biến:

*Thư viện cho phân tích dữ liệu và xử lý số liệu:*

* **NumPy**:  
  Thư viện mạnh mẽ dành cho tính toán khoa học và xử lý mảng đa chiều. NumPy cung cấp các công cụ để thực hiện các phép toán đại số tuyến tính, ma trận và thống kê cơ bản.
* **Pandas**:  
  Thư viện hỗ trợ thao tác và phân tích dữ liệu dạng bảng (dataframe). Pandas rất hữu ích trong việc xử lý dữ liệu lớn, lọc dữ liệu và biến đổi dữ liệu.
* **Matplotlib**:  
  Thư viện trực quan hóa dữ liệu, giúp tạo các biểu đồ như line chart, bar chart, scatter plot, và histogram.
* **Seaborn**:  
  Mở rộng từ Matplotlib, Seaborn cung cấp các biểu đồ đẹp mắt hơn với cú pháp dễ sử dụng, tập trung vào trực quan hóa thống kê.

*Thư viện cho học máy và trí tuệ nhân tạo:*

* **Scikit-learn**:  
  Một thư viện toàn diện cho học máy, hỗ trợ các thuật toán như phân loại, hồi quy, và cụm (clustering).
* **Keras**:  
  Giao diện cấp cao của TensorFlow, giúp việc xây dựng mô hình học sâu trở nên đơn giản và thân thiện hơn.
* **TensorFlow và PyTorch:**

Hai thư viện mạnh mẽ dành cho học sâu (deep learning). TensorFlow hỗ trợ xây dựng các mô hình mạng nơ-ron với quy mô lớn, trong khi PyTorch dễ dàng thử nghiệm và triển khai nhanh.

**Môi trường lập trình Python phổ biến**

*Các IDE (Integrated Development Environment):*

1. **PyCharm**:
   * Một IDE mạnh mẽ dành cho Python, cung cấp các công cụ như tự động hoàn thành mã, gỡ lỗi (debugging), và tích hợp với các thư viện phổ biến.
   * Phù hợp với các dự án lớn và phức tạp.
2. **Visual Studio Code (VS Code):**
   * Một trình soạn thảo mã nguồn mở, hỗ trợ nhiều ngôn ngữ, trong đó có Python.
   * VS Code nổi bật với các tiện ích mở rộng (extensions) như **Python Extension**, giúp hỗ trợ tự động hoàn thành, quản lý môi trường ảo, và chạy mã trực tiếp.
3. **Jupyter Notebook:**
   * Công cụ phổ biến trong lĩnh vực phân tích dữ liệu và học máy. Jupyter cho phép viết mã, trực quan hóa dữ liệu và ghi chú trên cùng một giao diện.
4. **Spyder:**
   * IDE dành riêng cho khoa học dữ liệu, tích hợp tốt với các thư viện như NumPy, Pandas, và Matplotlib.

*Môi trường ảo (Virtual Environment):*

* **Virtualenv:**   
  Dùng để tạo các môi trường độc lập, giúp quản lý các thư viện và phiên bản Python mà không ảnh hưởng đến hệ thống chính.
* **Conda**:  
  Công cụ mạnh mẽ để quản lý môi trường và thư viện, đặc biệt hữu ích khi làm việc với dữ liệu lớn và học máy.

*Nền tảng đám mây:*

1. **Google Colab:**
   * Một dịch vụ miễn phí của Google, cho phép chạy mã Python trên máy chủ đám mây với tài nguyên như GPU và TPU.
   * Rất hữu ích cho học máy và xử lý dữ liệu lớn.
2. **Kaggle Notebook:**
   * Một nền tảng trực tuyến hỗ trợ phân tích dữ liệu và xây dựng mô hình học máy.

Python kết hợp giữa hệ sinh thái thư viện đa dạng và các môi trường lập trình linh hoạt, giúp lập trình viên tối ưu hóa quy trình làm việc, từ nghiên cứu khoa học đến ứng dụng thực tiễn.

# **CHƯƠNG 2: MÔ TẢ TẬP DỮ LIỆU** **GIÁ MÁY TÍNH XÁCH TAY**

## **2.1. Các tập dữ liệu cần thu thập**

Trong đề tài này, nhóm chúng em đã thu thập thông tin chi tiết về các sản phẩm máy tính xách tay, bao gồm các đặc tính kỹ thuật và giá cả thực tế, từ một tệp CSV. Cụ thể như sau:

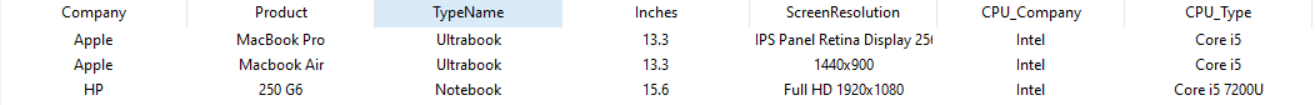
* File dataset.csv:
* Được sử dụng để huấn luyện và kiểm định mô hình dự đoán giá máy tính xách tay.
* Bao gồm các đặc trưng đầu vào (input features) như tên hãng sản xuất, loại CPU, RAM, dung lượng lưu trữ,... và giá thực tế (target: price).
* Dữ liệu này phù hợp để xây dựng mô hình học máy nhờ số lượng lớn các dòng dữ liệu.

Mô tả chi tiết:

|  |  |
| --- | --- |
| Thông tin | Mô tả |
| Tên tập dữ liệu | Laptop\_price – dataset.csv |
| Kích thước | 14 cột và 1303 dòng |
| Loại dữ liệu | Số nguyên, số thực, chuỗi ký tự |

Bảng dưới đây mô tả một số cột chính trong tập dữ liệu:

|  |  |
| --- | --- |
| Cột | Ý nghĩa |
| Company | Tên hãng sản xuất máy tính xách tay. |
| TypeName | Loại máy tính (Ultrabook, Gaming, Notebook,…). |
| Cpu | Thông tin bộ vi xử lý (CPU). |
| Ram | Dung lượng Ram. |
| Memory | Dung lượng và loại ổ cứng (SSD/HDD). |
| Gpu | Card đồ họa được trang bị. |
| Price | Giá thực tế của máy tính xách tay (đơn vị: Euro). |

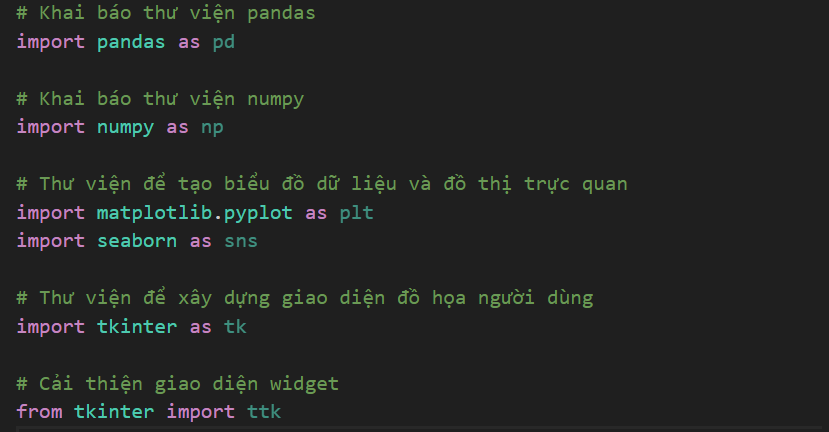


Hình 1: Mô tả vài cột trong dữ liệu.

# **CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU GIÁ MÁY TÍNH (LAPTOP\_PRICE)**

## **3.1. Khởi tạo dữ liệu**

**Khởi tạo các thư viện cần thiết**

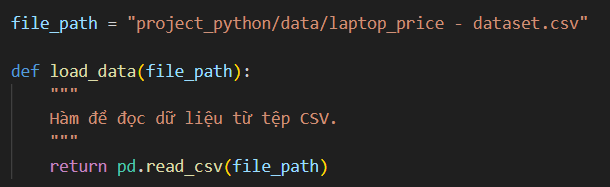


Hình 2:Code import các thư viện cần thiết cho việc phân tích và xây dự mô hình dự đoán Laptop prices.

## **3.2. Chuẩn hóa dữ liệu và làm sạch dữ liệu(Data normalizer and cleaning)**

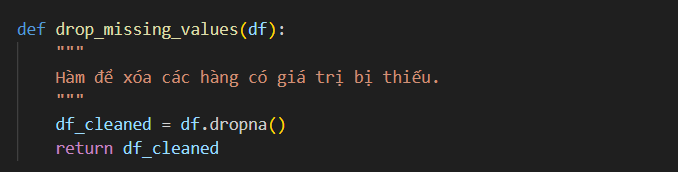
### **3.2.1. Làm sạch dữ liệu(Data Cleaning)**

Chức năng làm sạch dữ liệu (“Data Cleaning”) đảm bảo dữ liệu đồng nhất, không bị thiếu và phù hợp cho các thao tác xử lý sau này. Quá trình bao gồm các bước sau:



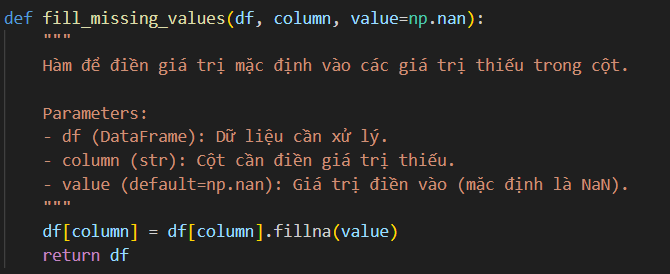
Hình 3: Code hàm để tải dử liệu

Sử dụng hàm `load\_data(file\_path)` để đọc tệp CSV và tải vào DataFrame.



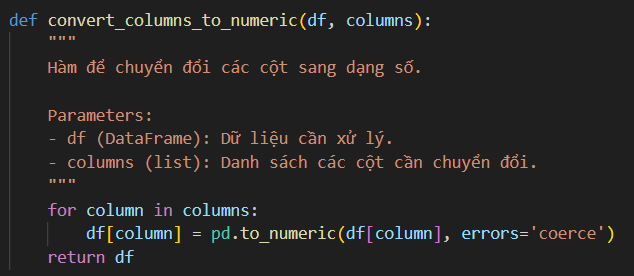
Hình 4: Code xóa các hang bị thiếu dữ liệu

Hàm “drop\_missing\_values(df)” xóa các hàng bị thiếu dữ liệu để giảm thiểu lỗi trong xử lý.



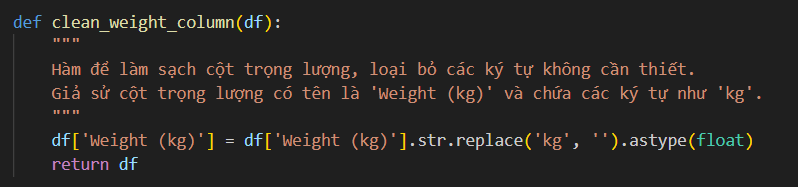
Hình 5: Code hàm cần điền dữ liệu mặc định

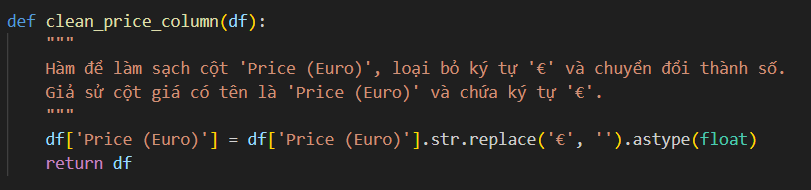
Hàm “fill\_missing\_values(df, column, value)” cho phép điền giá trị mặc định vào các ô bị thiếu trong cột chỉ định.



Hình 6: Code hàm chuyển đổi kiểu dữ liệu

Hàm “convert\_columns\_to\_numeric(df, columns)” chỗ hỗ chuyển các cột sang dạng số.

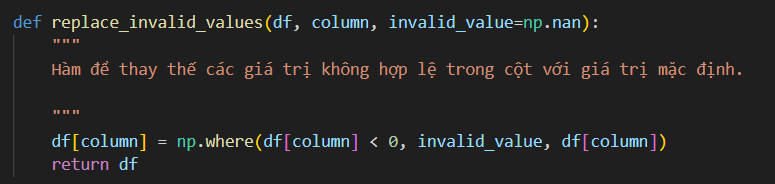




Hình 7: Code hàm làm sạch cột dữ liệu

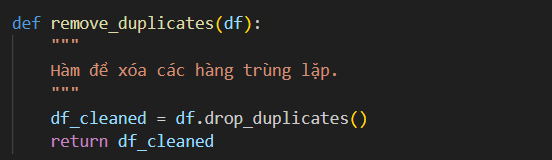
Hàm “clean\_weight\_column(df)” làm sạch cột "Weight (kg)", loại bỏ các ký tự như "kg" và chuyển sang số thực.

