**UBND THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO**

**MÔN:NHẬP MÔN MÁY HỌC**

**Medicine Classification**

**Giảng viên hướng dẫn: Thầy Nguyễn Thanh Phước**

|  |  |
| --- | --- |
| **Sinh viên** | **MSSV** |
| Trần Thanh Phương | 3122410333 |
| Hồ Văn Quyến | 3122410352 |
| Nguyễn Trường Sinh | 31224103 |

**Thành phố Hồ Chí Minh- Tháng 11/2024**

**Lời Mở Đầu**

Trong những năm gần đây, ngành y tế đã chứng kiến sự phát triển vượt bậc nhờ vào các tiến bộ trong công nghệ và khoa học dữ liệu. Sự kết hợp giữa công nghệ thông tin và y học không chỉ giúp cải thiện hiệu quả chẩn đoán mà còn mang lại các giải pháp mới trong việc quản lý và phân loại các loại thuốc. Một trong những thách thức lớn là tìm hiểu và phân loại chính xác tác dụng phụ của các loại thuốc dựa trên thành phần và tác dụng của chúng. Việc nhận diện và phân loại tác dụng phụ có thể giúp các chuyên gia y tế đưa ra những lời khuyên phù hợp và tăng cường an toàn cho người sử dụng thuốc.

Trong báo cáo này, chúng tôi sẽ ứng dụng các thuật toán máy học phi mạng nơ-ron để xây dựng một mô hình phân loại, nhằm dự đoán tác dụng phụ của thuốc dựa trên các đặc điểm về thành phần và công dụng. Thông qua đó, nghiên cứu này hy vọng đóng góp một phần nhỏ vào việc nâng cao hiệu quả của các dịch vụ y tế và cải thiện an toàn trong sử dụng thuốc cho cộng đồng.

# **I. ĐỊNH NGHĨA BÀI TOÁN**

* 1. **Vấn đề cần giải quyết là gì ?**

Thuốc là một phần không thể thiếu trong chăm sóc sức khỏe và có mặt rộng rãi trên toàn thế giới . Chúng thường được sử dụng để điều trị nhiều loại bệnh và cải thiện sức khỏe của bệnh nhân. Nhiều loại thuốc có giá trị điều trị cao, cung cấp dưỡng chất thiết yếu và nhắm vào các vấn đề sức khỏe cụ thể. Tuy nhiên , cũng có rất nhiều loại thuốc có thể gây ra tác dụng phụ, từ các triệu chứng nhẹ như buồn nôn, chóng mặt đến các phản ứng nghiêm trọng, bao gồm dị ứng, tổn thương cơ quan, thậm chí là các biến chứng đe dọa tính mạng . Vì vậy , việc nhận biết các tác dụng phụ tiềm ẩn và cách sử dụng thuốc đúng cách là rất quan trọng để tránh các tác dụng phụ không mong muốn và đảm bảo điều trị an toàn .

## **Định nghĩa bài toán cho người không cùng chuyên môn**

Vì nhiều loại thuốc trong y học có thành phần và tác dụng tương tự nhau, nên việc phân biệt chúng là điều khó khăn nếu không có kiến thức chuyên môn . Có rất nhiều loại thuốc trông giống nhau nhưng khác biệt ở một vài đặc điểm cụ thể như thành phần hoạt tính, liều lượng, hoặc tác dụng điều trị . Để xác định một loại thuốc có tác dụng phụ không mong muốn hay không, không chỉ cần biết về tên loại thuốc mà còn cần hiểu rõ về thành phần và cách hoạt động của nó . Chính vì những thách thức này mà nhiều công trình nghiên cứu trong y học và khoa học sức khỏe đều quan tâm đến nhiệm vụ phân loại thuốc để đảm bảo an toàn cho người dùng. Dự án này được thực hiện nhằm phát triển và nghiên cứu mô hình để phân loại thuốc theo thành phần và tác dụng, giúp nhận diện và phòng tránh các tác dụng phụ bằng cách sử dụng các thuật toán học máy.

## **Định nghĩa bài toán cho người có cùng chuyên môn**

Bài toán phân loại thuốc dựa trên thành phần và tác dụng là một vấn đề trong lĩnh vực xử lý dữ liệu và học máy , nhằm xác định loại thuốc và phân loại tác dụng phụ dựa trên các đặc trưng của chúng . Bằng cách thu thập và đo lường các thuộc tính của thuốc, như thành phần , tác dụng , chúng ta sẽ áp dụng các thuật toán học máy và thống kê như Random Forest , Decision Tree , và K-Nearest Neighbors (KNN) , để xây dựng mô hình phân loại. Quá trình này bao gồm việc tiền xử lý dữ liệu, lựa chọn đặc trưng , và tinh chỉnh tham số để đảm bảo độ chính xác cao cho mô hình.

## **Giải thiết của bài toán là gì ?**

Trong dự án này, chúng ta sẽ sử dụng bộ dữ liệu về phân loại thuốc từ các nguồn dữ liệu mở hoặc cơ sở dữ liệu y tế và sẽ phân tích, dự đoán tác dụng phụ của thuốc dựa trên các thành phần và tác dụng của chúng bằng các thuật toán học máy. Dữ liệu đầu vào sẽ bao gồm các thuộc tính như tên thuốc , thành phần , tác dụng , nhà sản xuất , phần trăm đánh giá xuất sắc , phần trăm đánh giá trung bình , phần trăm đánh giá kém . Sau khi xử lý dữ liệu, chúng sẽ được đưa vào các mô hình học máy trong thư viện scikit-learn để huấn luyện và đánh giá hiệu suất phân loại .

## **Các bài toán tương tự trong thực tế**

Các bài toán tương tự như phân loại các loại thực phẩm , phân loại hóa chất độc hại

# **II. Sự cần thiết của Project**

## **2.1 Động lực để giải quyết bài toán**

Hiện nay, trên thế giới có hàng triệu loại thuốc khác nhau, nhưng chỉ một phần nhỏ trong số đó là an toàn và hiệu quả cho người sử dụng. Sự gia tăng nhanh chóng của các loại thuốc mới, cùng với sự phát triển của công nghệ y tế, đã tạo ra một thách thức lớn trong việc phân loại và đánh giá tác dụng cũng như tác dụng phụ của chúng. Theo các nghiên cứu, có hàng triệu người gặp phải các tác dụng phụ từ thuốc mỗi năm, dẫn đến những vấn đề nghiêm trọng về sức khỏe .

Chẳng hạn, một nghiên cứu cho thấy rằng hàng triệu ca nhập viện mỗi năm có liên quan đến việc sử dụng thuốc không đúng cách hoặc tác dụng phụ không được phát hiện kịp thời. Do đó, việc phân loại thuốc dựa trên thành phần và tác dụng là nhiệm vụ quan trọng để đảm bảo an toàn cho người sử dụng và nâng cao hiệu quả điều trị.

## **2.2 Giải pháp mang lại những lợi ích gì ?**

Thuốc đóng vai trò rất quan trọng trong lĩnh vực y tế và đời sống con người, là nguồn cung cấp các phương pháp điều trị hiệu quả cho nhiều loại bệnh. Dù rằng số lượng thuốc an toàn và hiệu quả ngày càng tăng, nhưng vẫn tồn tại nhiều loại thuốc có tác dụng phụ nghiêm trọng hoặc tương tác không mong muốn. Việc phân loại thuốc dựa trên thành phần và tác dụng sẽ giúp nhận diện và quản lý những rủi ro này, từ đó bảo vệ sức khỏe của người tiêu dùng.

Bằng cách phân biệt giữa các loại thuốc an toàn và thuốc có thể gây hại, chúng ta có thể giảm thiểu nguy cơ ngộ độc và các phản ứng không mong muốn, đồng thời tối ưu hóa hiệu quả điều trị. Điều này không chỉ quan trọng đối với bệnh nhân mà còn góp phần vào việc nâng cao chất lượng dịch vụ chăm sóc sức khỏe nói chung.

## **2.3 Giải pháp được sử dụng như thế nào**

Trong phần này, tác giả sẽ thử nghiệm các thuật toán phân loại khác nhau để xác định các loại thuốc an toàn và thuốc có tác dụng phụ dựa trên các đặc trưng đã biết. Các thuộc tính như thành phần , tác dụng và tác dụng phụ sẽ được sử dụng làm cơ sở để xây dựng mô hình phân loại.

Các thuật toán học máy như Random Forest , Decision Tree , và K-Nearest Neighbors (KNN) sẽ được áp dụng để phân tích dữ liệu. Quá trình này sẽ bao gồm việc tiền xử lý dữ liệu, chọn lọc các đặc trưng quan trọng, huấn luyện mô hình và đánh giá độ chính xác của từng phương pháp. Mục tiêu là phát triển một mô hình có khả năng phân loại hiệu quả, từ đó hỗ trợ trong việc xác định các loại thuốc an toàn cho người sử dụng.

Quy trình của giải pháp :

● Bước 1 : Nhập các thư viện cần thiết, Nhập bộ dữ liệu phân loại thuốc .

● Bước 2 : Xử lý trước dữ liệu để loại bỏ dữ liệu bị thiếu .

● Bước 3 : Thực hiện phân chia tỷ lệ phần trăm 80% để chia tập dữ liệu thành tập huấn luyện và 20% cho tập kiểm tra .

● Bước 4 : Chọn thuật toán học máy như Random Forest , Decision Tree , và K-Nearest Neighbors (KNN) .

● Bước 5 : Xây dựng mô hình phân loại cho thuật toán học máy đã đề cập dựa trên tập huấn luyện .

● Bước 6 : Kiểm tra mô hình trình phân loại cho thuật toán học máy đã đề cập dựa trên tập kiểm tra .

● Bước 7 : Thực hiện so sánh đánh giá các kết quả hoạt động thử nghiệm thu được đối với mỗi bộ phân loại .

● Bước 8 : Thựchiện điều chỉnh tham số từ tham số mặc định trong mô hình phân loại dựa trên các biện pháp khác nhau .

● Bước 9 : Thực hiện so sánh đánh giá các kết quả với mô hình phân loại từ tham số mặc định, kết luận thuật toán hoạt động tốt nhất .

# **III. Giải pháp thủ công cho bài toán**

## **3.1 Giải pháp hiện tại của bài toán**

Trong dân gian, người ta thường nhận biết các loại thuốc dựa trên kinh nghiệm sử dụng và các dấu hiệu bên ngoài như bao bì, nhãn mác và màu sắc của thuốc. Tuy nhiên, việc phân biệt giữa thuốc an toàn và thuốc có tác dụng phụ tiềm ẩn không phải lúc nào cũng dễ dàng. Ngay cả các chuyên gia cũng có thể gặp khó khăn trong việc nhận diện, vì nhiều thuốc có thể có thành phần tương tự nhưng lại có tác dụng khác nhau, dẫn đến sự nhầm lẫn trong quá trình kê đơn.

Một số thuốc có thể gây tác dụng phụ nghiêm trọng nhưng lại được người dùng biết đến như là thuốc an toàn chỉ vì chúng có cùng thành phần hoặc hình thức tương tự với những loại thuốc khác. Điều này nhấn mạnh sự cần thiết phải áp dụng các phương pháp phân loại chính xác dựa trên dữ liệu và thuật toán học máy để giảm thiểu rủi ro và bảo vệ sức khỏe người sử dụng.

## **3.2 Giải pháp CNTT cho bài toán**

Bài toán phân loại thuốc có thể được giải quyết bằng cách sử dụng các công nghệ và kỹ thuật tiên tiến. Đầu tiên, cần xây dựng một hệ thống hiệu quả để thu thập dữ liệu về các loại thuốc, đảm bảo rằng dữ liệu đủ đa dạng và phong phú. Điều này bao gồm việc thu thập thông tin về thành phần , tác dụng và các tác dụng phụ đã được ghi nhận.

Sau đó, chúng ta có thể áp dụng các phương pháp machine learning và deep learning để tiến hành phân loại thuốc. Các thuật toán như Random Forest , Decision Tree , và K-Nearest Neighbors (KNN) có thể được sử dụng để tạo ra các mô hình phân loại hiệu quả. Bên cạnh đó, việc kết hợp các kỹ thuật xử lý dữ liệu như phân tích văn bản từ báo cáo y tế và tài liệu khoa học sẽ giúp tăng cường độ chính xác và hiệu suất của mô hình phân loại thuốc.

