

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**BÁO CÁO
KHAI KHOÁNG DỮ LIỆU**

**Đề tài
XÂY DỰNG MÔ HÌNH PHÂN LOẠI TỰ ĐỘNG CÁC
ĐÁNH GIÁ VỀ THUỐC DỰA TRÊN TẬP DỮ LIỆU
DRUG REVIEW DATASET**

Sinh viên thực hiện:
Nguyễn Khắc Duy - B2017027
Trần Minh Được - B2017033
Đình Hoàng Nhân - B2007253

Giảng viên hướng dẫn:
TS. Lưu Tiến Đạo

Cần Thơ, 10/2023

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn đến Thầy Lưu Tiến Đạo. Trong quá trình học và thực hiện đồ án môn học, chúng em đã nhận được sự hỗ trợ tận tình từ Thầy. Qua đó giúp chúng em có thể được tiếp thu và củng cố những kiến thức quan trọng về môn học Khai Khoáng Dữ Liệu. Từ những kiến thức có được đã giúp chúng em hoàn thành được đồ án này và rèn luyện thêm kinh nghiệm làm việc và học tập.

Trong quá trình thực hiện đồ án, dù chúng em đã cố gắng nhưng vẫn còn tồn tại nhiều thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp đến từ Thầy để bài báo cáo của chúng em được chỉnh chu và chính xác hơn.

Chúng em chúc Thầy có nhiều sức khỏe và thành công trong công việc của mình.

Cần Thơ, ngày 23 tháng 10 năm 2023

Người viết

Nguyễn Khắc Duy,

Đinh Hoàng Nhân,

Trần Minh Được.

MỤC LỤC

LỜI CẢM ƠN	1
MỤC LỤC	2
DANH MỤC HÌNH	4
TÓM TẮT	6
ABSTRACT	6
PHẦN GIỚI THIỆU	8
1. Đặt vấn đề	8
2. Mục tiêu đề tài	8
3. Phạm vi nghiên cứu	8
4. Phương pháp nghiên cứu	8
4.1. Về lý thuyết	8
4.2. Về thực hành	9
5. Nội dung nghiên cứu	9
6. Kết quả đạt được	10
7. Bố cục bài báo cáo	10
PHẦN NỘI DUNG	11
CHƯƠNG 1: MÔ TẢ BÀI TOÁN	11
1. Mô tả chi tiết bài toán	11
2. Vấn đề và giải pháp liên quan đến bài toán	11
CHƯƠNG 2: TRỰC QUAN HÓA, XỬ LÝ DỮ LIỆU	13
1. Trực quan hóa tập dữ liệu	13
1.1. Tổng quan về tập dữ liệu	13
1.2. Trực quan hóa tập dữ liệu	14
1.2.1. Tên thuốc (drugName)	14
1.2.2. Condition (Tình trạng)	15

1.2.3. Đánh giá (review)	16
1.2.4. Xếp hạng đánh giá (Rating)	17
1.2.5. Thời gian (date)	19
1.2.6. Mức độ hữu ích của đánh giá (usefullCount)	20
1.3. Mối tương quan giữa các biến	20
1.3.1. Review và rating	20
1.3.2. Loại thuốc(drugName) và xếp hạng (rating)	22
1.3.3. Xếp hạng (rating) và Độ hữu ích (usefulCount)	24
1.3.4. DrugName và condition	25
2. Xử lý dữ liệu	26
CHƯƠNG 3: XÂY DỰNG MÔ HÌNH PHÂN LỚP	31
1. Thuật toán phân lớp	31
2. Tiêu chí đánh giá mô hình phân lớp	31
3. Xây dựng mô hình phân lớp	32
3.1. Cây quyết định (DecisionTreeClassifier)	33
3.2. Rừng ngẫu nhiên (RandomForestClassifier)	35
3.3. LightGBM()	37
4. Đánh giá.	39
5. Xây dựng trang web phân loại đánh giá	39