Hàm “clean\_price\_column(df)” làm sạch cột "Price (Euro)", loại bỏ các ký tự như "€" và chuyển sang dạng số.



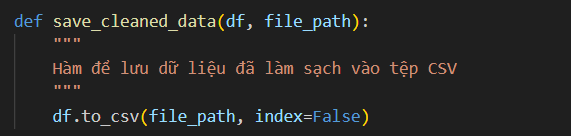
Hình 8: Code hàm thay thế giá trị không hợp lệ

Hàm “replace\_invalid\_values(df, column, invalid\_value)" thay thế các giá trị không hợp lệ trong cột với giá trị mặc định (vd: thay thế giá trị âm).



Hình 9: Code hàm xóa dữ liệu trùng lắp

Hàm “remove\_duplicates(df)” xóa các hàng trùng lắp trong DataFrame.



Hình 10: Code hàm lưu dữ liệu làm sạch

Hàm “save\_cleaned\_data(df, file\_path)” ghi DataFrame đã làm sạch vào tệp CSV.

**Kết quả**

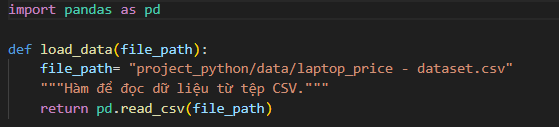
- Tất cả các khuyết điểm trong dữ liệu đã được khắc phục như: giá trị bị thiếu, giá trị không hợp lệ, và hàng trùng lắp.

- Dữ liệu được chuẩn bị sẵn sàng cho các thao tác xử lý tiếp theo.

### **3.2.2. Chuẩn hóa dữ liệu(Data normalizer)**

Chức năng chuẩn hóa dữ liệu được sử dụng để làm sạch và chuẩn hóa tập dữ liệu nhằm đảm bảo tính nhất quán trước khi thực hiện các thao tác khác. Quá trình chuẩn hóa bao gồm các bước sau:

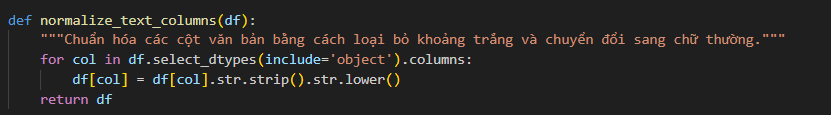
**Cơ chế hoạt động**

****

Hình 11: Code tải dữ liệu

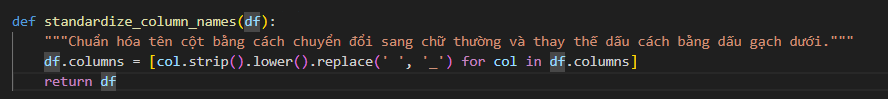
Sử dụng hàm “load\_data(file\_path)” để tải dữ liệu từ tệp CSV vào DataFrame.

Kiểm tra đường dẫn tệp và đảm bảo tệp tồn tại.



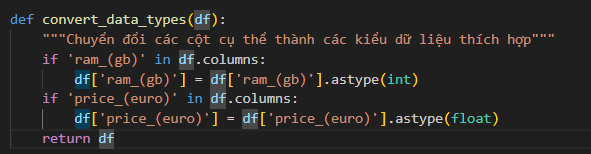
Hình 12: Code chuẩn hóa tên văn bản

Hàm “normalize\_text\_columns(df)” loại bỏ khoảng trắng thừa và chuyển đổi văn bản của các cột dạng chuỗi sang chữ thường để đồng nhất dữ liệu.



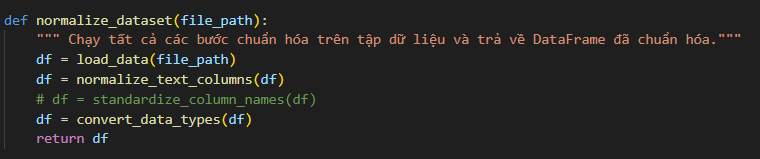
Hình 13: Code chuẩn hóa tên cột

Hàm “standardize\_column\_names(df)” chuyển đổi tên cột thành chữ thường và thay thế dấu cách bằng dấu gạch dưới (`\_`) để dễ xử lý hơn.



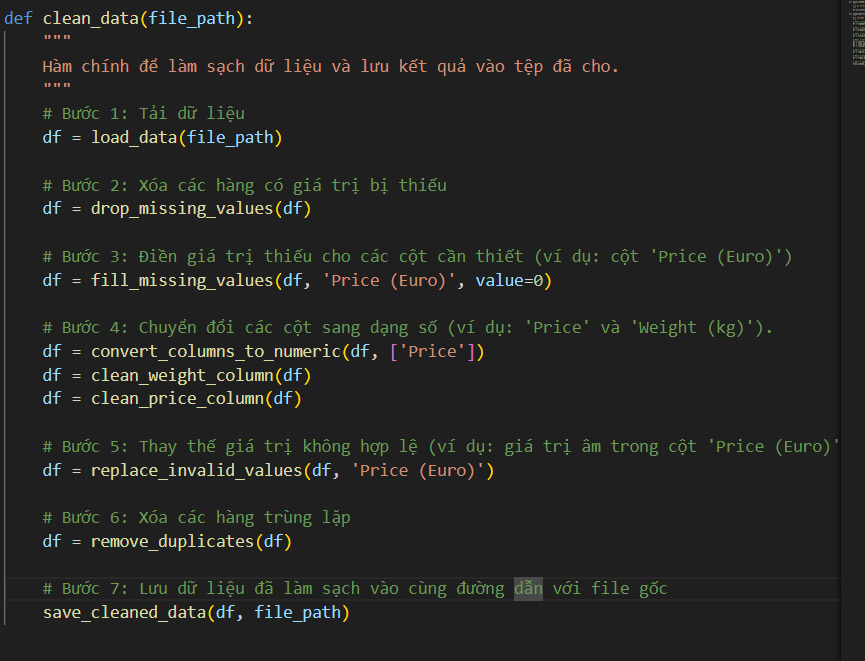
Hình 14: Code chuẩn hóa dữ liệu

Hàm “convert\_data\_types(df)” chuyển đổi các cột dữ liệu như `RAM (GB)` và `Price (Euro)` sang kiểu số để thuận tiện cho các tính toán và xử lý.



Hình 15: Code chuẩn hóa toàn bộ dữ liệu

Hàm “normalize\_dataset(file\_path)” thực hiện toàn bộ các bước trên và trả về DataFrame đã chuẩn hóa.



Hình 16: Code hàm chính để làm sạch dữ liệu và lưu vào kết quả đã cho

**Kết quả**

- Đảm bảo dữ liệu nhất quán và chuẩn hóa trước khi sử dụng.

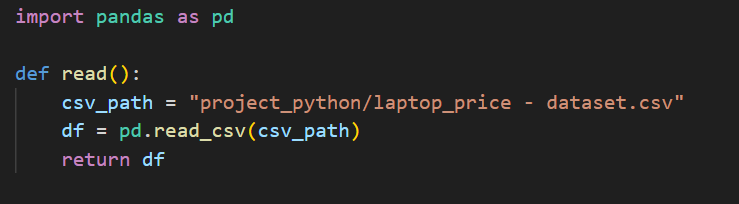
- Các bước chuẩn hóa được thực hiện tuần tự: tải dữ liệu, làm sạch văn bản, chuẩn hóa tên cột, và chuyển đổi kiểu dữ liệu.

- Tăng tính hiệu quả và độ chính xác cho các thao tác CRUD tiếp theo.

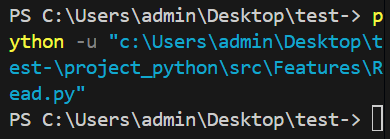
## **3.3. Thực hiện các cơ chế CRUD**

### **3.3.1. Cơ chế READ**

Chức năng READ được sử dụng để đọc dữ liệu từ một tệp CSV. Dữ liệu sẽ được tải vào một DataFrame của Pandas để sử dụng trong các thao tác khác.



Hình 17: Code thực hiện chức năng Read

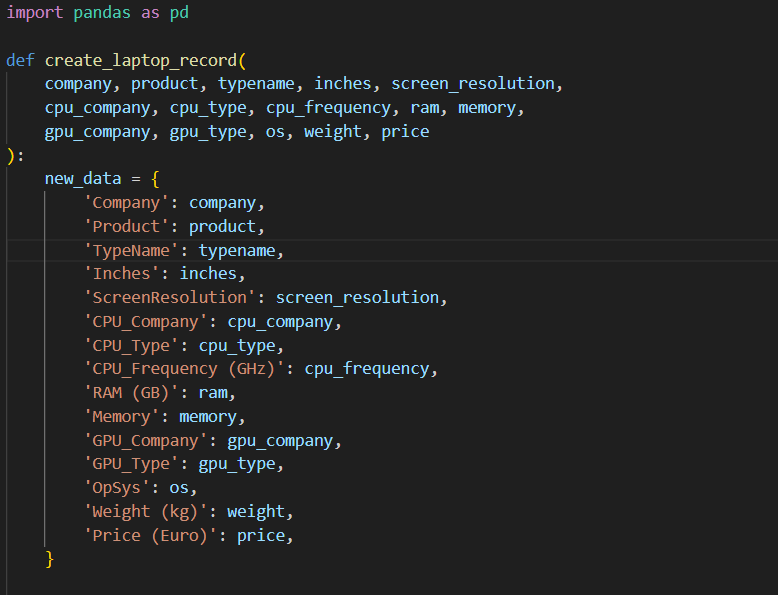


Kết quả: Đọc dữ liệu từ file CSV “laptop\_price - dataset.csv” thành công. Dữ liệu được lưu trong DataFrame “df”

### **3.3.2. Cơ chế CREATE**

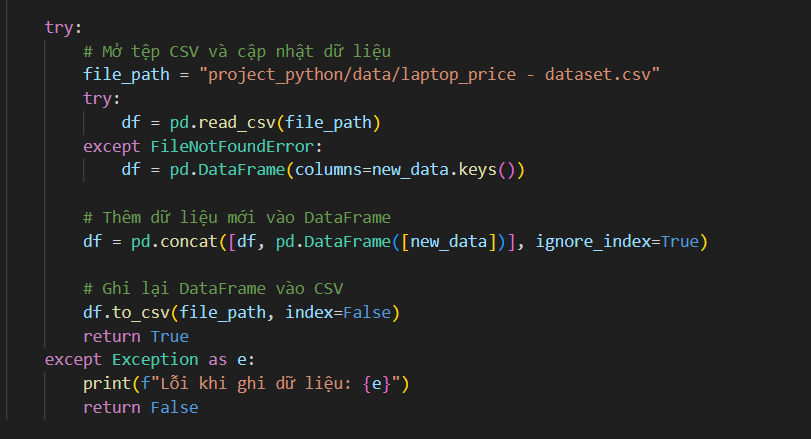
Chức năng CREATE cho phép thêm một bản ghi mới vào tập dữ liệu laptop. Nếu tệp CSV không tồn tại, một tệp mới sẽ được tạo.

**Cơ chế hoạt động**

****

Hình 18: Code tạo dữ liệu mới

Tạo dữ liệu mới: Hệ thống nhận các tham số từ người dùng bao gồm tên công ty, tên model, loại laptop, các thông tin kỹ thuật (RAM, CPU, GPU, v.v.), và giá.



Hình 19: Code kiểm tra và cập nhật File CSV

* Kiểm tra File CSV:

Nếu tệp CSV đã tồn tại, dữ liệu sẽ được tải lên.

Nếu tệp CSV không tồn tại, một DataFrame mới sẽ được tạo với các cột tương ứng.

* Thêm dữ liệu mới:

Dòng dữ liệu mới được gắn vào DataFrame hiện tại.