# **IV. Lấy dữ liệu**

## **4.1 Tải dữ liệu**

****

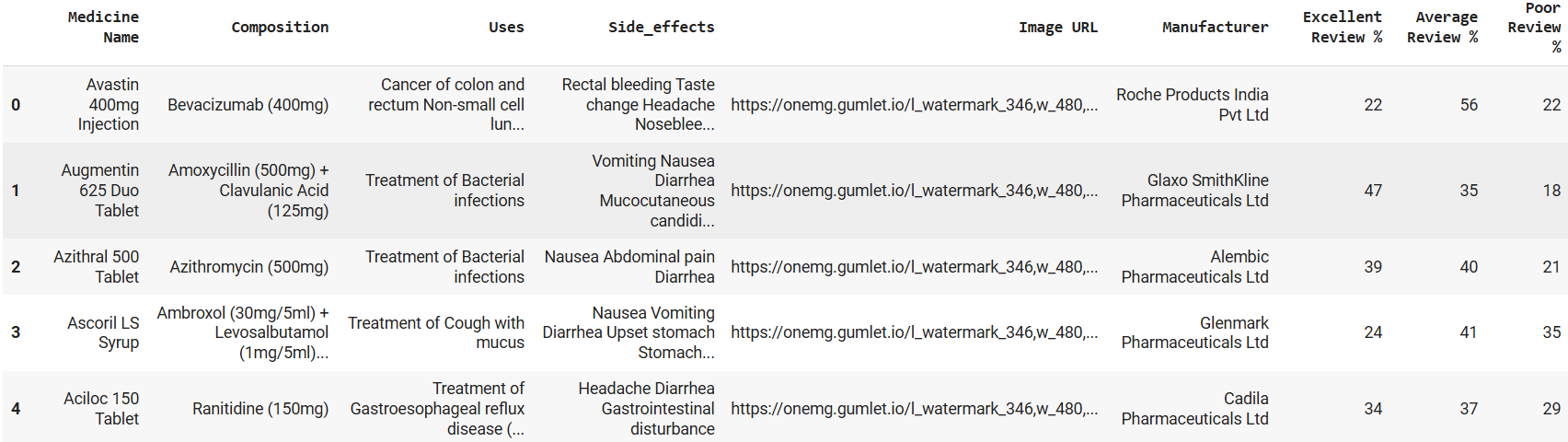
**Hình 4.1 Tải dữ liệu**

## **4.2 Mô tả dữ liệu**

****

Lệnh Medicine\_Details.head() trong Python (đặc biệt là khi dùng thư viện pandas) được sử dụng để xem nhanh một vài dòng đầu tiên của DataFrame Medicine\_Details. Mặc định, head() sẽ hiển thị 5 dòng đầu.

Bộ dữ liệu được sử dụng cho bài toán này là 11000 Medicine details công khai trên trang web Kaggle . Bộ dữ liệu này là một tài nguyên quý giá cho các chuyên gia y tế, nhà khoa học dữ liệu và những người đam mê quan tâm đến việc khám phá thế giới của các loại thuốc và sản phẩm chăm sóc sức khỏe. Nó chứa một kho thông tin phong phú được thu thập từ 1mg, một nền tảng dược phẩm và chăm sóc sức khỏe trực tuyến phổ biến, với hơn 11.000 loại thuốc .



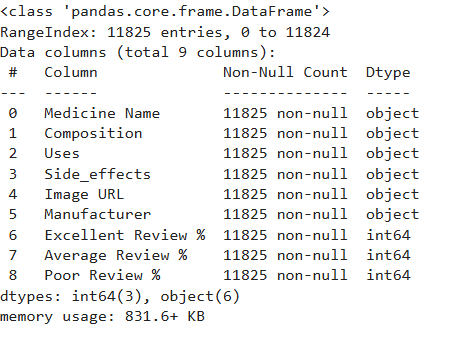
**Bảng 4.2.1 Một mẫu dữ liệu nhỏ được lấy ra từ bộ dữ liệu.**

Dữ liệu bao gồm 9 cột thể hiện cho 9 đặc trưng khác nhau của từng loại thuốc .

****

Lệnh Medicine\_Details.info() trong Python (khi dùng thư viện pandas) cung cấp thông tin chi tiết về DataFrame Medicine\_Details, bao gồm:

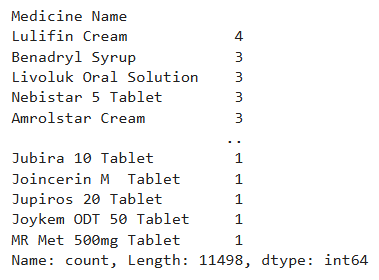
* Số lượng dòng và cột trong DataFrame.
* Tên các cột và kiểu dữ liệu của từng cột .
* Số lượng giá trị không rỗng (non-null) trong mỗi cột, giúp xác định các cột có dữ liệu bị thiếu (missing values) .
* Dung lượng bộ nhớ mà DataFrame chiếm dụng .



**Hình 4.2.2 Thể hiện lệnh .info()**

****

Lệnh Medicine\_Details["Medicine Name"].value\_counts() trong Python (với thư viện pandas) được sử dụng để đếm số lần xuất hiện của từng giá trị duy nhất trong cột "Medicine Name" của DataFrame Medicine\_Details.

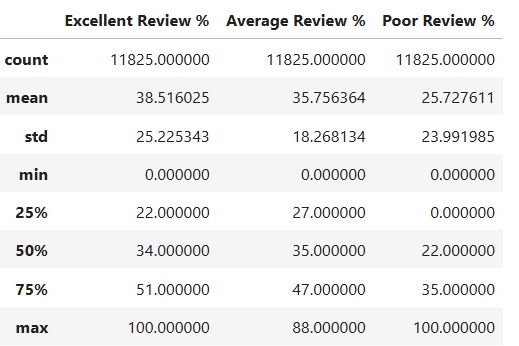


**Hình 4.2.3 Thể hiện lệnh .value\_counts()**



Lệnh Medicine\_Details.describe() trong Python (với thư viện **pandas**) được sử dụng để tạo ra các thống kê cơ bản về dữ liệu số trong **DataFrame** Medicine\_Details. Khi chạy lệnh này, bạn sẽ nhận được các thông tin như:

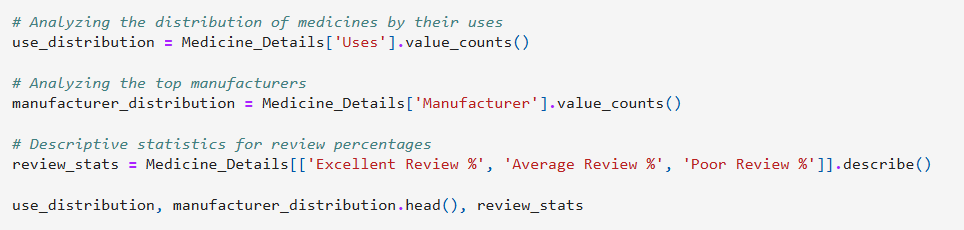
* **Count**: Số lượng giá trị không bị thiếu trong mỗi cột.
* **Mean**: Giá trị trung bình của mỗi cột.
* **Standard Deviation (std)**: Độ lệch chuẩn, cho biết mức độ phân tán của dữ liệu so với trung bình.
* **Minimum (min)** và **Maximum (max)**: Giá trị nhỏ nhất và lớn nhất của mỗi cột.
* **Percentiles (25%, 50%, 75%)**: Các phần trăm vị, giúp hiểu về sự phân bố dữ liệu .



**Hình 4.2.4 Thể hiện lệnh .describe()**

# **V. Khám phá và trực quan hóa dữ liệu**

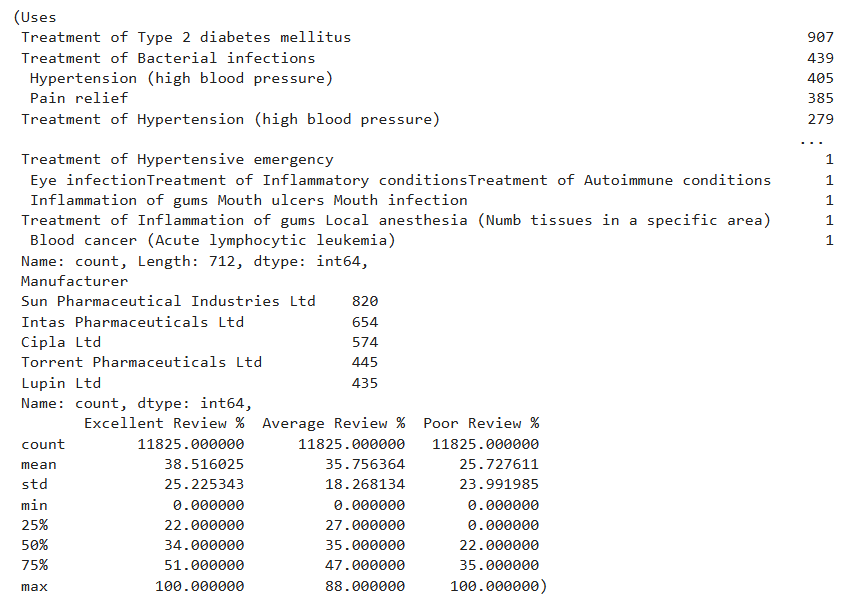
## **5.1 Trực quan hóa dữ liệu**

****

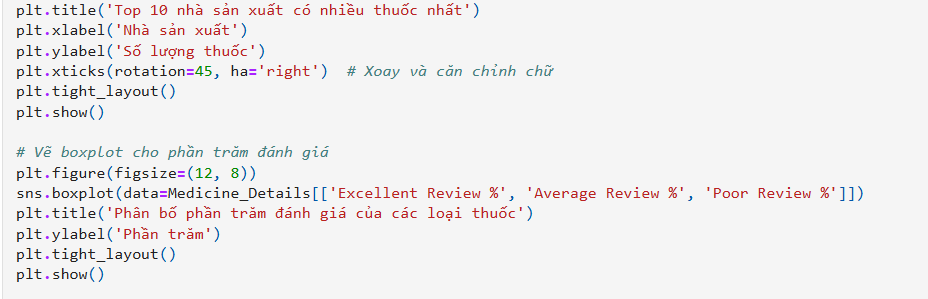
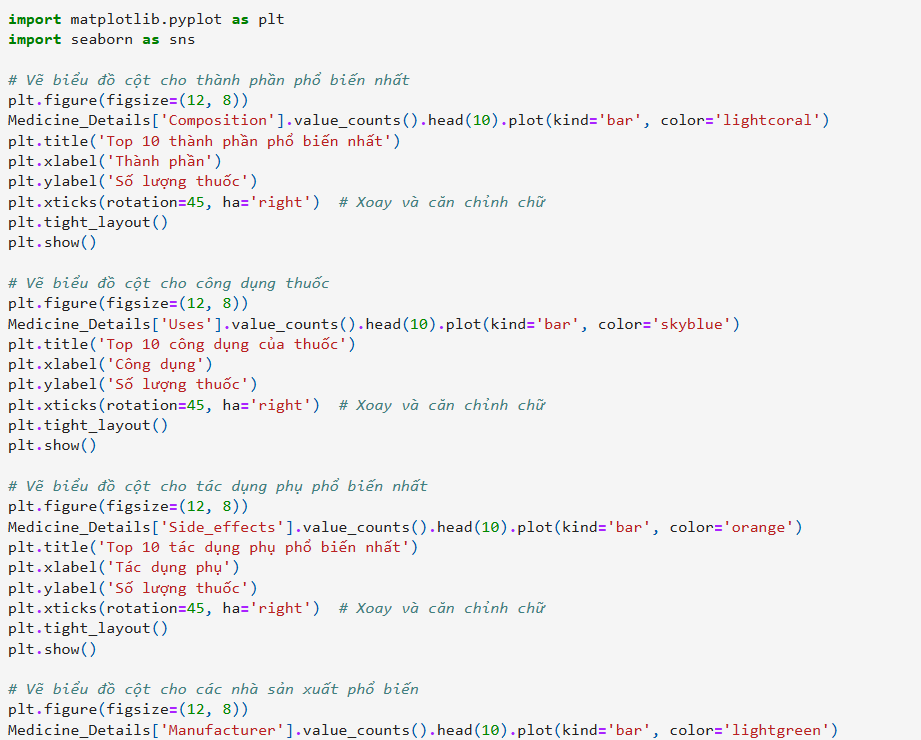
Đoạn code trên đưa ra dùng để phân tích tổng quan dữ liệu thuốc từ ba khía cạnh:

* Phân phối công dụng của thuốc - Xác định các công dụng thuốc phổ biến nhất.
* Phân phối nhà sản xuất hàng đầu - Xác định các nhà sản xuất phổ biến nhất trong dữ liệu.
* Thống kê mô tả về đánh giá thuốc - Cung cấp thông tin thống kê (như trung bình, độ lệch chuẩn) cho các cột đánh giá về mức độ hài lòng của người dùng.