DANH MỤC HÌNH

Hình 1. 1: Tổng quan tập dữ liệu	14
Hình 1. 2: Số lượng thuốc có trong tập dữ liệu	14
Hình 1. 3: 20 loại thuốc phổ biến	15
Hình 1. 4: Dữ liệu null của thuộc tính "condition"	15
Hình 1. 5: 20 tình trạng phổ biến trong tập dữ liệu	16
Hình 1. 6: Số lượng đánh giá qua các năm	17
Hình 1. 7: Trục quan thuộc tính rating	17
Hình 1. 8: Sự phân bố các đánh giá từ 1 - 10 sao	18
Hình 1. 9: Xu hướng đánh giá qua các năm	19
Hình 1. 10: Phân bố giá trị về độ hữu ích của các đánh giá	20
Hình 1. 11: Cài đặt thư viện plotly	21
Hình 1. 12: Phân loại các từ thuộc 2 nhóm đánh giá	21
Hình 1. 13: Các từ xuất hiện trong các đánh giá 1-5 sao	21
Hình 1. 14: Các từ xuất hiện trong các đánh giá từ 6 - 10 sao	22
Hình 1. 15: Cài đặt thư viện để biểu diễn sự tương quan dữ liệu	22
Hình 1. 16: Top 20 loại thuốc với đánh giá 10/10 sao	23
Hình 1. 17: Lấy những tên thuốc được đánh giá 1 sao	23
Hình 1. 18: Top 20 loại thuốc được đánh giá 1/10 sao	24
Hình 1. 19: Sử dụng thư viện seaborn	24
Hình 1. 20: Tương quan giữa rating và usefulCount	25
Hình 1. 21: Lọc 20 condition phổ biến và biểu diễn tương quan giữa drugName và condition	25
Hình 1. 22: Tương quan giữa 2 biến drugName và condition	26
Hình 2. 1: Minh họa dữ liệu dư thừa	26
Hình 2. 2: Xóa ký tự '	27
Hình 2. 3: Ký tự đặc biệt	27
Hình 2. 4: Xóa ký tự đặc biệt	27
Hình 2. 5: Xóa ký tự ASCII	27
Hình 2. 6: Xóa ký tự khoảng trống	27
Hình 2. 7: Xóa những dấu chấm liên tục	28
Hình 2. 8: Loại bỏ những stop_words	28

Hình 2. 9: Ví dụ xóa stop_words	28
Hình 2. 10: Xóa những stem_words	28
Hình 2. 11: Minh họa xóa stem_words	29
Hình 2. 12: Dữ liệu chưa chuẩn hóa	29
Hình 2. 13: Dữ liệu sau chuẩn hóa	29
Hình 2. 14: Cài đặt TfidfVectorizer	30
Hình 2. 15: Vector hóa cột review và drugName	30
Hình 2. 16: Dữ liệu sau khi vector hóa	30
Hình 2. 17: Ma trận nhầm lẫn	32
Hình 2. 18: Cài đặt chỉ số đánh giá và ma trận nhầm lẫn	33
Hình 2. 19: Cây quyết định	33
Hình 2. 20: Cài đặt mô hình Cây quyết định	33
Hình 2. 21: Dự đoán mô hình	34
Hình 2. 22: Kết quả dự đoán	34
Hình 2. 23: Ma trận nhầm lẫn cho mô hình Cây quyết định	34
Hình 2. 24: Cài đặt mô hình Rừng ngẫu nhiên	35
Hình 2. 25: Kết quả dự đoán của mô hình Rừng ngẫu nhiên	35
Hình 2. 26: Ma trận nhầm lẫn của mô hình Rừng ngẫu nhiên	35
Hình 2. 27: Cài đặt mô hình LightGBM	36
Hình 2. 28: Huấn luyện mô hình LightGBM	36
Hình 2. 29: Kết quả dự đoán của mô hình LightGBM	36
Hình 2. 30: Ma trận nhầm lẫn của mô hình LightGBM	37
Hình 5. 1: Giao diện dự đoán	40
Hình 5. 2: Kết quả dự đoán	40