* Ghi lên File:

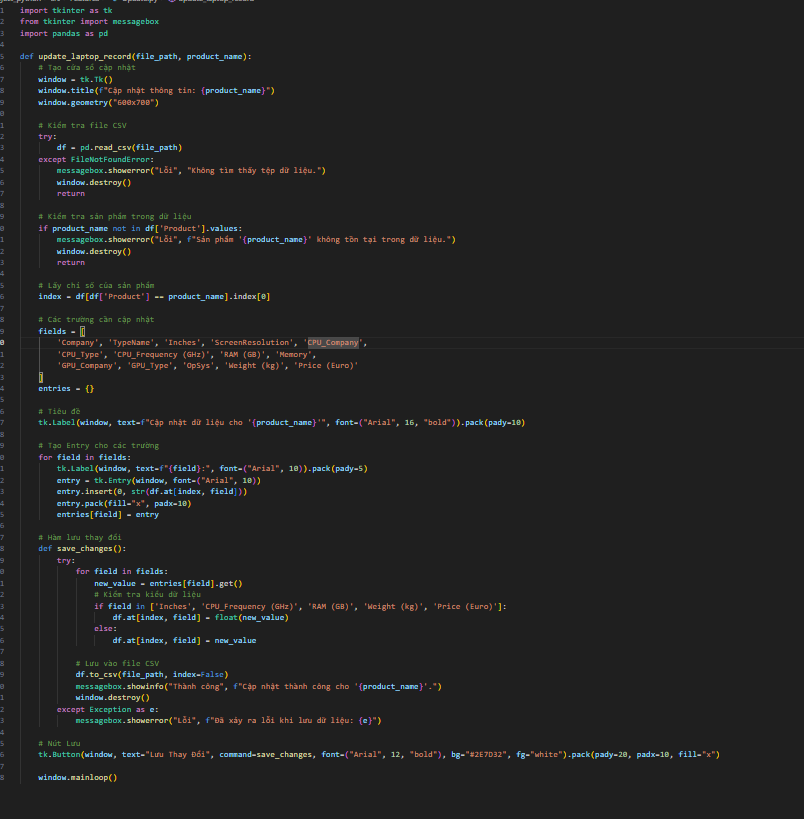
DataFrame được ghi lại và lưu trữ trong tệp CSV.

Nếu quá trình ghi dữ liệu gặp lỗi, thông báo sẽ được hiển thị để người dùng xử lý.

* Thêm bản ghi mới thành công vào file “laptop\_price - dataset.csv”, đồng thời tệp được cập nhật hoặc tạo mới nếu không tồn tại.

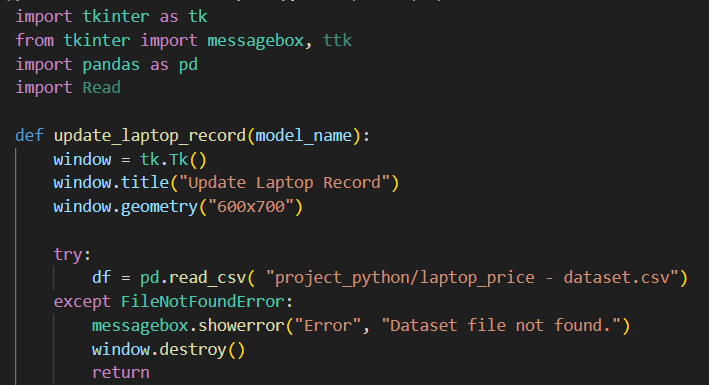
### **3.3.3. Cơ chế UPDATE**

Chức năng UPDATE cho phép cập nhật thông tin của một bản ghi dựa trên `Product`. Giao diện người dùng được xây dựng bằng `tkinter` để nhập các giá trị mới.



Hình 20: Tổng quan code thực hiện cư chế UPDATE

**Cơ chế hoạt động:**

****

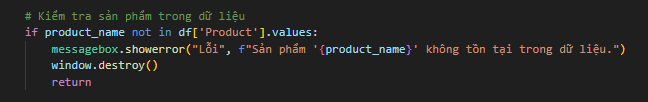
Hình 21: Code kiểm tra tệp CSV

Khởi tạo hàm “update\_laptop\_record(model\_name)

Thực hiện kiểm tra tệp CSV

Hệ thống mở tệp CSV chứa dữ liệu laptop.

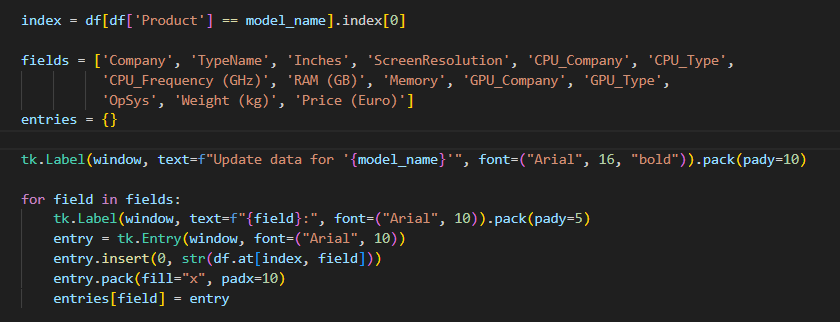
Nếu tệp không tồn tại, giao diện sẽ hiện thông báo lỗi và dừng thực thi.



Hình 22: Code tìm kiếm model\_name cần cập nhật

Kiểm tra xem product\_name có tồn tại trong dữ liệu hay không.

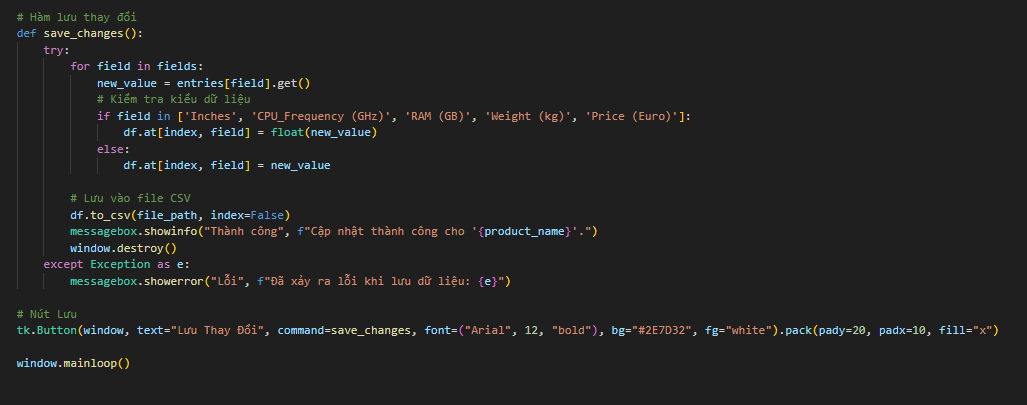
Nếu product\_name không tìm thấy, giao diện hiện thông báo lỗi



Hình 23: Code hiển thị giao diện nhập liệu

Giao diện được xây dựng bằng “Tinker” hiện danh sách các trường dữ liệu cần cập nhật.

Người dùng có thể nhập thông tin mới cho từng trường.



Hình 24: Code lưu thay đổi

Khi nhấn "Update", hệ thống lưu các thay đổi vào DataFrame.

Tính toán lại các chỉ số như “Price\_per\_GB\_RAM”, “Price\_per\_Inch”,…(nếu có)

* Ghi lên File CSV

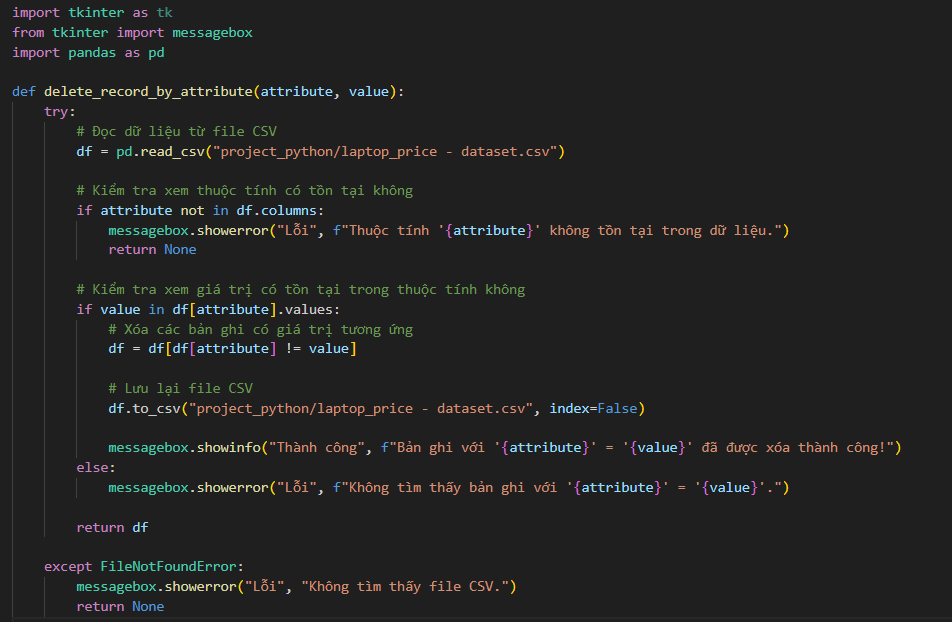
DataFrame sau khi cập nhật sẽ được ghi lại tệp CSV.

Thông báo thành công khi cập nhật xong.

* Cập nhật thành công thông tin của model cần chỉnh sửa trong tệp CSV.

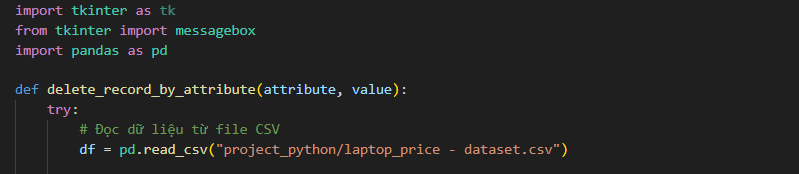
### **3.3.4. Cơ chế DELETE**

Chức năng DELETE cho phép xóa bản ghi khỏi tập dữ liệu dựa trên một thuộc tính và giá trị cụ thể.



Hình 25 :Tổng quan code thực hiện cơ chế Delete

**Cơ chế hoạt động:**

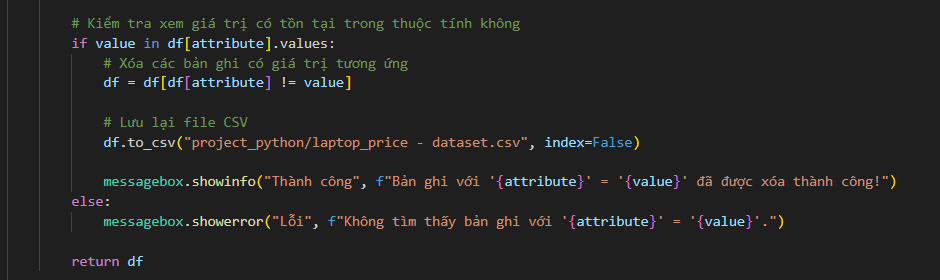
****

Hình 26: Code kiểm tra tệp CSV

Xây dựng hàm "delete\_record\_by\_attribute” với hai thuộc tính attribute và value

Hệ thống mở tệp CSV chứa dữ liệu laptop.

Nếu tệp CSV không tồn tại, giao diện hiện thông báo lỗi và dừng thực thi.



Hình 27: Code kiểm tra thuộc tính

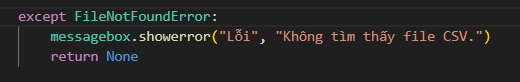
* Xác minh xem thuộc tính được cung cấp có tồn tại trong tập dữ liệu hay không. Nếu thuộc tính không tồn tại, giao diện hiện thông báo lỗi.
* Hệ thống tìm bản ghi có giá trị được cung cấp trong thuộc tính tương ứng.

Nếu không tìm thấy bản ghi nào phù hợp, giao diện hiện thông báo lỗi.

* Loại bỏ tất cả các bản ghi có giá trị tương ứng trong DataFrame.

DataFrame sau khi xóa bản ghi được ghi lại tệp CSV.

Nếu quá trình ghi gặp lỗi, giao diện sẽ hiện thông báo.

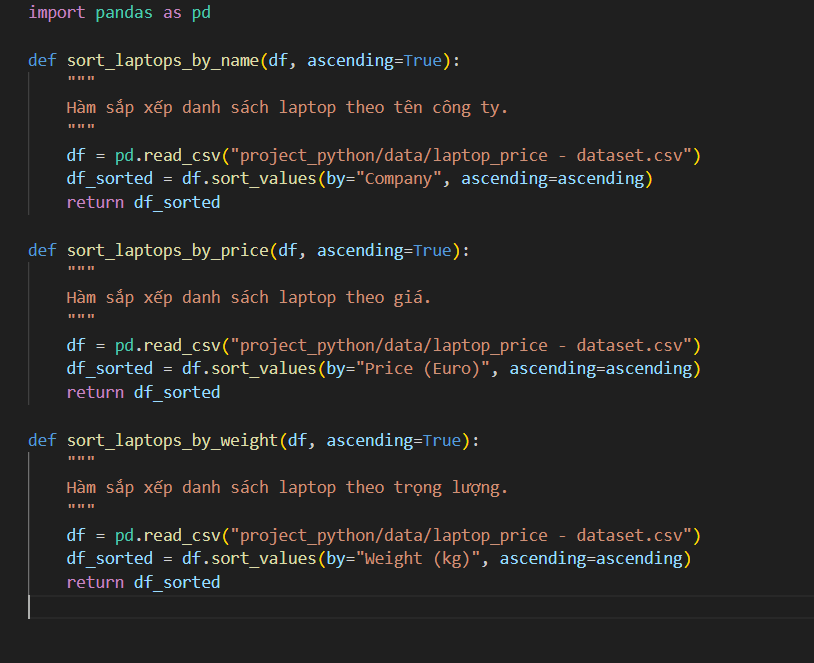


* Xóa bản ghi thành công khỏi tệp CSV dựa trên giá trị thuộc tính. Nếu không thực hiện được cơ chế DELETE, thông báo rõ ràng cho người dùng trong trường hợp lỗi hoặc không tìm thấy dữ liệu.