Mục đích chung là để có cái nhìn sơ bộ về phân phối và chất lượng của dữ liệu thuốc.

****

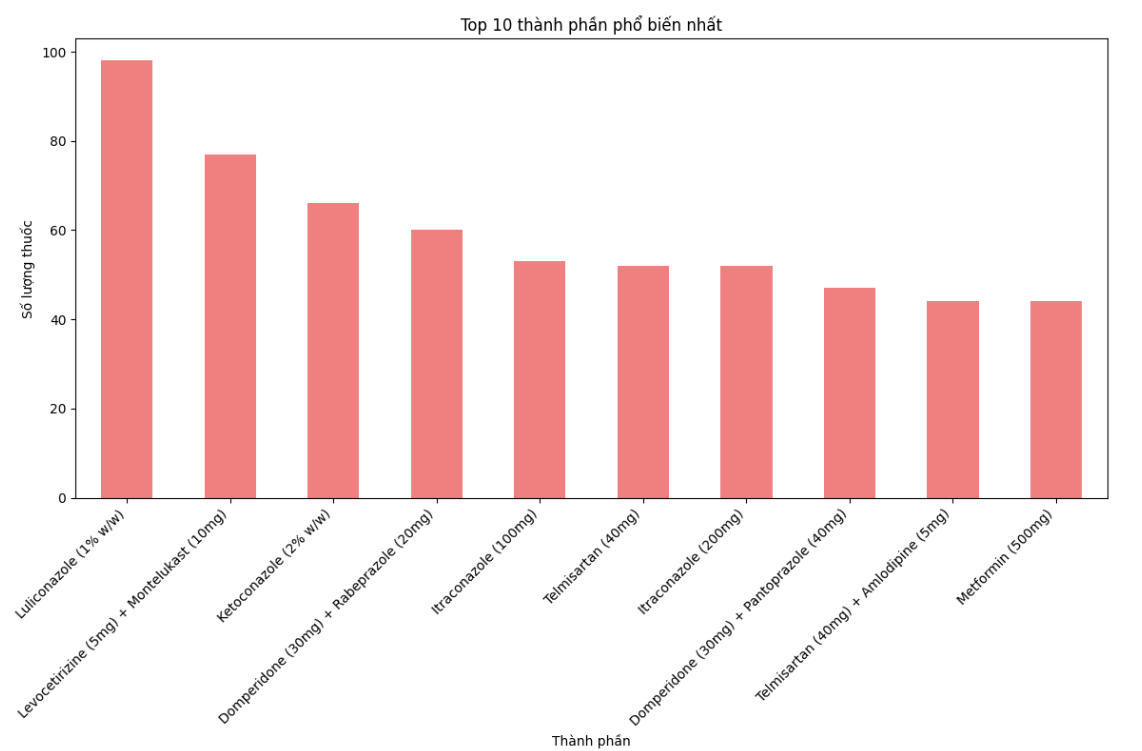
**Hình 5.1.1 Thể hiện đoạn code trên**



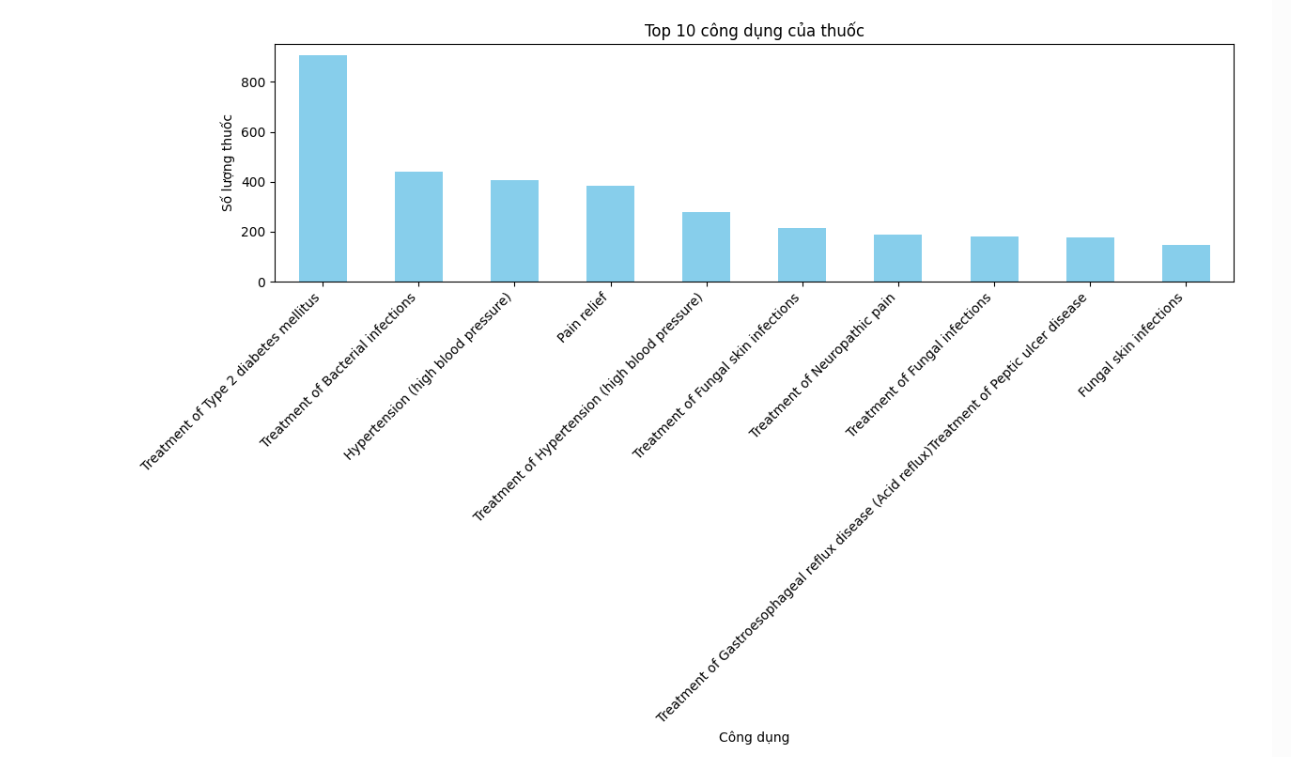
Đoạn mã trên thực hiện một loạt các thao tác trực quan hóa dữ liệu từ bộ dữ liệu Medicine\_Details bằng cách sử dụng thư viện matplotlib và seaborn. Dưới đây là các biểu đồ được vẽ:

* **Biểu đồ cột cho thành phần phổ biến nhất**: Hiển thị 10 thành phần thuốc phổ biến nhất và số lượng thuốc tương ứng.
* **Biểu đồ cột cho công dụng thuốc**: Hiển thị 10 công dụng thuốc phổ biến nhất và số lượng thuốc tương ứng.
* **Biểu đồ cột cho tác dụng phụ phổ biến nhất**: Hiển thị 10 tác dụng phụ phổ biến nhất và số lượng thuốc tương ứng.
* **Biểu đồ cột cho các nhà sản xuất phổ biến**: Hiển thị 10 nhà sản xuất có số lượng thuốc nhiều nhất.
* **Boxplot cho phần trăm đánh giá**: Hiển thị phân bố phần trăm đánh giá (xuất sắc, trung bình, kém) của các loại thuốc.

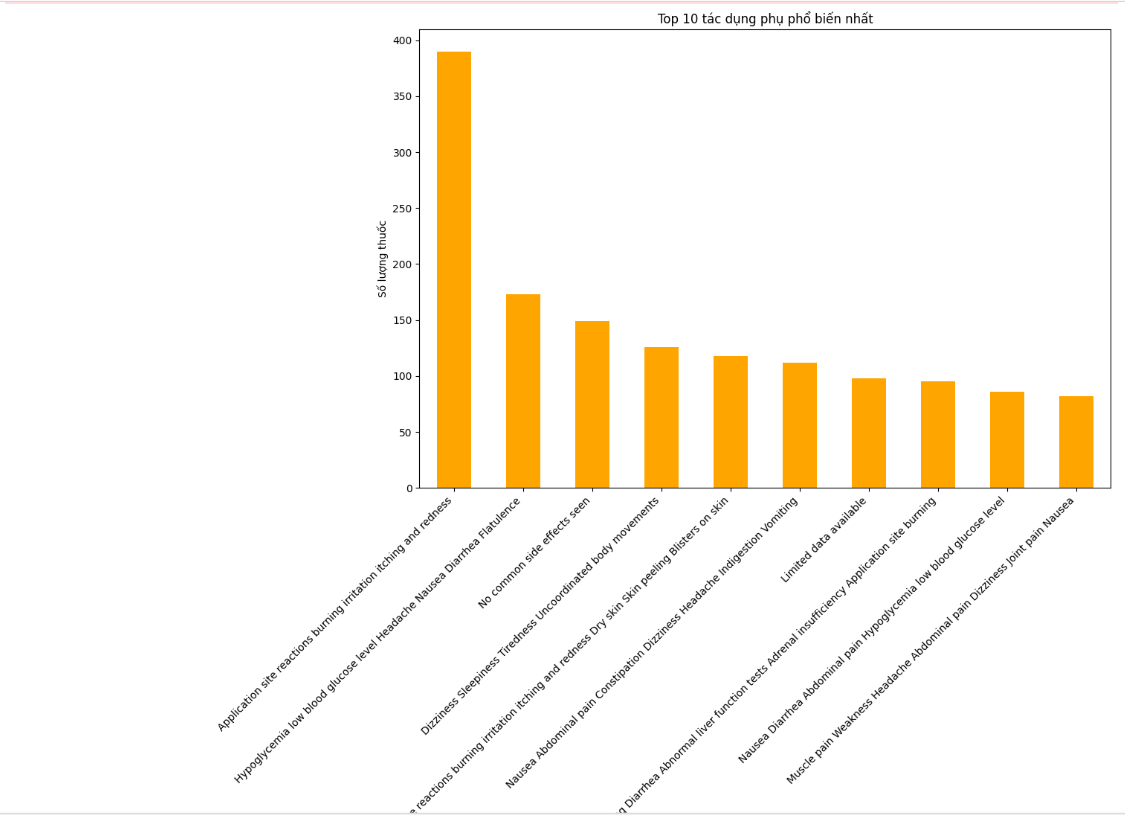
Những biểu đồ này giúp người xem có cái nhìn tổng quan về thông tin và xu hướng liên quan đến thành phần, công dụng, tác dụng phụ và nhà sản xuất thuốc trong bộ dữ liệu.