TÓM TẮT

Bộ dữ liệu tổng hợp những đánh giá về thuốc (Drug review dataset) đóng vai trò quan trọng trong nghiên cứu y tế và ngành công nghiệp dược phẩm. Dữ liệu cung cấp là những thông tin cần thiết về đánh giá của người dùng về các loại thuốc và tác dụng phụ của chúng. Qua đó giúp các nhà nghiên cứu y tế và doanh nghiệp dược phẩm có thể phát triển và cải tiến các sản phẩm y tế ngày một tốt hơn.

Quá trình nghiên cứu tập dữ liệu bao gồm thu thập và phân tích dữ liệu qua những thông tin như tên thuốc, đánh giá, bình luận về thuốc và những thông tin phụ khác được tổng hợp qua 215,063 đánh giá mà người dùng cung cấp.

Mục tiêu của đề tài là xây dựng mô hình phân loại dữ liệu tự động các đánh giá về thuốc, nhằm tư vấn và hỗ trợ quá trình đánh giá các loại thuốc.

ABSTRACT

The synthesized dataset of drug reviews plays a pivotal role in both medical research and the pharmaceutical industry. It provides essential information regarding user evaluations of various medications and their associated side effects. Consequently, this dataset empowers healthcare researchers and pharmaceutical enterprises to enhance and advance healthcare products.

The research process involves the collection and analysis of data encompassing details such as drug names, user ratings, comments on the drugs, and supplementary information. This amalgamation stems from a comprehensive pool of 215,063 user reviews.

The objective of this study is to construct an automated classification model for drug review data, aiming to advise and facilitate the evaluation process of different medications.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

- Một trong những vấn đề được quan tâm nhất trong thời đại hiện nay đó là sức khỏe con người. Việc sử dụng thuốc và những sản phẩm chức năng để điều trị hoặc hỗ trợ sức khỏe là nhu cầu tất yếu của mọi người. Đi kèm với đó là việc tìm hiểu và lựa chọn những sản phẩm phù hợp với tình trạng cá nhân mỗi người. Đồng thời những doanh nghiệp trong ngành công nghiệp y tế cũng cần biết những phản hồi của người dùng về các loại thuốc để từ đó nghiên cứu và cải thiện tác dụng của thuốc.
- Do đó, cần thiết phải có một ứng dụng có thể giúp người tiêu dùng và những doanh nghiệp y tế có thể có cái nhìn tổng quan về những loại thuốc.

2. Mục tiêu đề tài

- Từ tập dữ liệu được cung cấp, tiến hành phân tích trực quan, mô tả và tìm hiểu sự liên quan giữa các thuộc tính. Từ đó xây dựng mô hình phân loại tự động các đánh giá về thuốc.

3. Phạm vi nghiên cứu

- Lập trình bằng ngôn ngữ Python.
- Công cụ sử dụng: Google Colab, PyCharm

4. Phương pháp nghiên cứu

4.1. Về lý thuyết

- Tìm hiểu về ngôn ngữ Python.
- Hiểu được cách hoạt động của các thư viện, các hàm hỗ trợ.
- Hiểu được bản chất của mô hình phân lớp và áp dụng
- Làm việc nhóm, trao đổi thông tin để giải quyết vấn đề.

4.2. Về thực hành

- Tìm hiểu về tập dữ liệu qua việc phân tích mô tả trực quan tập dữ liệu.
- Xây dựng mô hình phân lớp để phân lớp tự động những đánh giá về thuốc.