### **3.3.5. Cơ chế sắp xếp dữ liệu(Sort)**

Chức năng sắp xếp dữ liệu giúp người dùng sắp xếp danh sách laptop dựa trên các tiêu chí như tên công ty, giá bán, hoặc trọng lượng. Quá trình sắp xếp bao gồm các bước sau:

Dữ liệu từ tệp CSV được tải vào DataFrame sử dụng hàm “pd.read\_csv()”.



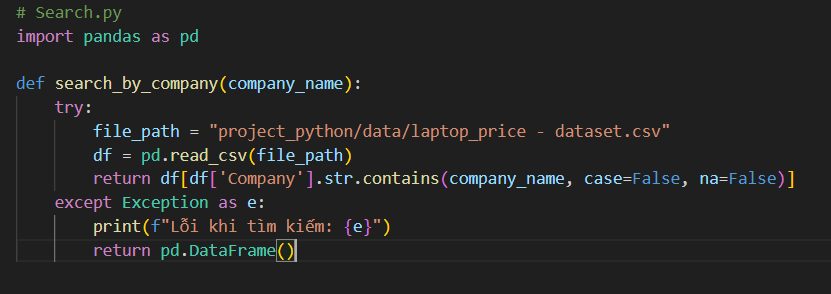
Hình 28: Code hàm sắp xếp dữ liệu

Các hàm “sort\_laptops\_by\_name”, “sort\_laptops\_by\_price”, và “sort\_laptops\_by\_weight” sử dụng phương thức “sort\_values()” của Pandas để sắp xếp dữ liệu theo từng tiêu chí.

Thứ tự sắp xếp có thể là tăng dần (`ascending=True`) hoặc giảm dần (`ascending=False`) tùy vào tham số đầu vào.

### **3.3.6. Cơ chế tìm kiếm dữ liệu(Search)**

Module Search.py được thiết kế để thực hiện các chức năng tìm kiếm thông tin trong một tập dữ liệu dạng CSV. Tập dữ liệu được sử dụng trong module là một file chứa thông tin giá laptop với tên file: laptop\_price - dataset.csv. Chức năng chính của module là hỗ trợ tìm kiếm dữ liệu dựa trên tên công ty (Company) được cung cấp từ người dùng.

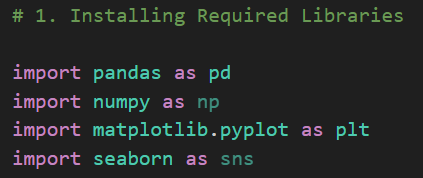
****

Hình 29: Code hàm tìm kiếm dữ liệu

**Hàm search\_by\_company** thực hiện tìm kiếm các dòng dữ liệu trong tệp CSV chứa thông tin liên quan đến một công ty cụ thể, dựa trên từ khóa người dùng nhập vào.

## **3.4. Phân tích dữ liệu (Data analysis)**

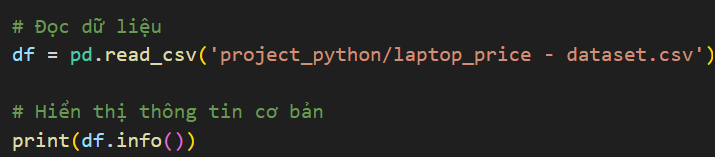
### **3.4.1. Các thư viện cần thiết**

****

Hình 30: Code các thư viện cần thiết cho việc phân tích dữ liệu

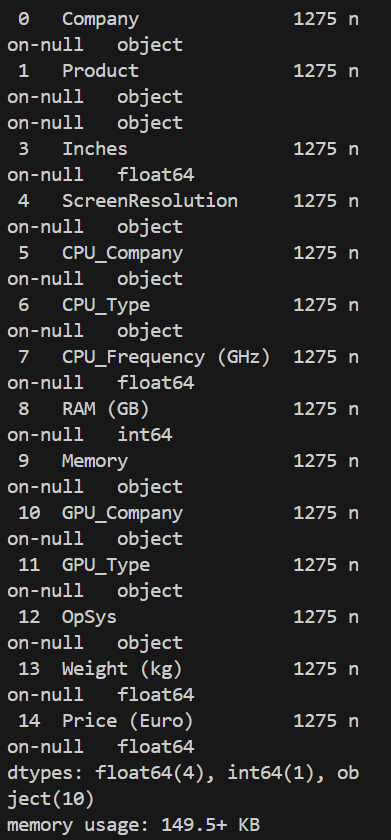
* **pandas**: Để quản lý và phân tích dữ liệu dạng bảng (DataFrame).
* **numpy**: Hỗ trợ các phép toán số học và xử lý dữ liệu dạng mảng.
* **matplotlib.pyplot**: Vẽ các biểu đồ trực quan cơ bản.
* **seaborn**: Thư viện trực quan hóa dữ liệu nâng cao, tích hợp tốt với **pandas**.

### **3.4.2. Xử lý dữ liệu thiếu và trùng lặp**

****

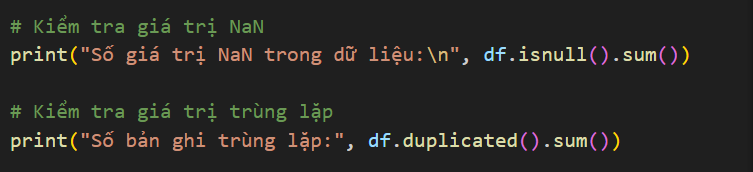
Hình 31 :Code đọc dữ liệu file và hiển thị thông tin cơ bản

Đọc tệp CSV chứa dữ liệu về laptop vào một DataFrame để xử lý. Và đồng thời hiển thị thông tin về các cột, kiểu dữ liệu, và số lượng giá trị không rỗng (non-null) bằng cách sử dụng **infor()**.



Hình 32: Hiển thị danh sách

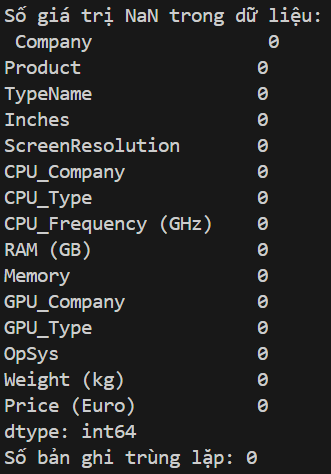
Khi làm việc với dữ liệu thực tế, xử lý giá trị thiếu (NaN) và dữ liệu trùng lặp là bước đầu tiên rất quan trọng. Việc này đảm bảo chất lượng dữ liệu và độ chính xác của phân tích dữ liệu



Hình 33: Code kiểm tra giá trị bị thiếu trong từng cột và kiểm tra các bản ghi trùng lặp

Sử dụng **isnull()** và **sum()** để kiểm tra số lượng giá trị bị thiếu trong từng cột.

Sử dụng **duplicated()** để kiểm tra các bản ghi bị trùng lặp.



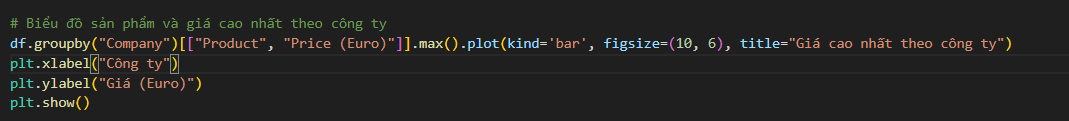
Hình 34: Xuất ra các giá trị bị thiếu trong từng cột và các bản ghi trùng lặp

Dựa vào kết quả kiểm tra dữ liệu, ta thấy rằng:

* Không có giá trị thiếu trong bất kỳ cột nào. Điều này có nghĩa rằng tập dữ liệu đã sạch về mặt thiếu giá trị và không cần thực hiện thêm các bước xử lý dữ liệu thiếu.
* Số bản ghi trùng lặp bằng không. Điều này cho thấy dữ liệu không có bất kỳ bản ghi nào bị lặp lại, do đó không cần xử lý thêm để loại bỏ bản ghi trùng lặp.

### **3.4.3 Nhóm dữ liệu theo công ty và vẽ biểu đồ**

**Biểu đồ sản phẩm và giá cao nhất theo công ty**

****

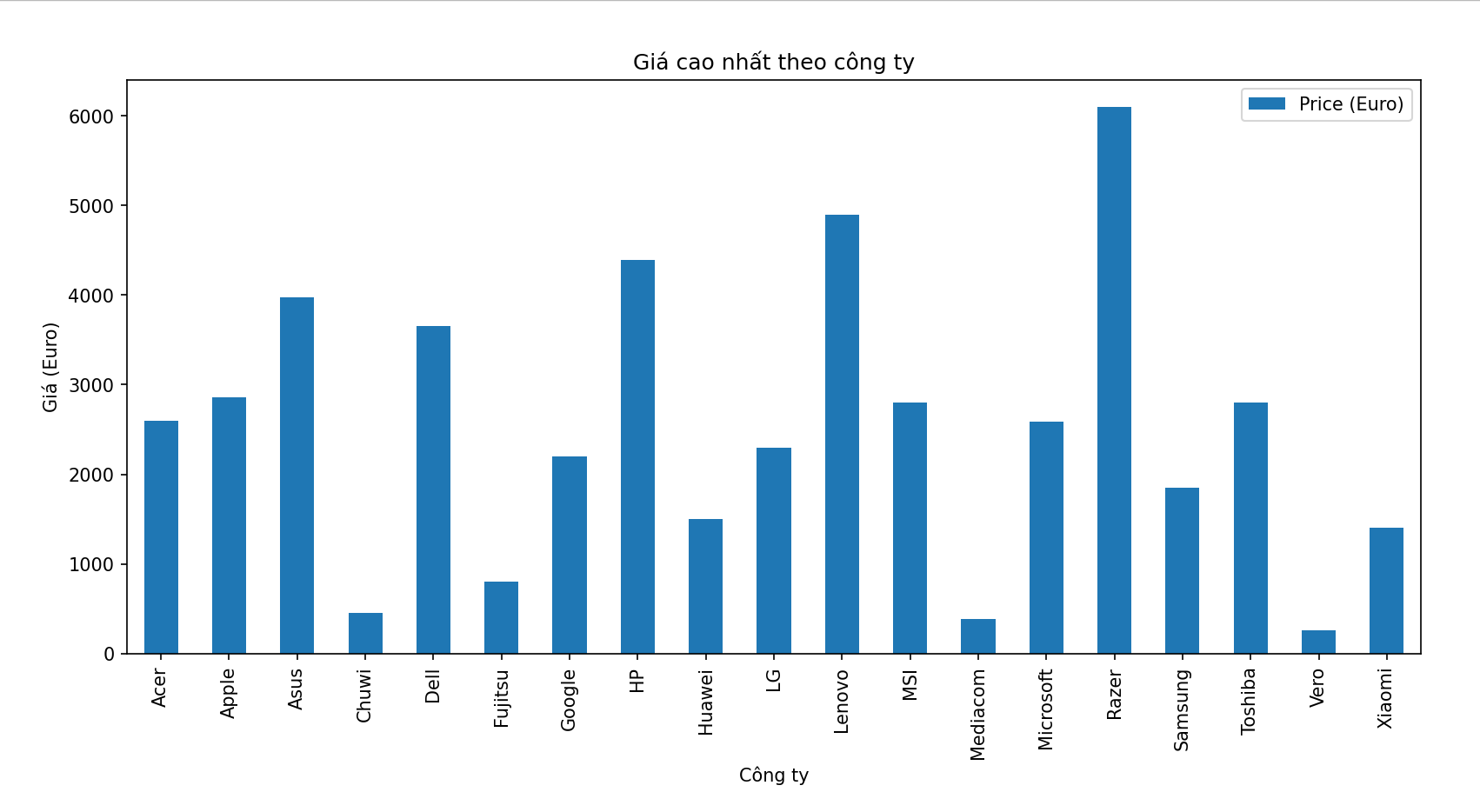
Hình 35: Code biểu đồ sản phẩm và giá cao nhất theo công ty

**Mục tiêu**:

* Tìm giá cao nhất của laptop theo từng công ty.
* Biểu diễn kết quả bằng biểu đồ cột (**bar**).