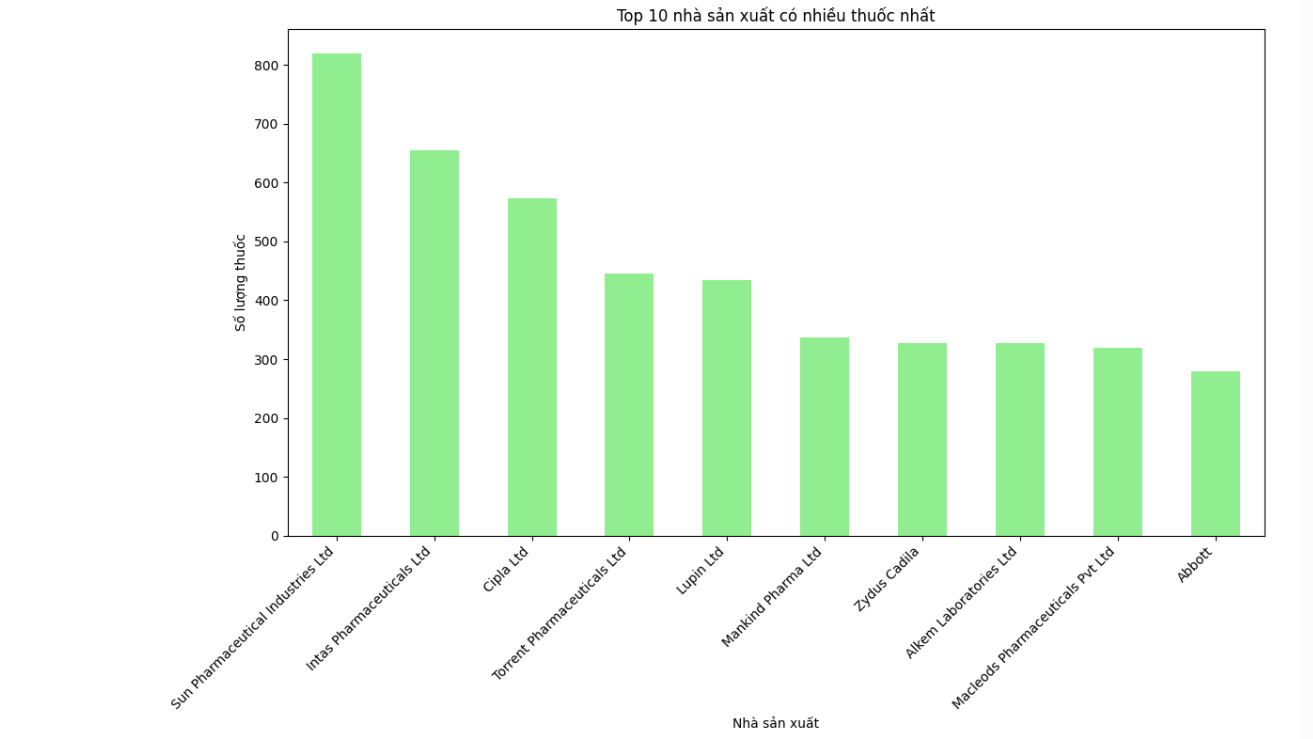
**Hình 5.1.2 Biểu đồ top 10 thành phần phổ biến nhất**

****

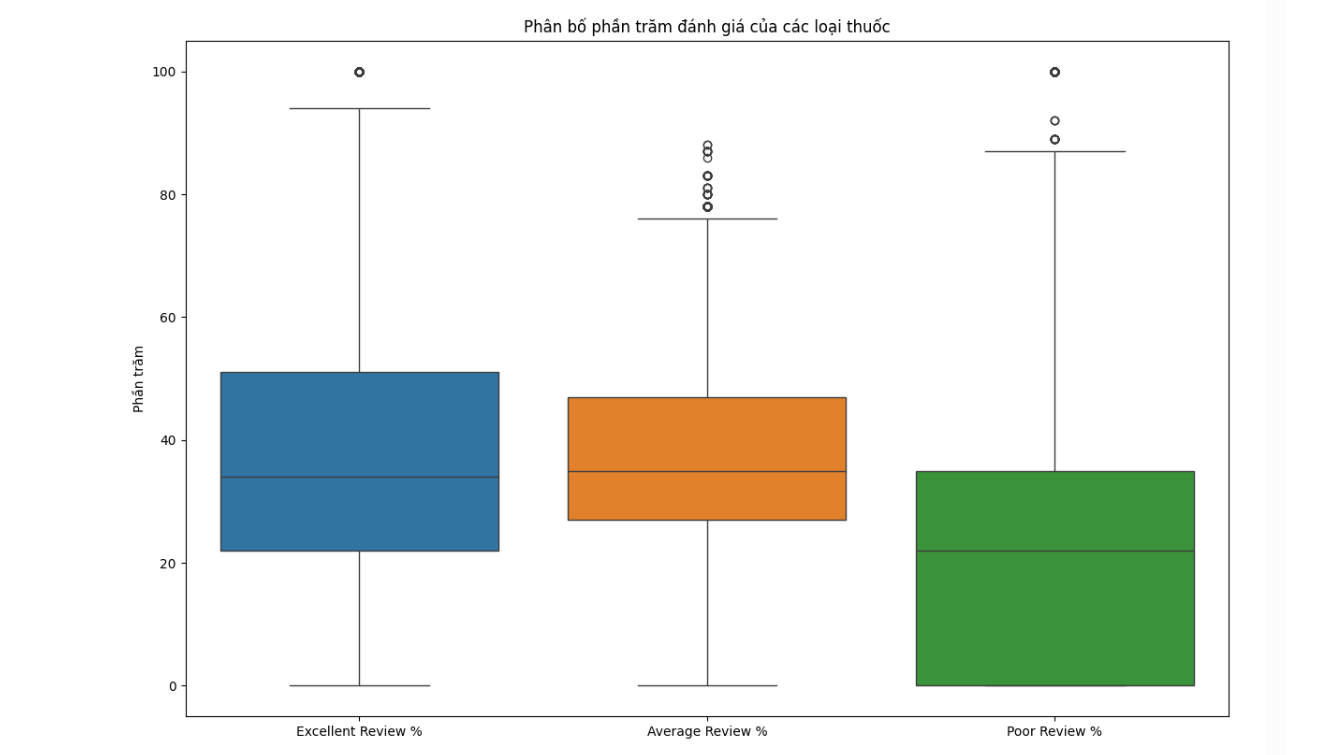
**Hình 5.1.3 Biểu đồ top 10 công dụng của thuốc**

****

**Hình 5.1.4 Biểu đồ top 10 tác dụng phụ phổ biến nhất**

****

**Hình 5.1.5 Biểu đồ top 10 nhà sản xuất có nhiều thuốc nhất**

****

**Hình 5.1.6 Thể hiện phân bố phần trăm đánh giá của các loại thuốc**

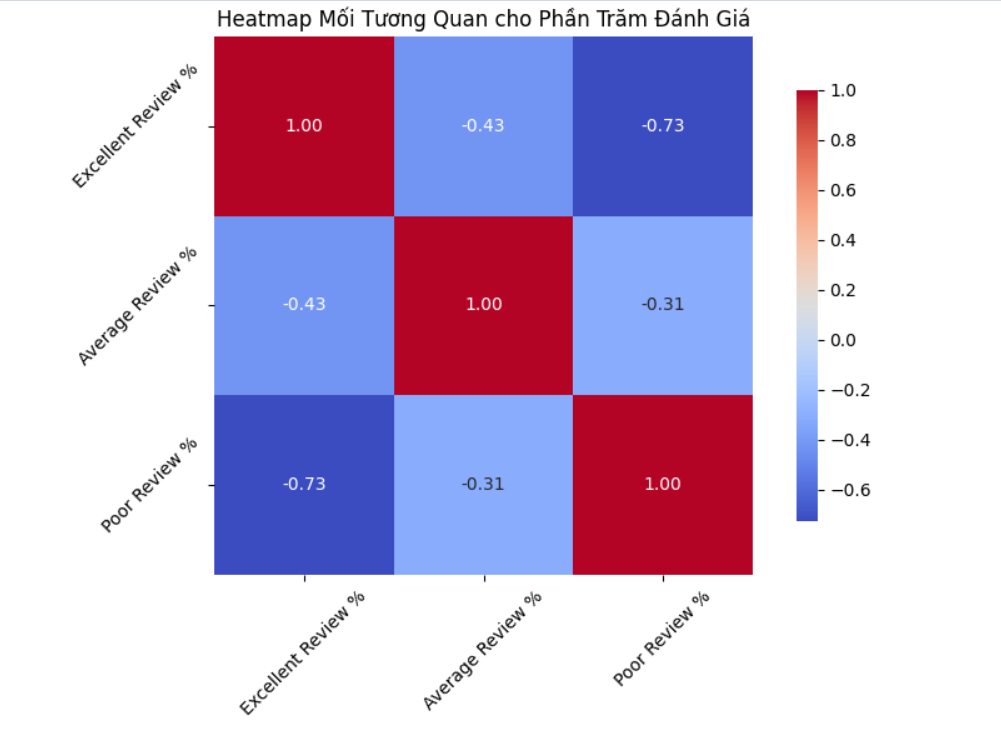
## **5.2 Tìm kiếm mối tương quan**

****

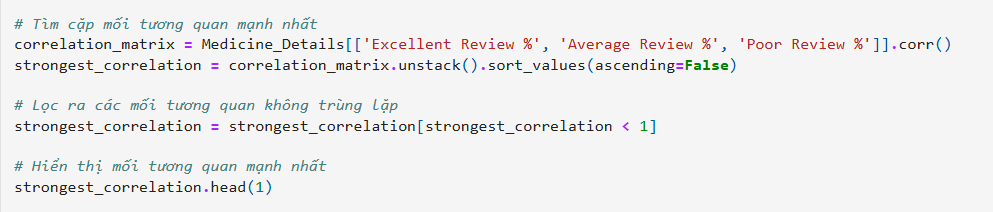
Đoạn mã trên thực hiện các bước sau để tìm kiếm mối tương quan giữa các phần trăm đánh giá của thuốc trong bộ dữ liệu Medicine\_Details:

* Tính toán ma trận tương quan: Sử dụng phương thức corr() để tính toán ma trận tương quan giữa các cột phần trăm đánh giá: Excellent Review %, Average Review %, và Poor Review %.
* Thiết lập hình vẽ: Thiết lập kích thước cho biểu đồ.
* Vẽ heatmap: Sử dụng seaborn để vẽ biểu đồ heatmap, hiển thị ma trận tương quan. Các giá trị tương quan được chú thích trên biểu đồ với định dạng hai chữ số thập phân.
* Thêm tiêu đề và nhãn: Thiết lập tiêu đề cho biểu đồ và điều chỉnh góc độ chữ cho các nhãn.

Biểu đồ heatmap này giúp người xem dễ dàng nhận biết mối quan hệ giữa các phần trăm đánh giá, từ đó hiểu rõ hơn về cách mà các đánh giá khác nhau có thể liên quan đến nhau.



**Hình 5.2.1 Heatmap mối tương quan cho phần trăm đánh giá**

****

Đoạn mã trên thực hiện các bước sau để tìm cặp mối tương quan mạnh nhất giữa các phần trăm đánh giá trong bộ dữ liệu Medicine\_Details:

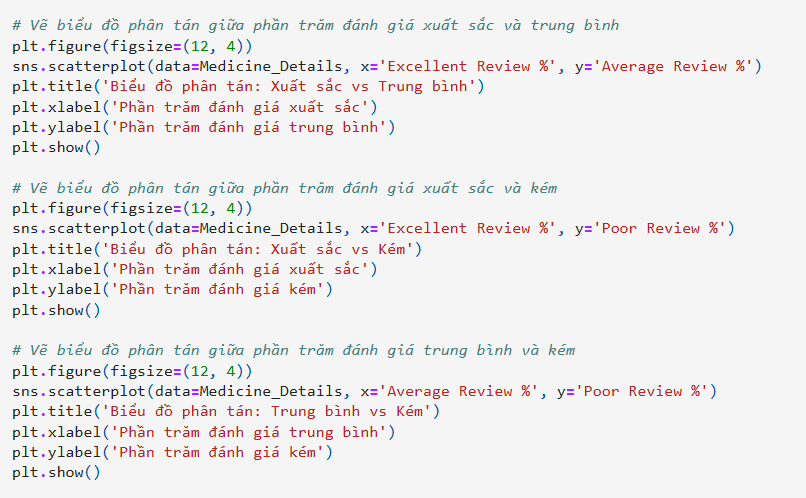
* Tính toán ma trận tương quan: Sử dụng phương thức corr() để tính toán ma trận tương quan giữa các cột Excellent Review %, Average Review %, và Poor Review %.
* Lấy và sắp xếp mối tương quan: Chuyển đổi ma trận tương quan thành một Series không có cấu trúc (unstack()), sau đó sắp xếp các giá trị theo thứ tự giảm dần.
* Lọc mối tương quan không trùng lặp: Loại bỏ các mối tương quan có giá trị bằng 1 (tương quan hoàn toàn).
* Hiển thị mối tương quan mạnh nhất: Lấy và hiển thị cặp mối tương quan mạnh nhất trong số các mối tương quan đã lọc.

Mục tiêu của đoạn mã này là xác định cặp đánh giá nào có mối tương quan mạnh nhất, giúp hiểu rõ hơn về mối liên hệ giữa các loại đánh giá trong bộ dữ liệu.