5. Nội dung nghiên cứu

- Nội dung:

- Tìm hiểu tập dữ liệu
- Phân tích trực quan
- Phân tích sự tương quan giữa các thuộc tính
- Xây dựng mô hình phân lớp
- Đánh giá hiệu quả của mô hình

- Phân công công việc

- Nguyễn Khắc Duy:
 - ◆ Tìm hiểu tập dữ liệu.
 - ◆ Mô tả tập dữ liệu.
 - ◆ Xử lý dữ liệu.
 - ◆ Viết báo cáo.
 - ◆ Xây dựng mô hình Rừng ngẫu nhiên.
- Đinh Hoàng Nhân:
 - ◆ Tìm hiểu tập dữ liệu.
 - ◆ Xây dựng mô hình Cây quyết định.
 - ◆ Đề xuất phương án tiền xử lý dữ liệu
- Trần Minh Được:
 - ◆ Tìm hiểu tập dữ liệu.
 - ◆ Mô tả tập dữ liệu.
 - ◆ Xây dựng mô hình.
 - ◆ Viết slide.
 - ◆ Xây dựng mô hình LightGBM.

- ◆ Xây dựng website mô phỏng dự đoán đánh giá.
- Tất cả các thành viên đều nỗ lực và hỗ trợ nhau trong quá trình hoàn thiện bài báo cáo này nhằm đạt kết quả tốt nhất.

6. Kết quả đạt được

7. Bố cục bài báo cáo

- Phần giới thiệu:

- Giới thiệu tổng quát về đề tài, mục tiêu của đề tài, phạm vi nghiên cứu, phương pháp nghiên cứu và kết quả đạt được.

- Phần nội dung:

- Chương 1: Mô tả bài toán.
- Chương 2: Trực quan hóa, xử lý dữ liệu.
- Chương 3: Xây dựng mô hình phân lớp.

- Phần kết quả:

- Đưa ra kết quả và đánh giá mô hình.

PHẦN NỘI DUNG

CHƯƠNG 1: MÔ TẢ BÀI TOÁN

1. Mô tả chi tiết bài toán

- Để phân loại một đánh giá về một loại thuốc là tích cực, tiêu cực hay trung lập, chúng ta cần có dữ liệu về loại thuốc đó bao gồm tên thuốc, bình luận. Những thuộc tính đó đã được tổng hợp trong bộ dữ liệu Drug Review Dataset được lấy từ UCI Machine Learning Repository do hai tác giả Surya Kallumadi và Felix Grer tổng hợp.

- Đầu tiên cần tìm hiểu tổng quan về tập dữ liệu, phân tích mối quan hệ giữa các thuộc tính để có cái nhìn trực quan. Từ đó tìm ra được đặc trưng của tập dữ liệu để đưa ra mô hình hợp lý. Tiếp theo sẽ tiến hành huấn luyện tập dữ liệu trên mô hình đã chọn để tìm ra mô hình với độ chính xác tối ưu nhất để sử dụng cho việc phân loại tự động các đánh giá về thuốc. Nhận thấy rằng mô hình phân lớp là phù hợp cho tập dữ liệu trên và phù hợp với nhu cầu của bài toán.

2. Vấn đề và giải pháp liên quan đến bài toán

- Vấn đề được đặt ra là dựa vào các yếu tố nào để nhận biết loại thuốc đó có đánh giá là tốt hay xấu, dựa vào những yếu tố nào để xác định, các yếu tố được chọn để đánh giá thuốc có tính tương quan gì với nhau hay không. Nếu có thì các giá trị đó tương quan như thế nào, nó ảnh hưởng như thế nào đến kết quả dự đoán của mô hình. Và dùng phương pháp gì để có thể phân loại được các đánh giá đó và thời gian là bao lâu để hoàn thành . Đối với dữ liệu lớn thì giải quyết như nào.

- Giải pháp để giải các vấn đề trên là xây dựng một mô hình máy học để có thể đưa ra sự phân loại và đánh giá loại thuốc là tốt hay xấu. Với mô hình chỉ cần hai tham số đó là drugName(tên thuốc) và reviews (đánh giá) là

có thể xây dựng mô hình máy học dự đoán. Tại sao chỉ chọn hai tham số để đánh giá? Với mỗi loại thuốc cơ bản ta có thể dễ dàng nhận biết được thuốc là tốt hay xấu qua những từ ngữ. Sẽ có những từ ngữ biểu hiện sự hài lòng và không hài lòng. Bằng cách dựa vào những từ ngữ đó ta có thể nhận biết là thuốc đó là tốt hay là xấu.