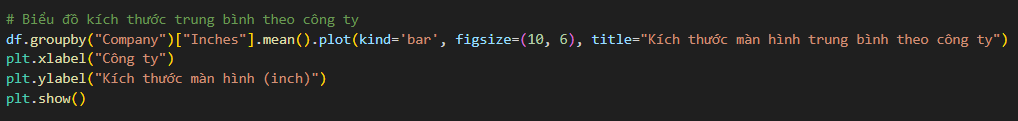
**Cách thức hoạt động**:

* **groupby("Company")**: Nhóm dữ liệu theo cột **Company**.
* **max()**: Lấy giá trị cao nhất trong từng nhóm.
* **plot(kind='bar'):** Vẽ biểu đồ dạng cột.

****

Hình 36: Biểu đồ sản phẩm và giá cao nhất theo công ty

**Biểu đồ kích thước màn hình trung bình theo công ty**

****

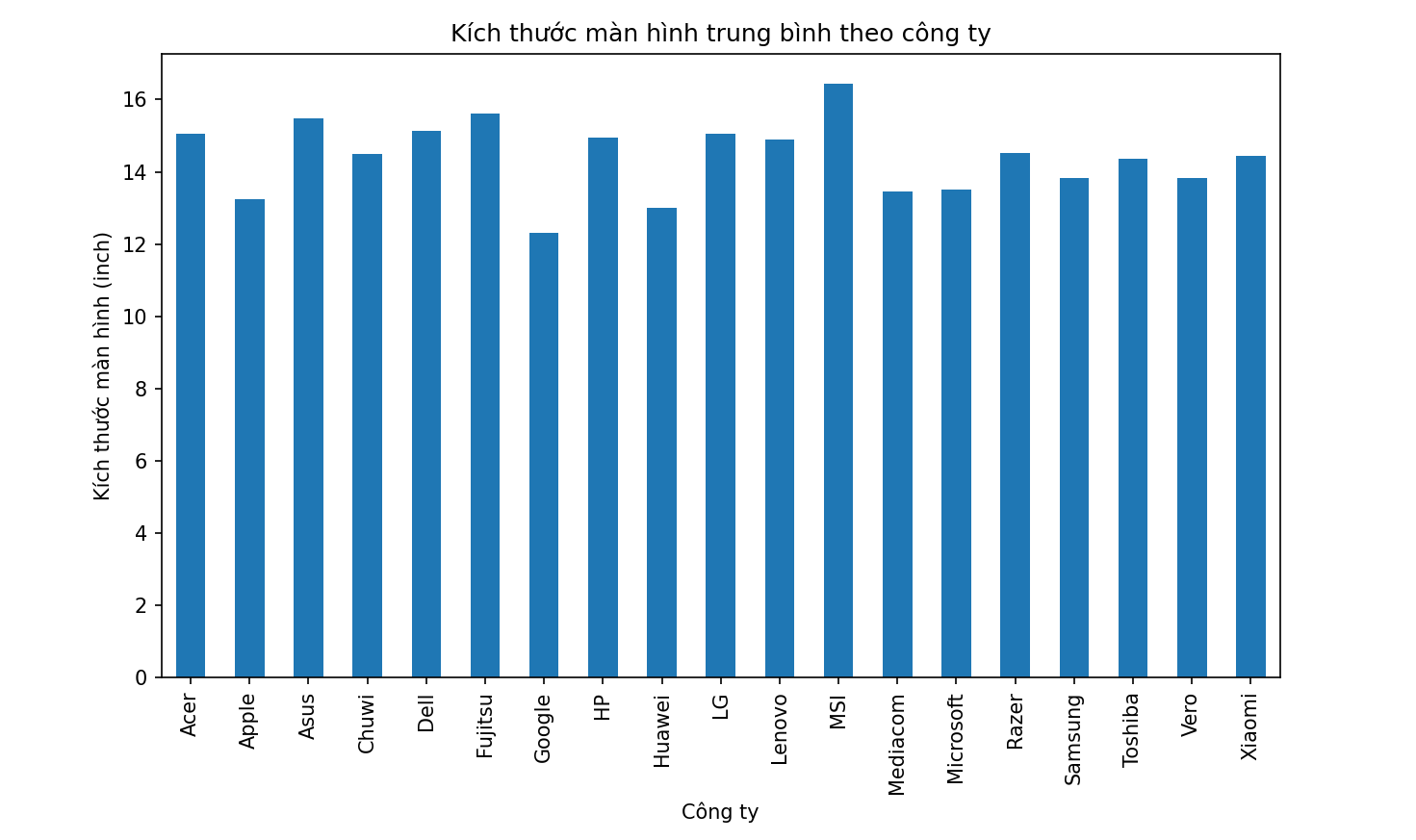
Hình 37: Code biểu đồ kích thước màn hình trung bình theo công ty

**Mục tiêu**:

* Tính kích thước màn hình trung bình (Inches) cho từng công ty.
* So sánh các công ty về kích thước màn hình trung bình.

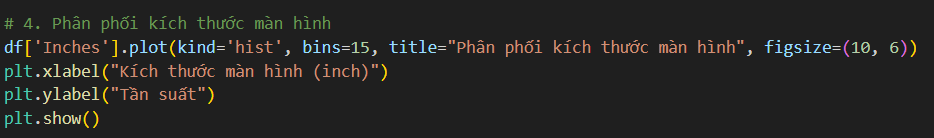
**Cách thức hoạt động**:

* groupby("Company"): Nhóm dữ liệu theo công ty.
* mean(): Tính giá trị trung bình.
* plot(kind='bar'): Hiển thị kết quả dưới dạng biểu đồ cột.

****

Hình 38: Biểu đồ kích thước màn hình trung bình theo công ty

### **3.4.4. Phân phối kích thước màn hình**

****

Hình 39: Code thể hiện biểu đồ kích thước màn hình

**Mục tiêu**:

* Hiển thị tần suất của các giá trị kích thước màn hình (Inches).
* Giúp nhận biết kích thước nào phổ biến nhất trong dữ liệu.

**Cách thức hoạt động**:

* plot(kind='hist'): Tạo biểu đồ histogram.
* bins=15: Chia kích thước màn hình thành 15 khoảng giá trị.
* title, xlabel, ylabel: Thêm tiêu đề và nhãn cho biểu đồ.

# **CHƯƠNG 4: GIAO DIỆN TƯƠNG TÁC NGƯỜI DÙNG**

## **4.1. Thư viện Tkinter:**

### **4.1.1. Định nghĩa và các thành phần của thư viện Tkinter:**

**Định nghĩa:**

Tkinter là thư viện tiêu chuẩn của Python để xây dựng giao diện đồ họa (GUI - Graphical User Interface). Nó cung cấp một tập hợp các công cụ và thành phần để tạo các cửa sổ giao diện tương tác với người dùng, giúp phát triển các ứng dụng desktop dễ dàng hơn. Tkinter được tích hợp sẵn trong các phiên bản Python và dựa trên thư viện đồ họa Tk của Tcl/Tk.

**Cách hoạt động của Tkinter:**

* Tkinter sử dụng mô hình hướng sự kiện (Event-driven programming), trong đó người dùng tương tác với giao diện (nhấp chuột, nhập liệu...), và chương trình phản hồi lại các sự kiện này.
* Giao diện được xây dựng bằng cách tạo các widget (thành phần giao diện) và bố trí chúng trong cửa sổ.

**Các thành phần chính của Tkinter**

**1.** **Tkinter Widgets (Các điều khiển)**

Các widget là các thành phần cơ bản trong giao diện đồ họa

|  |  |
| --- | --- |
| Widget | Mô tả |
| |  | | --- | | **Tk()** |  |  | | --- | |  | | |  | | --- | | Tạo cửa sổ chính cho ứng dụng. |  |  | | --- | |  | |
| |  | | --- | | **Label** |  |  | | --- | |  | | |  | | --- | | Hiển thị văn bản hoặc hình ảnh trong giao diện. |  |  | | --- | |  | |
| Button | |  | | --- | | Tạo nút bấm để thực hiện lệnh khi người dùng nhấn. |  |  | | --- | |  | |
| Entry | |  | | --- | | Tạo ô nhập liệu văn bản một dòng. |  |  | | --- | |  | |
| Text | |  | | --- | | Tạo vùng nhập liệu văn bản nhiều dòng. |  |  | | --- | |  | |
| Frame | |  | | --- | | Tạo khung chứa các widget khác để nhóm hoặc bố cục. |  |  | | --- | |  | |
| Canvas | |  | | --- | | Vẽ các hình dạng đồ họa như hình tròn, hình chữ nhật... |  |  | | --- | |  | |
| Checkbutton | |  | | --- | | Tạo ô kiểm tra (checkbox) để chọn/bỏ chọn một tùy chọn. |  |  | | --- | |  | |
| |  | | --- | | **Radiobutton** |  |  | | --- | |  | | |  | | --- | | Tạo nút radio để lựa chọn một trong nhiều tùy chọn. |  |  | | --- | |  | |
| Listbox | |  | | --- | | Hiển thị danh sách các mục và cho phép chọn một hoặc nhiều mục. |  |  | | --- | |  | |
| Menu | |  | | --- | | Tạo thanh menu hoặc menu ngữ cảnh. |  |  | | --- | |  | |
| Scrollbar | |  | | --- | | Tạo thanh cuộn cho các widget khác như Text hoặc Listbox. |  |  | | --- | |  | |
| |  | | --- | | **Scale** |  |  | | --- | |  | | |  | | --- | | Tạo thanh trượt để chọn giá trị trong một khoảng. |  |  | | --- | |  | |
| Combobox | |  | | --- | | Tạo hộp chọn kết hợp giữa Entry và Listbox. |  |  | | --- | |  | |
| Treeview | |  | | --- | | Hiển thị dữ liệu dạng bảng hoặc cây phân cấp. |  |  | | --- | |  | |
| Progressbar | |  | | --- | | Hiển thị tiến trình thực hiện của một tác vụ. |  |  | | --- | |  | |
| Toplevel | Tạo cửa sổ phụ trong giao diện. |

**2. Geometry Management (Quản lý bố cục)**

Tkinter cung cấp các phương pháp để sắp xếp widget trong cửa sổ. Có 3 trình quản lý bố cục chính:

* **pack()**: Sắp xếp widget theo hướng dọc hoặc ngang.

python

Sao chép mã

label.pack(side='top')

* **grid()**: Sắp xếp widget theo lưới (dòng và cột).

python

Sao chép mã

entry.grid(row=0, column=1)

* **place()**: Định vị widget theo tọa độ tuyệt đối (x, y).

python

Sao chép mã

button.place(x=50, y=100)

**3. Event Handling (Xử lý sự kiện)**

Tkinter lắng nghe các hành động từ người dùng như nhấn nút, nhập văn bản... thông qua các sự kiện.

* **Bind event**: Liên kết sự kiện với hàm xử lý.

python

Sao chép mã

button.bind("<Button-1>", click\_handler)

* **Các loại sự kiện phổ biến:**
  + "<Button-1>": Nhấn chuột trái.
  + "<KeyPress>": Nhấn phím trên bàn phím.
  + "<Enter>": Di chuyển chuột vào widget.
  + "<Leave>": Di chuyển chuột ra khỏi widget.

**4. Messagebox (Hộp thoại thông báo)**

Cung cấp các hộp thoại thông báo cho người dùng:

python

Sao chép mã

from tkinter import messagebox

messagebox.showinfo("Thông báo", "Dữ liệu đã được lưu!")

### **4.1.2. Ưu điểm, nhược điểm và ứng dụng của thư viện Tkinter:**

**Ưu điểm của Tkinter:**

1. Dễ học và sử dụng:  
   Tkinter cung cấp một API đơn giản và dễ hiểu, rất phù hợp cho những người mới bắt đầu lập trình GUI trong Python.
2. Tích hợp sẵn trong Python:  
   Không cần cài đặt thêm, Tkinter được tích hợp sẵn trong hầu hết các phiên bản Python, giúp giảm bớt công đoạn cài đặt.
3. Đa nền tảng:  
   Tkinter chạy trên nhiều hệ điều hành như Windows, macOS và Linux mà không cần thay đổi mã nguồn.
4. Cung cấp nhiều widget cơ bản:  
   Tkinter hỗ trợ nhiều widget (nút, nhãn, hộp văn bản, khung hình...) giúp xây dựng giao diện phong phú.
5. Khả năng mở rộng:  
   Có thể kết hợp Tkinter với các thư viện khác như Pillow (xử lý hình ảnh) hoặc ttk (giao diện đẹp hơn) để mở rộng chức năng.
6. Tài liệu phong phú:  
   Tkinter có tài liệu tham khảo chi tiết và nhiều cộng đồng hỗ trợ trực tuyến.