**Hình 5.2.2 Thể hiện đoạn mã trên**

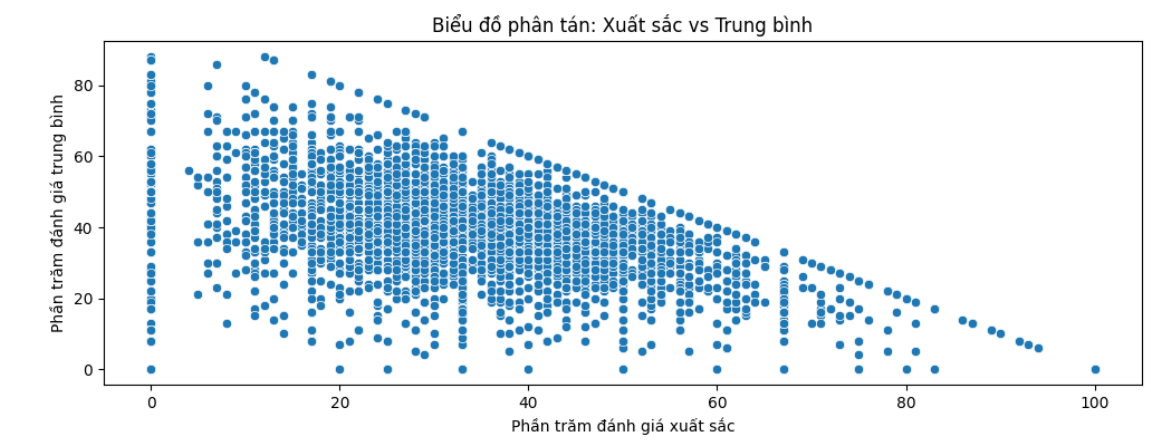
## **5.3 Thử nghiệm với các kết hợp thuộc tính**

****

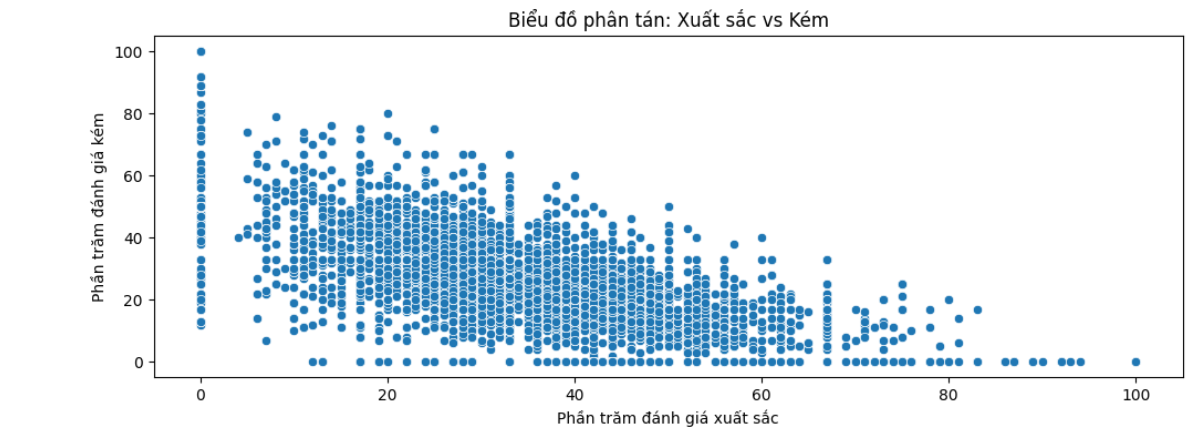
Đoạn mã trên thực hiện các bước sau để vẽ biểu đồ phân tán giữa các phần trăm đánh giá trong bộ dữ liệu Medicine\_Details:

* Biểu đồ phân tán giữa phần trăm đánh giá xuất sắc và trung bình:
  + Vẽ biểu đồ phân tán với Excellent Review % là trục x và Average Review % là trục y.
  + Tiêu đề biểu đồ là "Xuất sắc vs Trung bình".
* Biểu đồ phân tán giữa phần trăm đánh giá xuất sắc và kém:
  + Vẽ biểu đồ phân tán với Excellent Review % là trục x và Poor Review % là trục y.
  + Tiêu đề biểu đồ là "Xuất sắc vs Kém".
* Biểu đồ phân tán giữa phần trăm đánh giá trung bình và kém:
  + Vẽ biểu đồ phân tán với Average Review % là trục x và Poor Review % là trục y.
  + Tiêu đề biểu đồ là "Trung bình vs Kém".

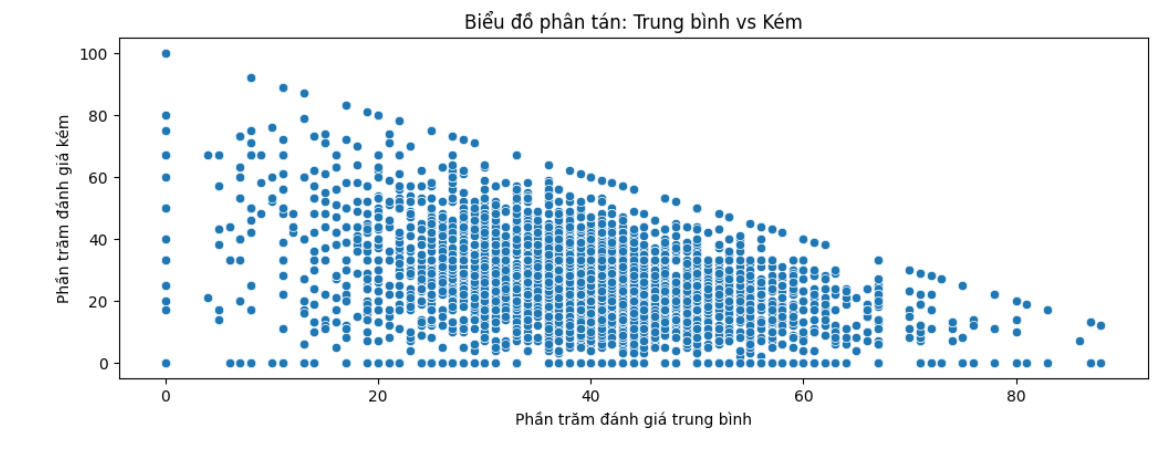
Mục tiêu của các biểu đồ này là để trực quan hóa mối quan hệ giữa các loại đánh giá khác nhau (xuất sắc, trung bình, kém) trong bộ dữ liệu, giúp hiểu rõ hơn về cách chúng tương tác với nhau.



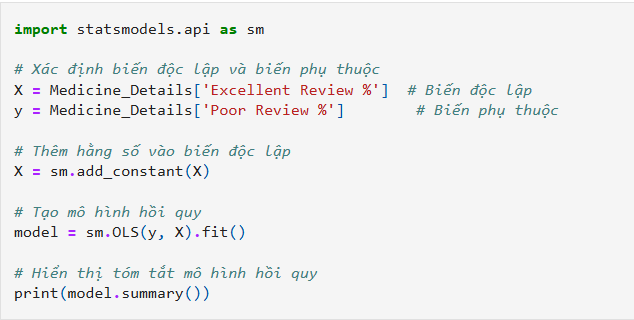
**Hình 5.3.1 Biểu đồ phân tán : Xuất sắc và Trung bình**

****

**Hình 5.3.2 Biểu đồ phân tán : Xuất sắc và Kém**

****

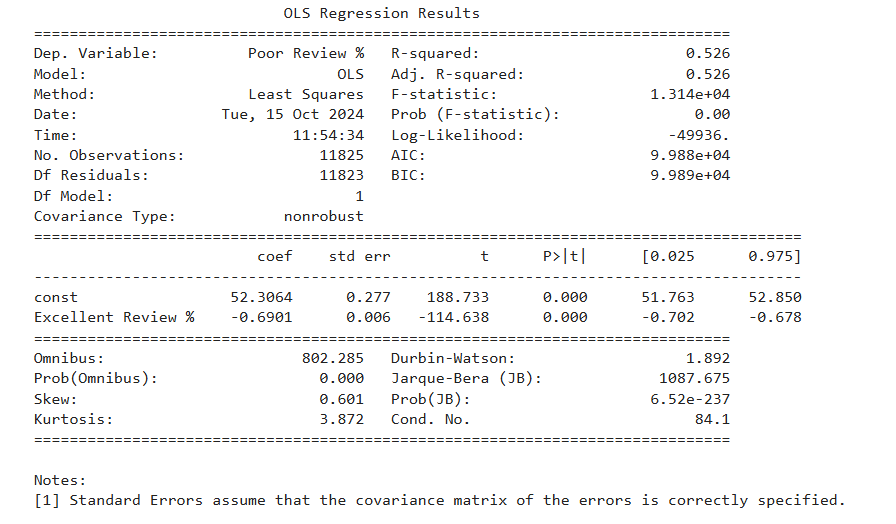
**Hình 5.3.3 Biểu đồ phân tán : Trung bình và Kém**

****

Đoạn mã trên thực hiện các bước sau để xây dựng một mô hình hồi quy tuyến tính sử dụng thư viện statsmodels:

* Xác định biến độc lập và biến phụ thuộc: X được xác định là phần trăm đánh giá xuất sắc (Excellent Review %), trong khi y là phần trăm đánh giá kém (Poor Review %).
* Thêm hằng số vào biến độc lập: Sử dụng sm.add\_constant(X) để thêm một hằng số vào biến độc lập, điều này cần thiết cho mô hình hồi quy tuyến tính.
* Tạo mô hình hồi quy: Sử dụng phương pháp OLS (Ordinary Least Squares) để xây dựng mô hình hồi quy tuyến tính với y là biến phụ thuộc và X là biến độc lập.
* Hiển thị tóm tắt mô hình hồi quy: Sử dụng model.summary() để hiển thị các thông tin tóm tắt về mô hình hồi quy, bao gồm các hệ số, giá trị p, R-squared và các thông số khác.

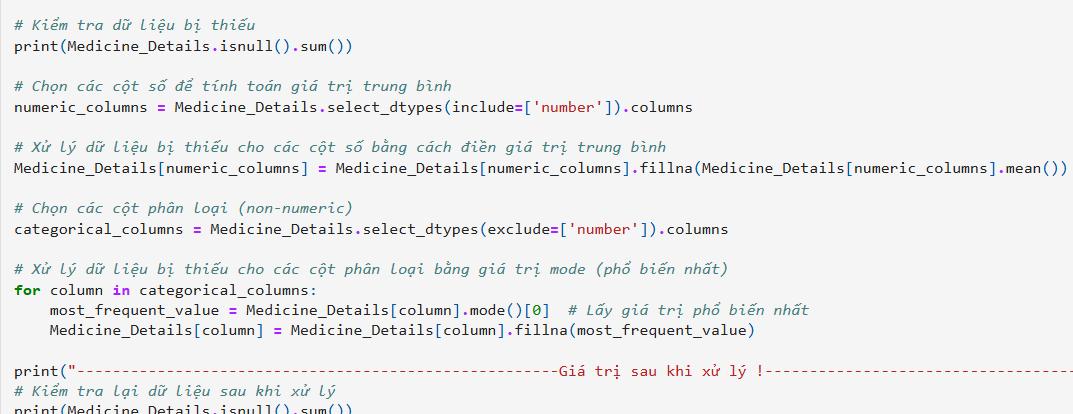
Mục tiêu của đoạn mã này là phân tích mối quan hệ giữa phần trăm đánh giá xuất sắc và phần trăm đánh giá kém, từ đó hiểu rõ hơn về ảnh hưởng của đánh giá xuất sắc đến đánh giá kém.

****

**Hình 5.3.4 Thể hiện đoạn mã trên**

# **VI Chuẩn bị dữ liệu cho các thuật toán học máy**

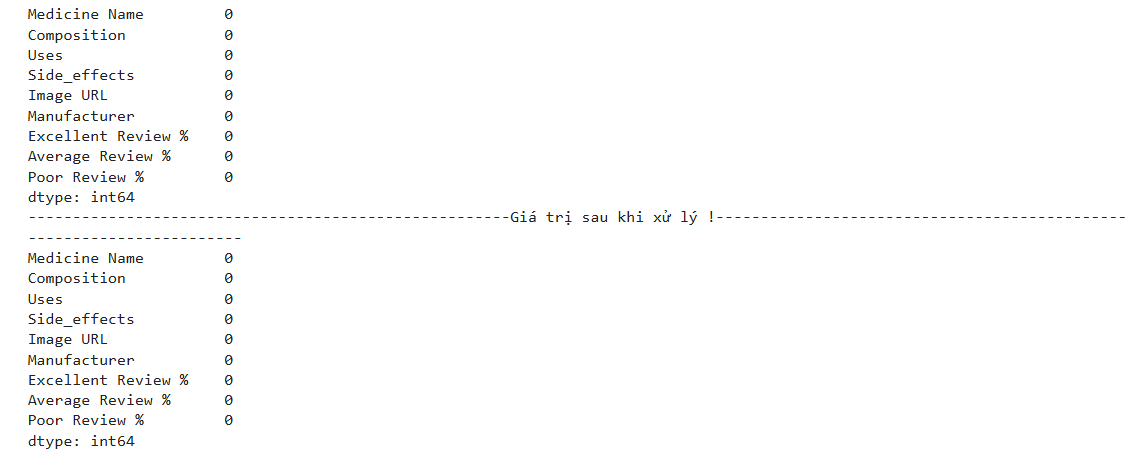
## **6.1 Làm sạch dữ liệu**

****

Đoạn mã trên thực hiện các bước sau để làm sạch dữ liệu trong bộ dữ liệu Medicine\_Details:

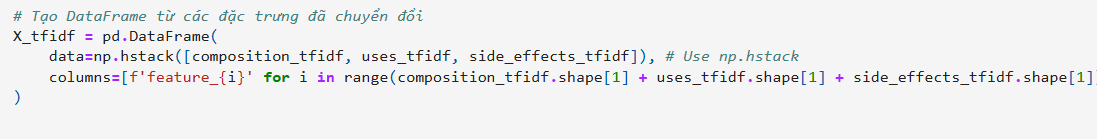
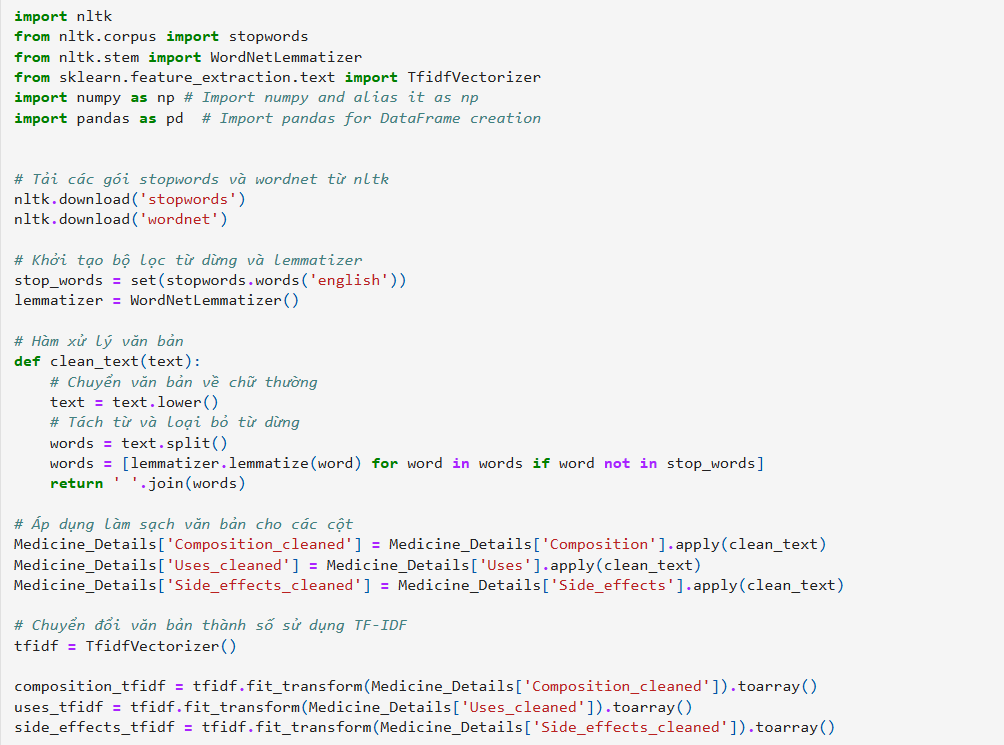
* Kiểm tra dữ liệu bị thiếu: Sử dụng isnull().sum() để đếm và hiển thị số lượng giá trị bị thiếu trong từng cột của bộ dữ liệu.
* Chọn các cột số: Xác định các cột chứa dữ liệu số để tính toán giá trị trung bình.
* Xử lý dữ liệu bị thiếu cho các cột số: Điền các giá trị bị thiếu trong các cột số bằng giá trị trung bình của mỗi cột.
* Chọn các cột phân loại (non-numeric): Xác định các cột không phải số (các cột phân loại) trong bộ dữ liệu.
* Xử lý dữ liệu bị thiếu cho các cột phân loại: Điền các giá trị bị thiếu trong các cột phân loại bằng giá trị phổ biến nhất (mode) của mỗi cột.
* Kiểm tra lại dữ liệu sau khi xử lý: Hiển thị số lượng giá trị bị thiếu trong từng cột sau khi đã xử lý.

Mục tiêu của đoạn mã này là làm sạch dữ liệu bằng cách xử lý các giá trị bị thiếu, đảm bảo rằng bộ dữ liệu sẵn sàng để sử dụng trong các thuật toán học máy.



**Hình 6.1 Thể hiện đoạn mã trên**

## **6.2 Xử lý các thuộc tính văn bản và phân loại**

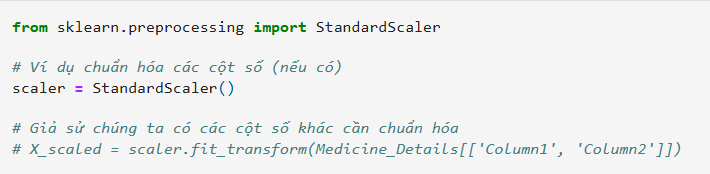
****

Đoạn mã trên thực hiện các bước để xử lý các thuộc tính văn bản trong bộ dữ liệu Medicine\_Details bằng cách sử dụng thư viện nltk và sklearn. Dưới đây là tóm tắt các bước:

* Tải các gói từ thư viện nltk: Tải các bộ từ dừng (stopwords) và từ điển WordNet để sử dụng trong quá trình xử lý văn bản.
* Khởi tạo bộ lọc từ dừng và lemmatizer: Tạo một tập hợp các từ dừng và khởi tạo một đối tượng lemmatizer để giảm các từ về dạng gốc.
* Hàm xử lý văn bản:
  + Chuyển văn bản về chữ thường.
  + Tách văn bản thành các từ và loại bỏ các từ dừng.
  + Sử dụng lemmatizer để giảm các từ về dạng gốc và trả về văn bản đã được làm sạch.
* Áp dụng hàm làm sạch cho các cột: Sử dụng hàm clean\_text để làm sạch các cột Composition, Uses, và Side\_effects, kết quả được lưu vào các cột mới (Composition\_cleaned, Uses\_cleaned, Side\_effects\_cleaned).
* Chuyển đổi văn bản thành số sử dụng TF-IDF:
  + Sử dụng TfidfVectorizer để chuyển đổi các văn bản đã làm sạch thành các vectơ số (đặc trưng) dựa trên trọng số TF-IDF.
  + Tạo các mảng cho các cột composition, uses, và side effects.
* Tạo DataFrame từ các đặc trưng đã chuyển đổi: Tạo một DataFrame mới (X\_tfidf) chứa tất cả các đặc trưng số đã được chuyển đổi từ các cột văn bản.

Mục tiêu của đoạn mã này là xử lý và chuyển đổi các thuộc tính văn bản trong bộ dữ liệu thành định dạng số để có thể sử dụng trong các thuật toán học máy.

## **6.3 Chuẩn hóa và biến đổi đặc trưng**

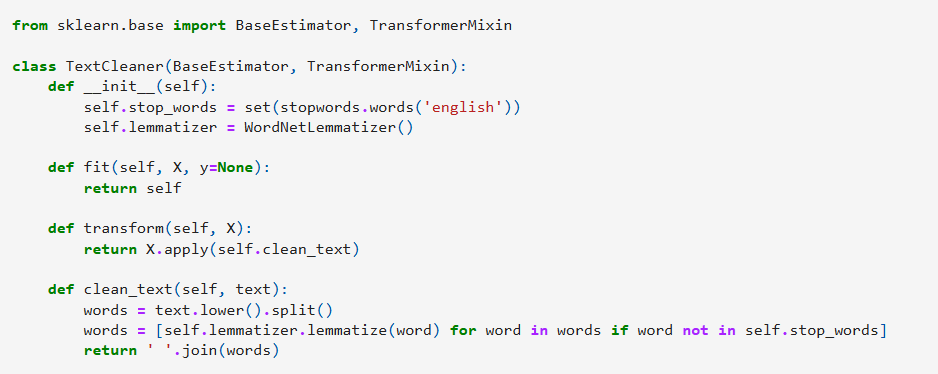
****

Đoạn mã trên thực hiện các bước sau để chuẩn hóa các cột số trong bộ dữ liệu Medicine\_Details:

* Nhập lớp StandardScaler từ sklearn.preprocessing: Lớp này được sử dụng để chuẩn hóa dữ liệu.
* Khởi tạo đối tượng StandardScaler: Tạo một đối tượng scaler để thực hiện việc chuẩn hóa.
* Chuẩn hóa các cột số:
  + Ví dụ giả định là có các cột số (chưa xác định cụ thể ở đây) trong bộ dữ liệu mà cần chuẩn hóa.
  + Sử dụng phương thức fit\_transform() để tính toán các tham số chuẩn hóa (trung bình và độ lệch chuẩn) từ dữ liệu và sau đó áp dụng chuẩn hóa cho các cột đó.

Mục tiêu của đoạn mã này là chuẩn hóa các đặc trưng số trong bộ dữ liệu để chúng có cùng quy mô, giúp cải thiện hiệu suất và độ chính xác của các thuật toán học máy.

## **6.4 Bộ chuyển đổi tùy chỉnh**

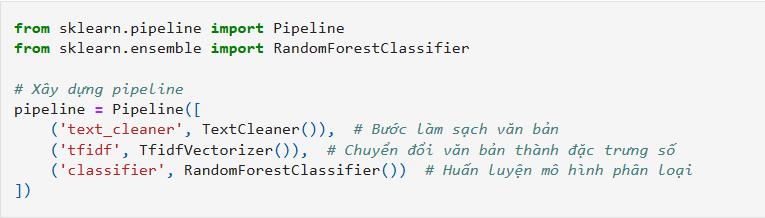
****

Đoạn mã trên định nghĩa một bộ chuyển đổi tùy chỉnh trong scikit-learn để xử lý văn bản, được gọi là TextCleaner. Dưới đây là tóm tắt các bước:

* Nhập các lớp cần thiết: BaseEstimator và TransformerMixin từ sklearn.base, cho phép tạo ra các bộ chuyển đổi tùy chỉnh theo chuẩn của scikit-learn.
* Định nghĩa lớp TextCleaner:
  + Khởi tạo các biến dừng (stop words) và lemmatizer trong phương thức \_\_init\_\_().
* Phương thức fit:
  + Phương thức này được gọi để "học" từ dữ liệu. Trong trường hợp này, nó chỉ trả về chính đối tượng mà không thực hiện bất kỳ phép toán nào (không cần điều chỉnh).
* Phương thức transform:
  + Nhận vào một Series (cột dữ liệu) và áp dụng phương thức clean\_text cho từng giá trị trong Series để làm sạch văn bản.
* Phương thức clean\_text:
  + Làm sạch văn bản bằng cách chuyển nó về chữ thường, tách từ, loại bỏ các từ dừng, và sử dụng lemmatizer để giảm các từ về dạng gốc.

Mục tiêu của đoạn mã này là tạo ra một bộ chuyển đổi tùy chỉnh có thể được tích hợp vào quy trình tiền xử lý dữ liệu trong các pipeline của scikit-learn, giúp xử lý các thuộc tính văn bản một cách tự động và nhất quán.

## **6.5 Pipeline biến đổi**

****

Đoạn mã trên xây dựng một pipeline trong scikit-learn để xử lý và phân loại dữ liệu văn bản. Dưới đây là tóm tắt các bước trong pipeline:

* Nhập các lớp cần thiết: Pipeline từ sklearn.pipeline và RandomForestClassifier từ sklearn.ensemble.
* Xây dựng pipeline: Tạo một đối tượng pipeline bao gồm ba bước:
  + Bước làm sạch văn bản (text\_cleaner): Sử dụng bộ chuyển đổi tùy chỉnh TextCleaner() để làm sạch văn bản.
  + Chuyển đổi văn bản thành đặc trưng số (tfidf): Sử dụng TfidfVectorizer() để chuyển đổi văn bản đã được làm sạch thành các vectơ số dựa trên trọng số TF-IDF.
  + Huấn luyện mô hình phân loại (classifier): Sử dụng RandomForestClassifier() để huấn luyện mô hình phân loại dựa trên các đặc trưng số đã được tạo ra từ văn bản.