**Nhược điểm của Tkinter**

1. Giao diện đơn giản, thiếu thẩm mỹ:  
   Giao diện mặc định của Tkinter khá cơ bản và không bắt mắt, kém hơn so với các thư viện GUI hiện đại khác như PyQt hoặc Kivy.
2. Hạn chế trong thiết kế giao diện phức tạp:  
   Khó khăn khi xây dựng giao diện người dùng có bố cục phức tạp hoặc yêu cầu hiệu ứng động.
3. Hiệu năng thấp:  
   Không phù hợp cho các ứng dụng cần đồ họa phức tạp hoặc xử lý nhiều dữ liệu, đặc biệt là khi so sánh với các framework GUI khác.
4. Cấu trúc sự kiện hạn chế:  
   Mô hình xử lý sự kiện của Tkinter đơn giản và đôi khi không linh hoạt, không phù hợp cho các ứng dụng lớn, phức tạp.
5. Thiếu hỗ trợ cho giao diện hiện đại:  
   Không hỗ trợ tốt các giao diện người dùng hiện đại, thích ứng trên các thiết bị màn hình cảm ứng.

**Ứng dụng của Tkinter**

1. Xây dựng các công cụ nhỏ (Utility tools):
   * Các công cụ đơn giản như máy tính, trình chỉnh sửa văn bản nhỏ, hoặc trình chuyển đổi định dạng dữ liệu.
2. Ứng dụng giáo dục:
   * Được sử dụng để giảng dạy lập trình GUI cho người mới bắt đầu vì dễ học và tích hợp sẵn.
3. Ứng dụng nội bộ doanh nghiệp:
   * Phát triển các phần mềm nhỏ cho nhu cầu nội bộ trong công ty, ví dụ như phần mềm quản lý thông tin cơ bản.
4. Ứng dụng quản lý cơ bản:
   * Ứng dụng CRUD (Create, Read, Update, Delete) để quản lý dữ liệu đơn giản trong các hệ thống nhỏ.
5. Prototype và thử nghiệm:
   * Tạo nhanh các bản mẫu (prototype) giao diện người dùng trước khi phát triển ứng dụng hoàn chỉnh với các công cụ mạnh hơn.
6. Công cụ hỗ trợ lập trình viên:
   * Tạo các script với giao diện đồ họa để tự động hóa một số công việc hoặc quản lý dự án.

## **4.2. Giao diện tương tác người dung:**

### **4.2.1. Mục đích của giao diện:**

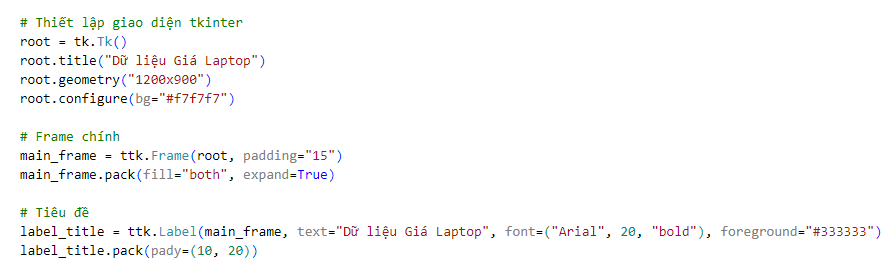
Giao diện được thiết kế nhằm mục đích:

* Hiển thị thông tin về các dòng laptop với các thuộc tính như Company, Product, TypeName, Inches, Screen Resolution, Price, v.v.
* Cung cấp các chức năng xử lý dữ liệu như tìm kiếm, phân trang, thêm mới, sửa, xóa, sắp xếp, làm sạch, chuẩn hóa và vẽ biểu đồ.
* Đơn giản hóa thao tác của người dùng đối với dataset lớn thông qua phân trang và biểu diễn trực quan dữ liệu.

### **4.2.2. Thiết kế giao diện chính:**

**1. Cửa sổ chính, frame chính và tiêu đề :**

Đoạn mã này tạo một giao diện Tkinter cơ bản với cửa sổ chính, một frame chứa các thành phần và một tiêu đề lớn hiển thị thông tin.



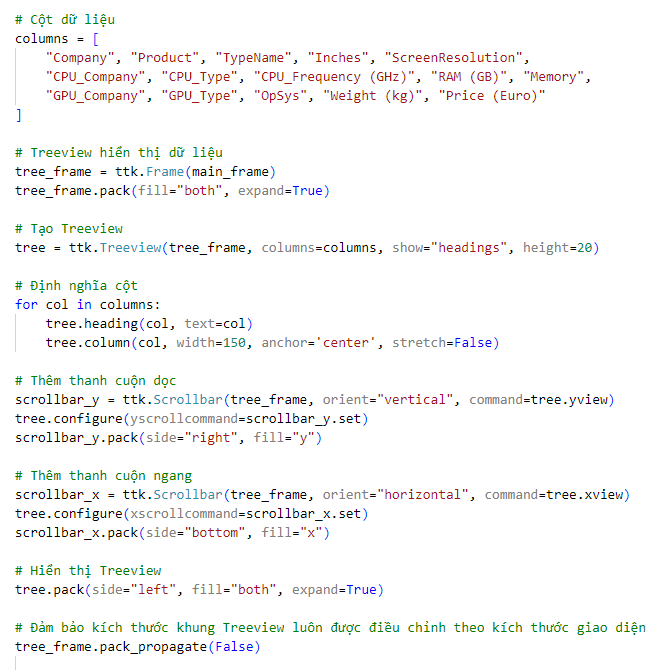
Hình 40: code tạo cửa sổ chính, frame chính và tiêu đề

**2. TreeView hiển thị dữ liệu:**

* TreeView được sử dụng để hiển thị dữ liệu laptop. Nó gồm các cột sau:

+ Company, Product, TypeName, Inches, Screen Resolution, CPU\_Company, CPU\_Type, CPU\_Frequency (GHz), RAM (GB), Memory, GPU\_Company, GPU\_Type, OpSys, Weight (kg), Price (Euro)

+ Thanh cuộn ngang và dọc được thêm để hỗ trợ khi số lượng cột và dòng lớn.



Hình 41: code treeview hiển thị dữ liệu

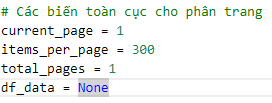
**3. Phân trang:**

* Tính năng:

+ Hiển thị dữ liệu theo từng trang để tránh quá tải khi số lượng bản ghi lớn.

+ Chuyển trang qua các nút "Trang trước" và "Trang sau".

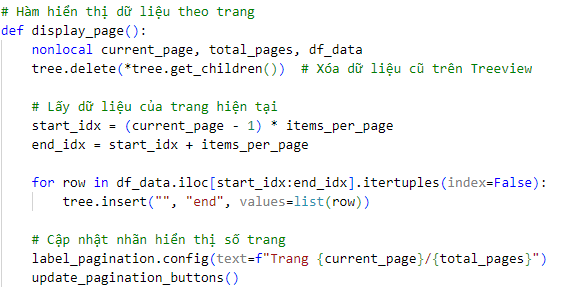
a. Biến toàn cục cho phân trang



Hình 42: biến toàn cục cho phân trang

* current\_page: Biến lưu trang hiện tại, mặc định là trang 1.
* items\_per\_page: Số lượng mục (dòng dữ liệu) hiển thị trên mỗi trang. Ở đây là 300 dòng/trang.
* total\_pages: Tổng số trang, được tính dựa trên số lượng dữ liệu.
* df\_data: Biến lưu trữ dữ liệu được đọc từ file CSV.

b. Hàm hiển thị dữ liệu theo trang



Hình 43: hàm hiển thị dữ liệu theo trang

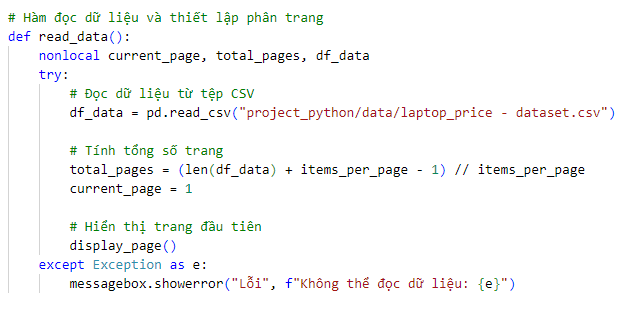
* Xóa dữ liệu cũ: Trước khi hiển thị trang mới, tree.delete(\*tree.get\_children()) xóa tất cả các dòng dữ liệu hiện có trên Treeview.
* Tính phạm vi dữ liệu:

+ start\_idx: Vị trí bắt đầu của dòng dữ liệu trên trang hiện tại.

+ end\_idx: Vị trí kết thúc của dòng dữ liệu trên trang hiện tại.

* Thêm dữ liệu vào Treeview: Lặp qua các dòng dữ liệu trong phạm vi xác định và thêm chúng vào Treeview.
* Cập nhật giao diện: Nhãn hiển thị số trang được cập nhật để phản ánh trang hiện tại.

c. Hàm đọc dữ liệu và thiết lập phân trang

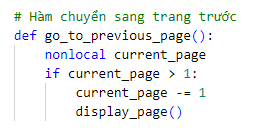


Hình 44: hàm đọc dữ liệu và thiết lập phân trang

* Đọc dữ liệu từ file CSV: Sử dụng Pandas để đọc dữ liệu.
* Hiển thị trang đầu tiên: Gọi hàm display\_page() để hiển thị dữ liệu của trang 1.

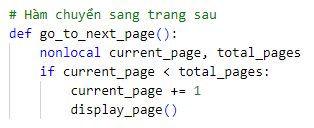
d. Hàm chuyển trang:

* Chuyển sang trang trước:



Hình 45: hàm chuyển trang trước

* Chuyển sang trang sau:



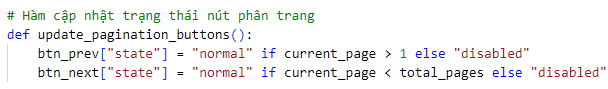
Hình 46: hàm chuyển trang sau

* Kiểm tra điều kiện:

+ Chỉ cho phép chuyển trang nếu không phải trang đầu tiên hoặc trang cuối cùng.

+ Cập nhật current\_page và gọi lại display\_page().

e. Cập nhật trạng thái nút phân trang:



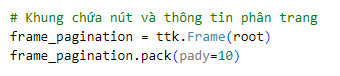
Hình 47: hàm cập nhật trạng thái phân trang

* Vô hiệu hóa hoặc kích hoạt nút:

+ Nút "Trang trước" bị vô hiệu hóa nếu đang ở trang đầu tiên.

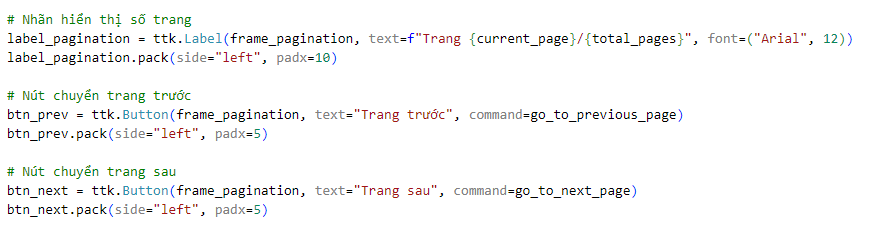
+ Nút "Trang sau" bị vô hiệu hóa nếu đang ở trang cuối cùng.

f. Tạo giao diện phân trang:



Hình 48: khung chứa nút phân trang

* Khung chứa nút phân trang: Được đặt dưới Treeview, với khoảng cách dọc 10 pixel.



Hình 49: nhãn hiển thị số trang và nút điều khiển

* Nhãn hiển thị số trang

+ ttk.Label(): Tạo một nhãn hiển thị số trang hiện tại và tổng số trang.

+ text=f"Trang {current\_page}/{total\_pages}": Hiển thị thông tin về trang hiện tại (ví dụ: Trang 1/10).

+ Font chữ: Sử dụng font Arial với kích thước 12.

+ padx=10: Tạo khoảng cách ngang 10 pixel xung quanh nhãn.

* Nút chuyển trang trước

+ ttk.Button(): Tạo một nút điều hướng để chuyển về trang trước.

+ text="Trang trước": Nhãn trên nút.

+ command=go\_to\_previous\_page: Gọi hàm go\_to\_previous\_page() khi nút được nhấn.

+ padx=5: Tạo khoảng cách ngang 5 pixel xung quanh nút.

* Nút chuyển trang sau

+ text="Trang sau": Nhãn trên nút.

+ command=go\_to\_next\_page: Gọi hàm go\_to\_next\_page() khi nút được nhấn.

+ Vị trí và khoảng cách: Tương tự như nút Trang trước.