Mục tiêu của đoạn mã này là tạo ra một quy trình tự động hóa, nơi dữ liệu văn bản được xử lý và phân loại theo từng bước một cách dễ dàng và có thể tích hợp vào các mô hình học máy khác.

## **6.6 Tạo tập dữ liệu kiểm tra và tập dữ liệu huấn luyện**

## Đoạn mã trên thực hiện các bước để tạo ra các tập dữ liệu huấn luyện và kiểm tra từ bộ dữ liệu Medicine\_Details. Dưới đây là tóm tắt các bước:

* Nhập các lớp cần thiết: train\_test\_split từ sklearn.model\_selection và LabelEncoder từ sklearn.preprocessing.
* Khởi tạo LabelEncoder: Tạo ba đối tượng LabelEncoder để mã hóa các cột văn bản (thành phần, công dụng và tác dụng phụ).
* Mã hóa các cột:
  + Sử dụng fit để học từ các giá trị trong các cột Composition, Uses, và Side\_effects.
  + Lưu các mô hình mã hóa đã được huấn luyện vào các tệp tin (.pkl) để sử dụng sau này.
* Ghi nhớ các cột đã mã hóa: Thêm các cột mới đã được mã hóa (Composition\_encoded, Uses\_encoded, Side\_effects\_encoded) vào bộ dữ liệu.
* Chọn các đặc trưng và cột mục tiêu:
  + Chọn các cột mã hóa (Composition\_encoded, Uses\_encoded) làm đặc trưng (X).
  + Chọn cột Side\_effects\_encoded làm cột mục tiêu (y).
* Chia dữ liệu thành tập huấn luyện và tập kiểm tra: Sử dụng train\_test\_split để chia dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%).
* In ra kích thước của các tập dữ liệu: In ra kích thước của các tập huấn luyện và kiểm tra để xác nhận.

Mục tiêu của đoạn mã này là chuẩn bị dữ liệu để huấn luyện và kiểm tra mô hình học máy, đảm bảo rằng mô hình có thể học từ một phần dữ liệu và được đánh giá trên phần còn lại.

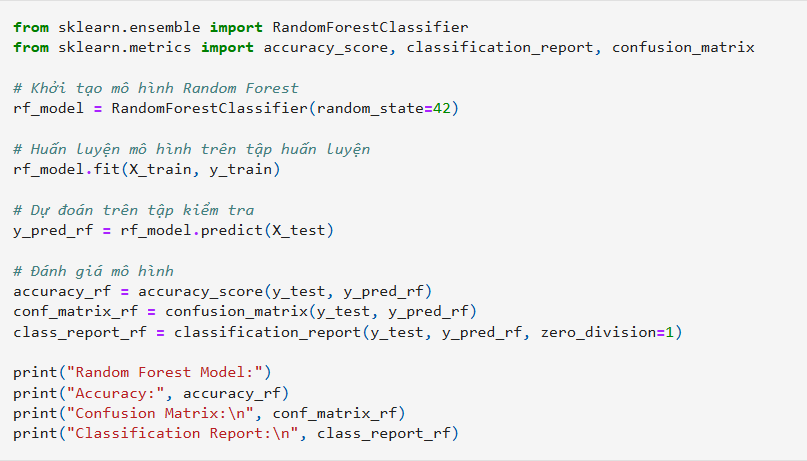


**Hình 6.6 Thể hiện kích thước tập huấn luyện và kiểm tra**

# **VII. Chọn mô hình và huấn luyện nó**

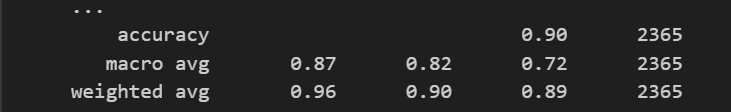
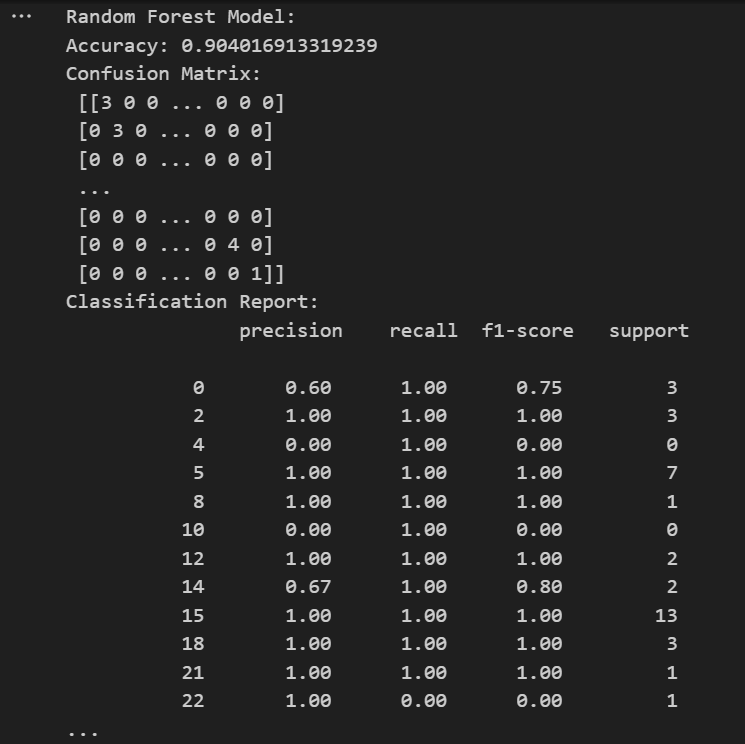
## **7.1 Huấn luyện và đánh giá trên tập huấn luyện**

**a.RandomForest**

****

Đoạn mã trên:

* Khởi tạo và huấn luyện mô hình Random Forest trên tập huấn luyện.
* Dự đoán kết quả trên tập kiểm tra.
* Đánh giá mô hình bằng cách tính độ chính xác (accuracy), ma trận nhầm lẫn (confusion matrix), và báo cáo phân loại (classification report).

Mục tiêu là đánh giá hiệu suất mô hình Random Forest trên tập kiểm tra. 

**Hình 7.1.a Kết quả của thuật toán Random Forest**

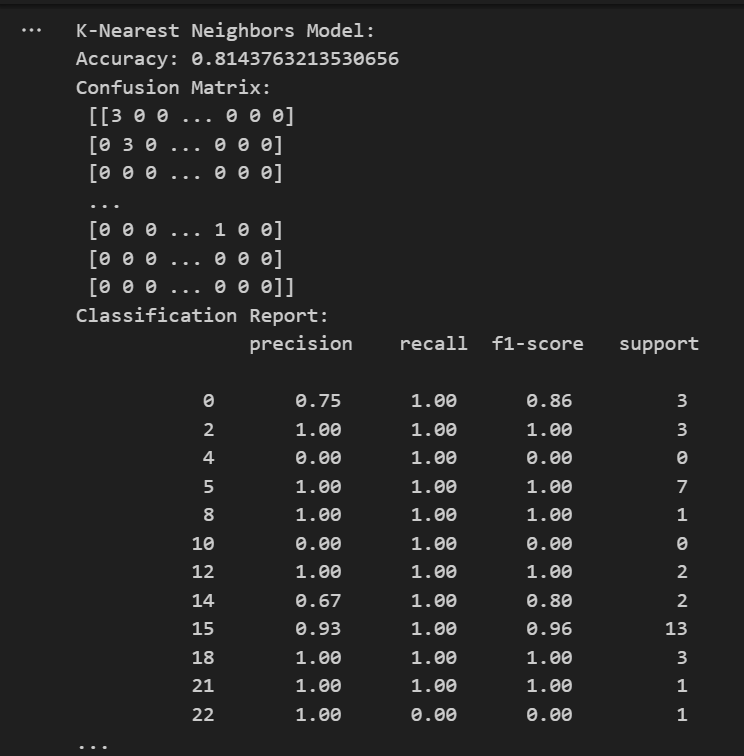
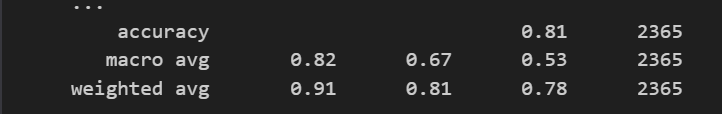
**b.K-NearestNeighbors(KNN)**

****

Đoạn mã trên:

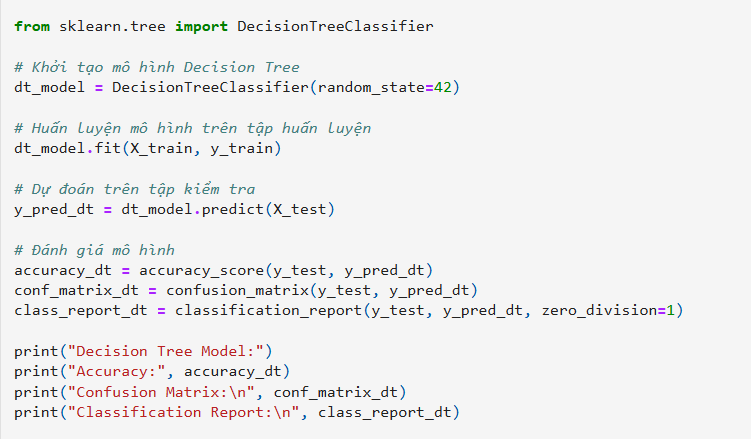
* Khởi tạo và huấn luyện mô hình K-Nearest Neighbors (KNN) với 5 láng giềng gần nhất trên tập huấn luyện.
* Dự đoán kết quả trên tập kiểm tra.
* Đánh giá mô hình bằng cách tính độ chính xác (accuracy), ma trận nhầm lẫn (confusion matrix), và báo cáo phân loại (classification report).

Mục tiêu là đánh giá hiệu suất của mô hình KNN trên tập kiểm tra.

**Hình 7.1.b Kết quả của thuật toán KNN**

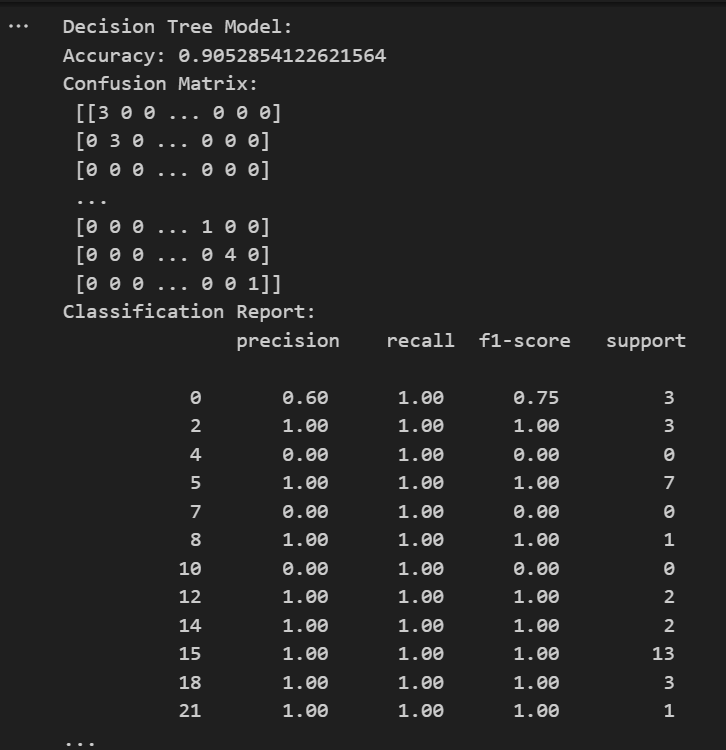
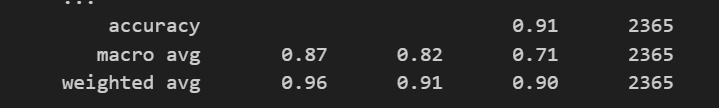
**c.DecisionTree**

****

Đoạn mã trên:

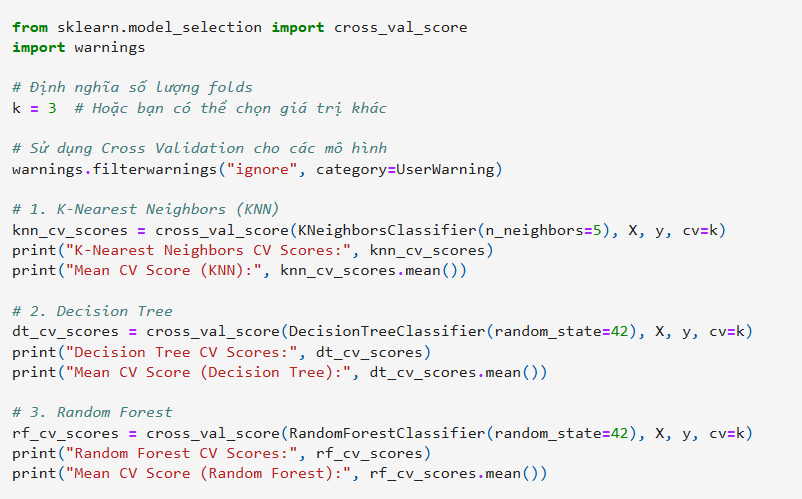
* Khởi tạo và huấn luyện mô hình Decision Tree trên tập huấn luyện.
* Dự đoán trên tập kiểm tra.
* Đánh giá mô hình bằng độ chính xác (accuracy), ma trận nhầm lẫn (confusion matrix), và báo cáo phân loại (classification report).