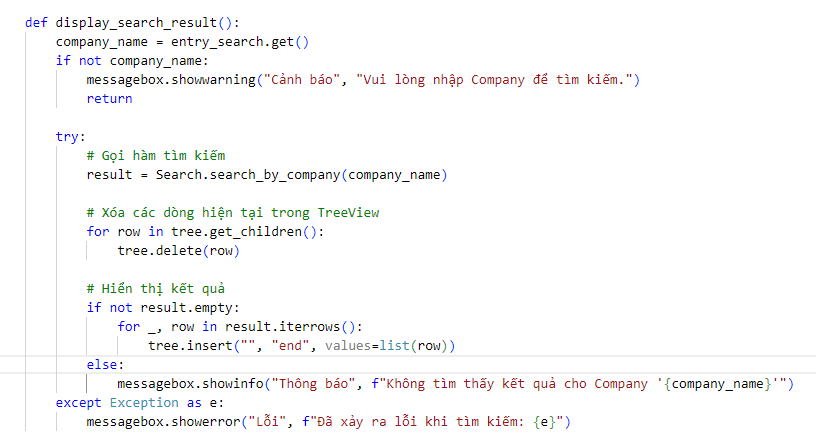
### **4.2.3. Thiết kế giao diện phần Feature**

**1. Tìm kiếm:**

* Tính năng:

+ Người dùng có thể tìm kiếm thông tin laptop theo tên công ty.

+ Kết quả được hiển thị trong TreeView.



Hình 50: hàm hiển thị kết quả tìm kiếm

* Nhập và kiểm tra dữ liệu đầu vào

+ entry\_search.get(): Lấy giá trị từ ô nhập liệu (entry\_search).

+ Kiểm tra rỗng:

Nếu người dùng không nhập gì vào ô tìm kiếm, hiển thị thông báo cảnh báo và thoát khỏi hàm.

* Gọi hàm tìm kiếm

+ Search.search\_by\_company(): Hàm này giả định là nằm trong một lớp hoặc module Search, thực hiện tìm kiếm trong dữ liệu dựa trên tên công ty được cung cấp.

+ Kết quả trả về: Được lưu trong result, thường là một DataFrame của Pandas.

* Xóa dữ liệu hiện tại trong Treeview

+ Mục đích: Xóa tất cả các dòng hiện có trong Treeview trước khi hiển thị kết quả mới.

+ tree.get\_children(): Trả về danh sách tất cả các mục trong Treeview.

+ tree.delete(row): Xóa từng dòng một.

* Hiển thị kết quả tìm kiếm

+ Kiểm tra kết quả:

if not result.empty:: Kiểm tra xem DataFrame có rỗng không.

Nếu có kết quả:

result.iterrows(): Lặp qua từng dòng của DataFrame.

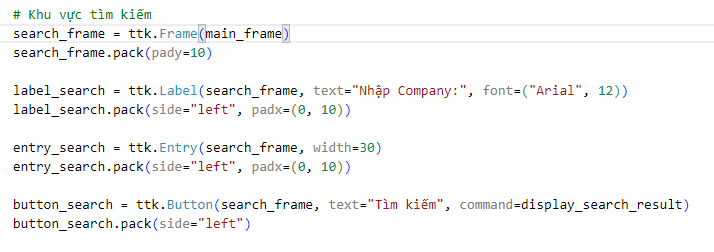
tree.insert(): Thêm mỗi dòng dữ liệu vào Treeview.

+ Nếu không có kết quả:

Hiển thị thông báo cho người dùng rằng không tìm thấy dữ liệu phù hợp.

* Xử lý ngoại lệ

+ Bắt lỗi: Nếu có bất kỳ lỗi nào xảy ra trong quá trình tìm kiếm hoặc xử lý dữ liệu, một hộp thoại lỗi sẽ hiển thị thông tin chi tiết về lỗi đó.



Hình 51: khu vực tìm kiếm

* Tạo Frame chứa các thành phần tìm kiếm

+ search\_frame: Là một Frame con trong main\_frame, dùng để chứa các thành phần liên quan đến tìm kiếm.

+ pady=10: Tạo khoảng cách dọc 10 pixel giữa Frame tìm kiếm và các thành phần xung quanh.

* Nhãn hướng dẫn nhập liệu

+ Nhãn hướng dẫn: Hiển thị dòng chữ "Nhập Company:" để hướng dẫn người dùng biết cần nhập tên hãng.

+ Font chữ: Sử dụng font Arial kích thước 12.

+ padx=(0, 10): Tạo khoảng cách ngang 10 pixel bên phải nhãn.

* Ô nhập liệu (Entry)

+ ttk.Entry(): Tạo một ô nhập liệu, cho phép người dùng nhập tên công ty cần tìm kiếm.

+ width=30: Độ rộng của ô nhập liệu (số lượng ký tự hiển thị tối đa).

+ padx=(0, 10): Tạo khoảng cách ngang 10 pixel bên phải ô nhập liệu.

* Nút tìm kiếm

+ Nút tìm kiếm: Hiển thị nút có nhãn "Tìm kiếm".

+ command=display\_search\_result: Khi nhấn nút, hàm display\_search\_result() sẽ được gọi để thực hiện tìm kiếm.\

**2. Thêm bản ghi:**

* Tính năng:

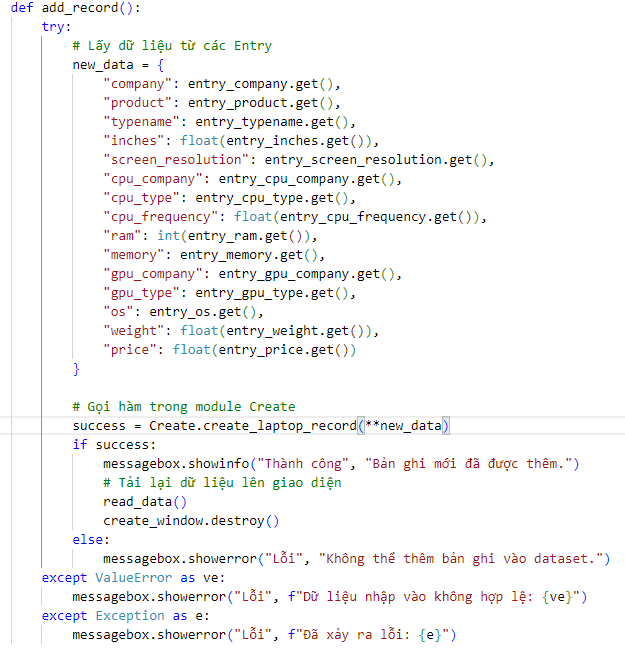
+ Cung cấp giao diện nhập liệu để thêm bản ghi mới.

+ Sử dụng module Create để lưu dữ liệu mới vào file CSV.



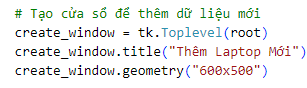
Hình 52: hàm mở cửa sổ tạo bản ghi mới

* Mở một cửa sổ phụ (Toplevel) để nhập thông tin.



Hình 53: hàm xử lý việc them bản ghi

* Lấy dữ liệu từ Entry: Lấy thông tin từ các ô nhập liệu và chuyển đổi kiểu dữ liệu phù hợp (ví dụ: float cho giá, int cho RAM).
* Xử lý ngoại lệ: Kiểm tra lỗi dữ liệu (ví dụ: nhập chữ vào trường số) và thông báo lỗi tương ứng.
* Gọi hàm thêm dữ liệu: Giả định rằng hàm Create.create\_laptop\_record xử lý việc thêm dữ liệu vào dataset hoặc file CSV.



Hình 54: tạo cửa sổ nhập liệu

* Toplevel: Tạo cửa sổ con độc lập nhưng liên kết với cửa sổ chính (root).
* Kích thước: Cửa sổ được thiết lập kích thước 600x500 pixel.



Hình 55: tạo các trường nhập liệu

* Nhãn và Entry tự động tạo: Sử dụng vòng lặp để tạo Label và Entry cho mỗi trường thông tin.
* row=i // 4 và column=(i % 4) \* 2: Sắp xếp các trường theo hàng ngang, mỗi hàng 4 cột.
* entries: Lưu các Entry widget vào một từ điển (dictionary) để dễ truy cập.



Hình 56: nút thêm bản ghi

* command=add\_record: Gọi hàm add\_record() khi nhấn nút.
* Vị trí nút: Đặt nút ở dưới cùng của cửa sổ.



Hình 57: nút tạo Laptop mới

* ttk.Button(): Tạo một nút sử dụng thư viện ttk (themed Tkinter widgets) để có giao diện hiện đại.

+ text="Tạo Laptop Mới": Hiển thị trên nút.

+ command=open\_create\_window: Gọi hàm open\_create\_window() khi nút được nhấn.

+ width=25: Đặt chiều rộng của nút (25 ký tự).

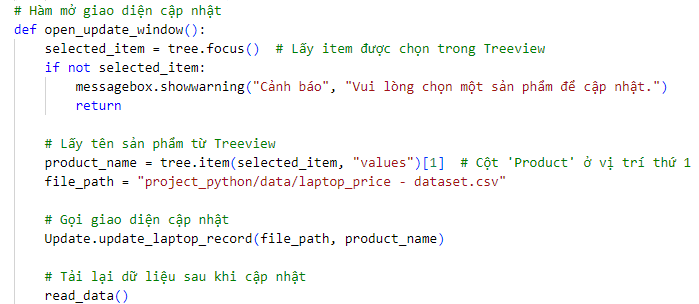
* pack(pady=10): Đặt khoảng cách dọc (padding vertical) là 10 pixel.

**3. Cập nhật bản ghi:**

-Tính năng:

+Người dùng chọn bản ghi từ TreeView để chỉnh sửa thông tin.

+Sử dụng module Update để áp dụng thay đổi.



Hình 58: hàm mở giao diện cập nhật

-Chức năng chính: Mở giao diện cập nhật thông tin cho sản phẩm được chọn trong Treeview.

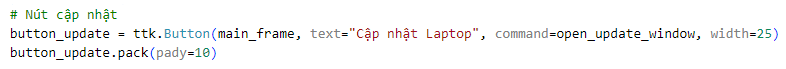
+selected\_item = tree.focus(): Lấy item đang được chọn trong Treeview.

+Kiểm tra chọn item: Nếu không có sản phẩm nào được chọn, hiển thị cảnh báo yêu cầu chọn sản phẩm.

+Lấy tên sản phẩm: Sử dụng tree.item(selected\_item, "values") để truy xuất thông tin của sản phẩm được chọn. Tên sản phẩm nằm ở cột "Product", có chỉ số là 1.

+Gọi hàm cập nhật: Sau khi xác định được sản phẩm, gọi hàm Update.update\_laptop\_record để mở giao diện cập nhật thông tin sản phẩm.

+Tải lại dữ liệu: Sau khi cập nhật, gọi read\_data() để làm mới danh sách sản phẩm trong Treeview.



Hình 59: tạo nút cập nhật

-ttk.Button(): Tạo nút sử dụng ttk (thư viện giao diện người dùng hiện đại của Tkinter).

+text="Cập nhật Laptop": Nội dung hiển thị trên nút.

+command=open\_update\_window: Khi nút được nhấn, hàm open\_update\_window sẽ được gọi, mở giao diện để cập nhật thông tin laptop.

+width=25: Đặt chiều rộng của nút là 25 ký tự.

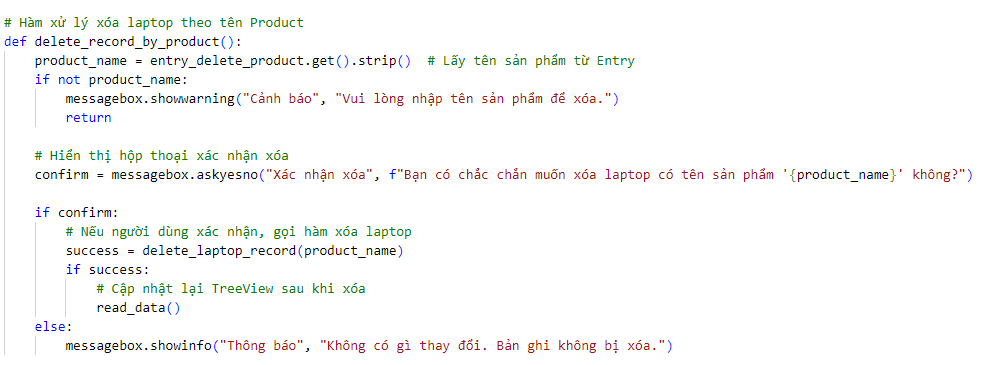
-pack(pady=10): Đặt nút vào giao diện với khoảng cách dọc (vertical padding) là 10 pixel giữa các thành phần khác trong khung

**4. Xóa bản ghi:**

-Tính năng:

+Người dùng nhập tên sản phẩm cần xóa. Dữ liệu sẽ bị loại bỏ khỏi dataset.

+Sử dụng module Delete.

****

Hình 60: hàm xử lí xóa laptop theo tên product

-Lấy thông tin từ Entry:

+entry\_delete\_product.get().strip(): Lấy tên sản phẩm cần xóa từ ô nhập liệu, loại bỏ khoảng trắng đầu/cuối.

-Kiểm tra dữ liệu đầu vào:

+Hiển thị cảnh báo nếu người dùng chưa nhập tên sản phẩm.

-Hộp thoại xác nhận:

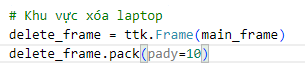
+messagebox.askyesno(): Hộp thoại xác nhận với tùy chọn Yes hoặc No. Nếu người dùng chọn Yes, tiến hành xóa dữ liệu.

-Gọi hàm xóa:

+delete\_laptop\_record(product\_name): Hàm này thực hiện thao tác xóa bản ghi (giả định đã được định nghĩa trong mã chính).

-Cập nhật lại dữ liệu:

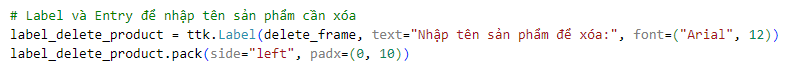
+Sau khi xóa thành công, gọi read\_data() để tải lại dữ liệu và cập nhật giao diện Treeview.



Hình 61: khung chứa các thành phần liên quan đến xóa

-ttk.Frame(main\_frame): Tạo một frame con trong main\_frame để chứa các thành phần xóa.

-pady=10: Khoảng cách dọc giữa khung xóa và các thành phần khác.



Hình 62: label hường dẫn người dung

-Hiển thị thông tin rõ ràng: Người dùng được hướng dẫn nhập tên sản phẩm cần xóa.

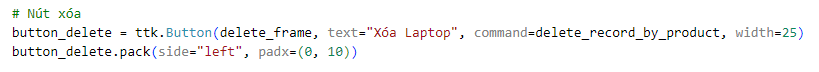
-Kiểu chữ Arial, kích thước 12: Đảm bảo dễ đọc và đồng nhất với giao diện.



Hình 63: entry để nhập tên sản phẩm

-Kích thước hợp lý: width=30 đảm bảo đủ chỗ cho người dùng nhập tên sản phẩm.

-Nhận dữ liệu từ người dùng: Dữ liệu được dùng làm đầu vào cho hàm delete\_record\_by\_product().



Hình 64: nút xóa

-Nút kích hoạt hàm xóa: Khi người dùng nhấn nút, hàm delete\_record\_by\_product() được gọi để xử lý.

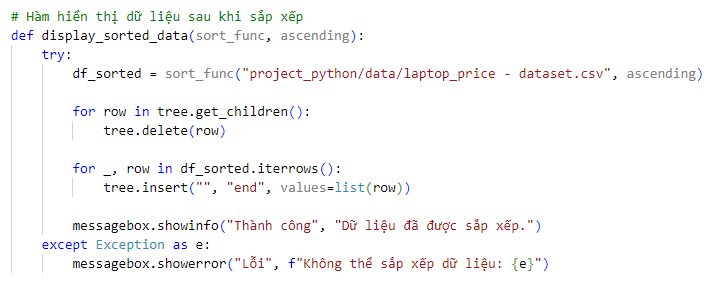
-Rộng 25 ký tự: Đảm bảo nút có kích thước dễ nhấn và đồng nhất với giao diện.

**5. Sắp xếp:**

* Tính năng:

+Cho phép sắp xếp dữ liệu theo tên công ty, giá, hoặc trọng lượng.

+Sử dụng module Sort.



Hình 65: hàm hiển thị dữ liệu sau khi sắp xếp

* sort\_func: Hàm sắp xếp dữ liệu.
* ascending: Boolean xác định thứ tự sắp xếp (True: tăng dần, False: giảm dần).
* Gọi hàm sắp xếp:

+ sort\_func thực hiện việc đọc và sắp xếp dữ liệu từ tệp CSV.

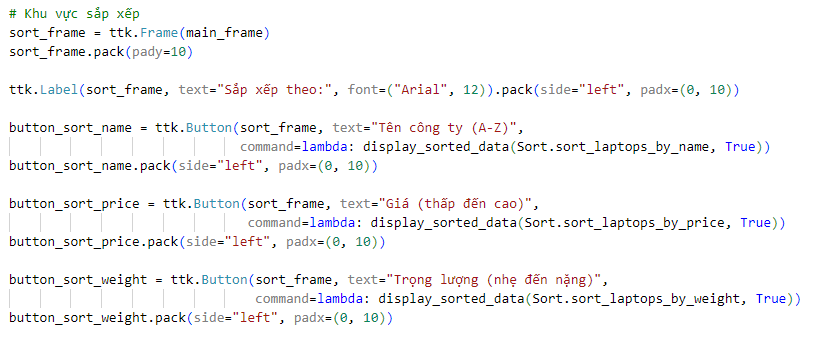
+ Tệp dữ liệu laptop\_price - dataset.csv được chỉ định cố định trong đường dẫn.

* tree.delete(): Xóa tất cả các dòng hiện có để chuẩn bị hiển thị dữ liệu đã sắp xếp mới.
* Duyệt qua từng dòng dữ liệu:

+df\_sorted.iterrows() duyệt qua từng dòng trong DataFrame đã sắp xếp.

+tree.insert() chèn các giá trị mới vào TreeView.

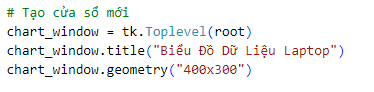
* Thông báo thành công: Xác nhận rằng dữ liệu đã được sắp xếp và hiển thị thành công.
* Xử lý lỗi: Nếu xảy ra lỗi trong quá trình sắp xếp hoặc hiển thị, người dùng sẽ nhận được thông báo chi tiết.



Hình 66: khu vực sắp xếp

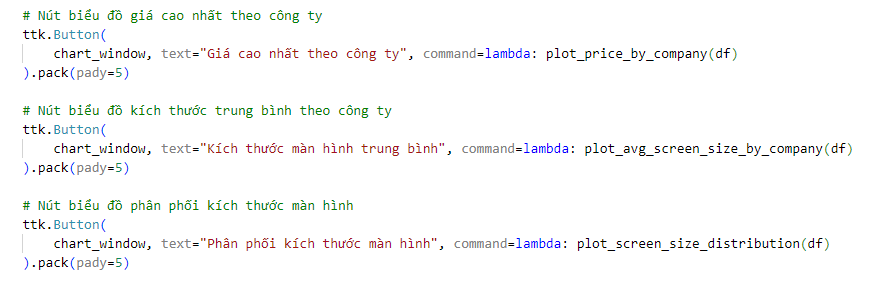
* Tạo khung chứa các nút điều khiển sắp xếp, giúp bố cục giao diện gọn gang.
* Hiển thị tiêu đề: Giúp người dùng hiểu rõ chức năng của các nút trong khung này.
* Sắp xếp dữ liệu theo tên công ty theo thứ tự bảng chữ cái (A-Z).
* Sắp xếp dữ liệu theo giá từ thấp đến cao.
* Sắp xếp dữ liệu theo trọng lượng từ nhẹ đến nặng.

**6. Vẽ biểu đồ:**

****

Hình 67: cửa sổ biểu đồ

* Chức năng: Mở một cửa sổ con (Toplevel) để chứa các tùy chọn vẽ biểu đồ.



Hình 68: các nút chọn biểu đồ

* Chức năng: Gọi các hàm vẽ biểu đồ cụ thể dựa trên dữ liệu từ dataset.

### **4.2.4. Thiết kế giao diện phần** **Data\_cleaning\_normalization:**



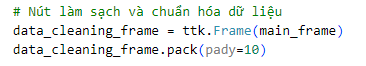
Hình 69: hàm làm sạch

* Chức năng chính: Làm sạch dữ liệu trong dataset.



Hình 70: hàm chuẩn hóa

* Chức năng chính: Chuẩn hóa dữ liệu trong dataset

****

Hình 71: khung chứa nút xử lý dữ liệu

* Mục đích: Tạo khung gọn gàng chứa các nút làm sạch và chuẩn hóa dữ liệu, giúp tổ chức giao diện dễ theo dõi.



Hình 72: nút làm sạch dữ liệu

* Chức năng:

+ Gọi hàm clean để thực hiện làm sạch dữ liệu.

+ Loại bỏ các giá trị trống hoặc không hợp lệ trong dataset.



Hình 73: nút chuẩn hóa dữ liệu

* Chức năng:

+ Gọi hàm normal để chuẩn hóa dữ liệu, đảm bảo các giá trị trong dataset được đưa về cùng một chuẩn

# **PHẦN KẾT LUẬN**

**Kết quả đạt được:**

* **Giao diện hoàn chỉnh:** Đã xây dựng một ứng dụng giao diện đồ họa (GUI) sử dụng thư viện Tkinter, cung cấp các chức năng chính như hiển thị dữ liệu, tìm kiếm, thêm, sửa, xóa, sắp xếp và vẽ biểu đồ.
* **Chức năng phong phú:**
  + Hiển thị thông tin chi tiết về laptop.
  + Phân trang để quản lý dataset lớn.
  + Tìm kiếm và lọc dữ liệu theo nhiều tiêu chí.
  + Chức năng CRUD (Create, Read, Update, Delete) hoạt động ổn định.
  + Tính năng làm sạch và chuẩn hóa dữ liệu.
  + Hỗ trợ trực quan hóa dữ liệu thông qua biểu đồ.

**Ưu điểm:**

* **Đơn giản và dễ sử dụng:** Giao diện trực quan với các nút và khung chức năng rõ ràng.
* **Phân trang hiệu quả:** Giúp người dùng quản lý dataset lớn mà không bị quá tải thông tin.
* **Tìm kiếm và CRUD tiện lợi:** Tăng khả năng kiểm soát và quản lý dữ liệu.
* **Mở rộng dễ dàng:** Có thể nâng cấp thêm tính năng hoặc tích hợp với các thư viện khác.

**Nhược điểm:**

* **Giao diện cơ bản:** Tkinter không hỗ trợ các giao diện hiện đại như PyQt hay Kivy.
* **Hiệu suất thấp:** Xử lý dữ liệu lớn có thể gặp khó khăn do giới hạn của Tkinter.
* **Hạn chế về tính năng nâng cao:** Chưa hỗ trợ tốt các hiệu ứng đồ họa phức tạp hoặc các tính năng giao diện nâng cao.
* **Phụ thuộc vào dữ liệu nhập liệu:** Chức năng xử lý dữ liệu cần đảm bảo dữ liệu đầu vào phải đúng chuẩn.

**Hướng phát triển:**

1. **Cải thiện giao diện:**
   * Sử dụng thư viện ttk để tăng tính thẩm mỹ.
   * Xem xét chuyển sang các framework hiện đại hơn như PyQt hoặc Kivy để tạo giao diện đẹp hơn.
2. **Nâng cao hiệu năng:**
   * Tối ưu hóa xử lý dữ liệu lớn.
   * Tích hợp cơ sở dữ liệu thay vì đọc/ghi trực tiếp từ CSV.
3. **Mở rộng chức năng:**
   * Thêm các tính năng nâng cao như xuất báo cáo, phân tích dữ liệu sâu hơn.
   * Cải thiện chức năng biểu đồ với nhiều loại biểu đồ và tùy chỉnh hơn.
4. **Tăng cường trải nghiệm người dùng:**
   * Tích hợp các công cụ hỗ trợ đa ngôn ngữ.
   * Cải thiện khả năng phản hồi và thông báo lỗi chi tiết hơn.

**PHẦN TÀI LIỆU THAM KHẢO**

[[1]. Kaggle - Laptop Price-dataset.](https://www.kaggle.com/datasets/ironwolf437/laptop-price-dataset?resource=download&fbclid=IwY2xjawGyx8JleHRuA2FlbQIxMAABHS0fkRAVEV-qde7H7_jwQrZkEHDmT9wk358gUy3wfRi4vin8ADyQgRLcXg_aem_GILbG2INtT0Er6-Jk9fTFQ)

[2]. [Hướng dẫn xây dựng CRUD](https://viblo.asia/p/xay-dung-ung-dung-web-crud-voi-python-va-flask-phan-mot-naQZRyydKvx?fbclid=IwY2xjawGyyMxleHRuA2FlbQIxMAABHSQCb_2gnfd_MYiyO-hpkjhWiRraqKaLSQ8cnN_i5mOQEKDJMKpUNXc_QQ_aem_yZ3dKWHvUqRkt5wHfcEgow)

[3]. [Vẽ đồ thị với matplotlib](https://topdev.vn/blog/ve-do-thi-trong-python-voi-thu-vien-matplotlib/?fbclid=IwY2xjawGyyWBleHRuA2FlbQIxMAABHbKzF9Sp9QeqhQBRsOUon2_pAvGMLzoABAMEBicQITCou2QThKZ2Xd9J-Q_aem_wrVkaVlY8v-LYX-B1uxpZw)