Mục tiêu là đánh giá hiệu suất của mô hình Decision Tree trên tập kiểm tra.

**** ****

**Hình 7.1.c Kết quả của thuật toán Decision Tree**

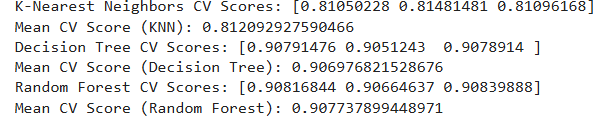
## **7.2 Đánh giá tốt hơn bằng cách sử dụng Cross Validation (Xác thực chéo)**

****

Đoạn mã trên:

* Thiết lập Cross Validation với k=3 folds.
* Thực hiện Cross Validation cho ba mô hình: K-Nearest Neighbors, Decision Tree, và Random Forest.
* Tính điểm trung bình từ các lần chạy của mỗi mô hình và hiển thị kết quả.

Mục tiêu là đánh giá độ chính xác của từng mô hình bằng Cross Validation.

****

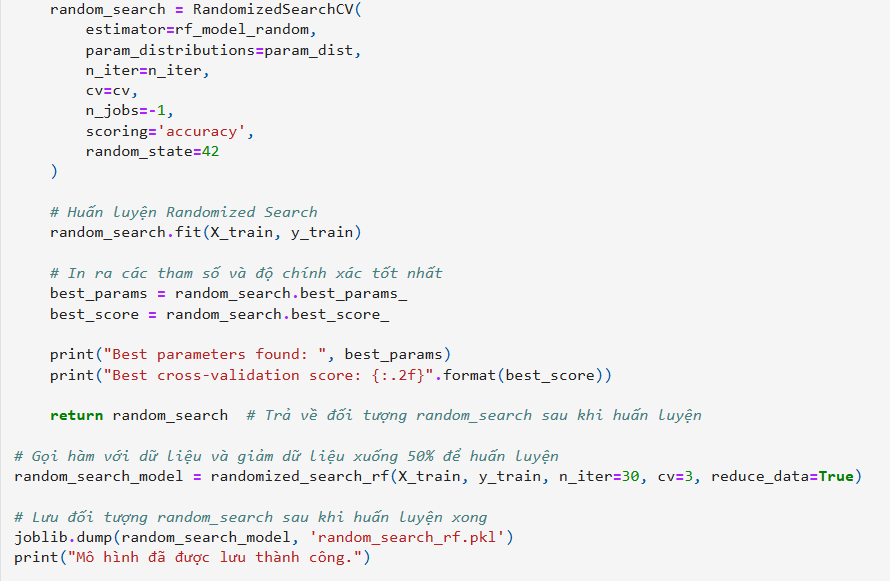
**Hình 7.2 Kết quả khi so sánh 3 thuật toán**

# **VIII. Điều chỉnh mô hình**

**8.1RandomizedSearch**

**A screenshot of a computer program

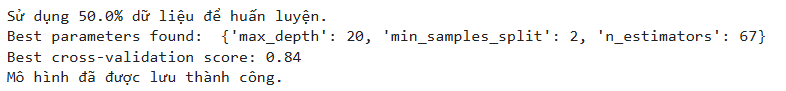
Description automatically generated**



Đoạn mã thực hiện điều chỉnh mô hình Random Forest bằng cách sử dụng Randomized Search để tìm kiếm các tham số tối ưu cho mô hình. Cụ thể:

* Định nghĩa các tham số cần điều chỉnh như n\_estimators, max\_depth, và min\_samples\_split.
* Giảm dữ liệu nếu được yêu cầu, giữ lại một phần nhỏ hơn để huấn luyện.
* Sử dụng RandomizedSearchCV với các tham số xác định và số lần lặp n\_iter, sau đó thực hiện tìm kiếm trên tập huấn luyện (X\_train và y\_train).
* In ra các tham số tốt nhất và điểm độ chính xác cao nhất tìm được qua cross-validation.
* Lưu mô hình sau khi điều chỉnh vào file 'random\_search\_rf.pkl'.

Việc này giúp tối ưu mô hình Random Forest trước khi triển khai.

****

**Hình 8.1 Kết quả của Randomized Search**

## **8.2 Phương pháp Tổ hợp (Ensemble Methods)**

Mã này thực hiện các bước sau:

* Khởi tạo các mô hình: Tạo ba mô hình phân loại khác nhau: KNeighborsClassifier, RandomForestClassifier và DecisionTreeClassifier.
* Tạo Voting Classifier: Kết hợp các mô hình này thành một Voting Classifier với chiến lược "hard voting", nghĩa là mô hình sẽ dự đoán dựa trên kết quả của các mô hình thành viên (mô hình nào dự đoán nhiều nhất sẽ được chọn).
* Huấn luyện mô hình: Huấn luyện Voting Classifier trên tập huấn luyện (X\_train, y\_train).
* Dự đoán và đánh giá: Dự đoán kết quả trên tập kiểm tra (X\_test) và tính toán độ chính xác của Voting Classifier, sau đó in ra độ chính xác đó.

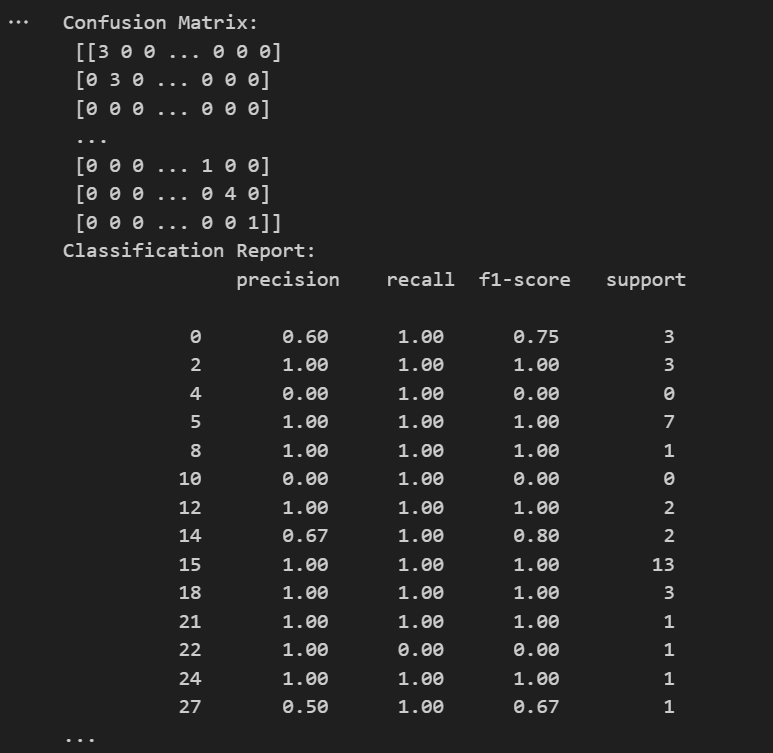
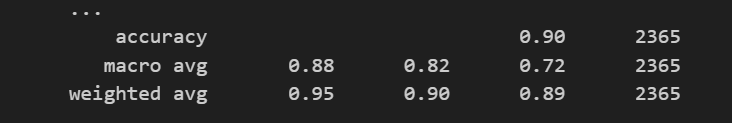


**Hình 8.2 Kết quả của đoạn mã**

## **8.3** **Phân tích các mô hình tốt nhất và lỗi**

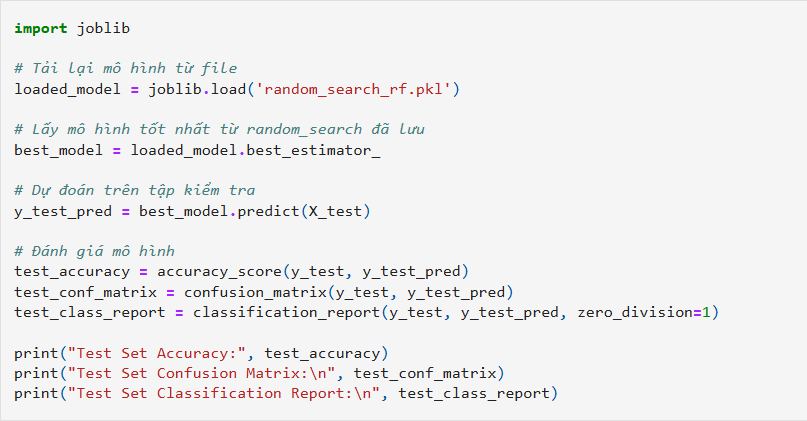
## 

Mã này tính toán và hiển thị ma trận nhầm lẫn và báo cáo phân loại cho mô hình tốt nhất (Voting Classifier) trên tập kiểm tra, giúp đánh giá hiệu suất của mô hình dự đoán.

**** ****

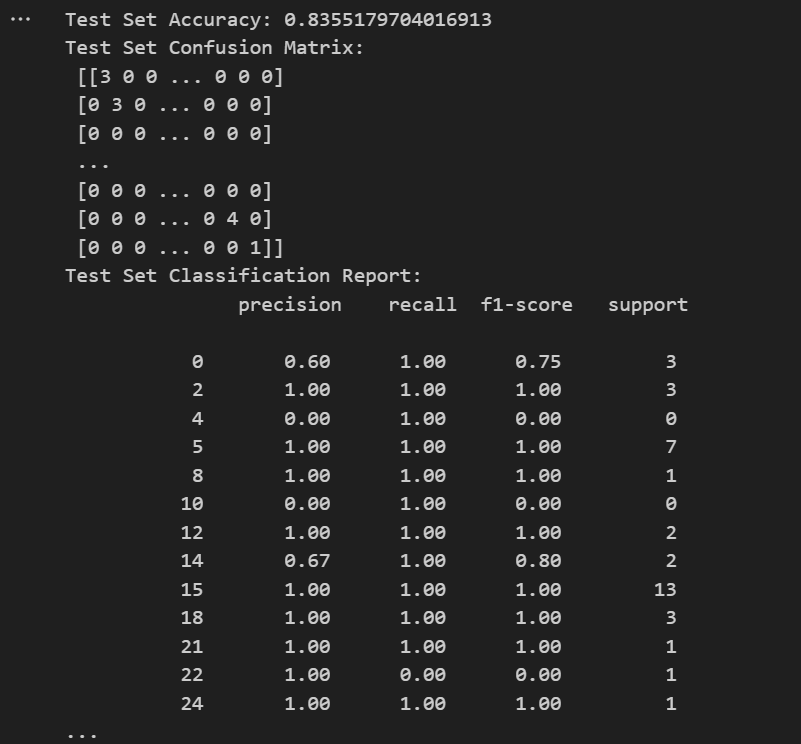
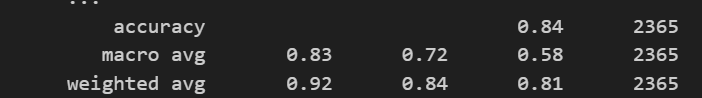
**Hình 8.3 Kết quả của việc phân tích**

## **8.4 Đánh giá hệ thống của bạn trên tập kiểm tra**

****

Đoạn mã này thực hiện các bước sau:

* Tải mô hình: Tải lại mô hình Random Forest tốt nhất đã lưu từ file random\_search\_rf.pkl.
* Dự đoán: Sử dụng mô hình tốt nhất để dự đoán trên tập kiểm tra.
* Đánh giá mô hình: Tính toán và in ra độ chính xác, ma trận nhầm lẫn, và báo cáo phân loại cho các dự đoán trên tập kiểm tra.

**** ****

**Hình 8.3 Kết quả của đánh giá hệ thống**