CHƯƠNG 2: TRỰC QUAN HÓA, XỬ LÝ DỮ LIỆU

1. Trực quan hóa tập dữ liệu

1.1. Tổng quan về tập dữ liệu

- Dữ liệu được thu thập từ :

<https://archive.ics.uci.edu/dataset/462/drug+review+dataset+drugs+com>

- Tập dữ liệu chứa tổng cộng 215,063 mẫu đánh giá từ người dùng với các cột thuộc tính: “drugName”, “condition”, “review”, “rating”, “date” và “usefulCount”. Với nhãn là cột “rating”, cột sẽ quy định một đánh giá là tích cực hay tiêu cực dựa trên số sao.

- Cột “drugName”: tên của loại thuốc được đánh giá. Thuộc tính sẽ được sử dụng để nhóm các đánh giá theo thuốc và phân tích hiệu quả của từng loại thuốc theo những tình trạng cụ thể.
- Cột “condition”: thuộc tính này mô tả tình trạng bệnh lý mà loại thuốc được sử dụng để điều trị.
- Cột “review”: đây là phần chính của tập dữ liệu, nơi người dùng viết các bài đánh giá về loại thuốc. Đánh giá có thể chứa thông tin về kinh nghiệm sử dụng thuốc, tác dụng phụ, hiệu quả, hoặc bất kỳ thông tin liên quan nào mà người dùng muốn chia sẻ.
- Cột “rating”: điểm đánh giá là giá trị số nằm trong khoảng từ 1-10 mà người dùng gán cho loại thuốc nào đó. Nó thể hiện mức độ hài lòng hoặc không hài lòng của người dùng về thuốc. Điểm cao thường biểu hiện sự hài lòng cao và ngược lại.
- Cột “date”: đây là ngày mà người dùng đăng đánh giá về loại thuốc. Thông qua thuộc tính này, chúng ta có thể xem xét xu hướng đánh giá theo thời gian, ví dụ, xem xét liệu một loại thuốc đã được sử dụng trong một khoảng thời gian dài hay ngắn.

- Cột “usefullCount”: thuộc tính này theo dõi số lần mà người dùng khác đã đánh dấu đánh giá này là hữu ích. Nó có thể cho biết mức độ sự quan tâm của người đọc đối với bài đánh giá cụ thể hoặc độ tin cậy của nó.

1.2. Trục quan hóa tập dữ liệu

- Tập dữ liệu có 215,063 mẫu và 7 cột:

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount
0	206461	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combinati...	9.0	May 20, 2012	27
1	95260	Guanfacine	ADHD	"My son is halfway through his fourth week of ...	8.0	April 27, 2010	192
2	92703	Lybrel	Birth Control	"I used to take another oral contraceptive, wh...	5.0	December 14, 2009	17
3	138000	Ortho Evra	Birth Control	"This is my first time using any form of birth...	8.0	November 3, 2015	10
4	35696	Buprenorphine / naloxone	Opiate Dependence	"Suboxone has completely turned my life around...	9.0	November 27, 2016	37
...
53761	159999	Tamoxifen	Breast Cancer, Prevention	"I have taken Tamoxifen for 5 years. Side effe...	10.0	September 13, 2014	43
53762	140714	Escitalopram	Anxiety	"I've been taking Lexapro (escitaploprgra...	9.0	October 8, 2016	11
53763	130945	Levonorgestrel	Birth Control	"I'm married, 34 years old and I have no ...	8.0	November 15, 2010	7
53764	47656	Tapentadol	Pain	"I was prescribed Nucynta for severe neck/shou...	1.0	November 28, 2011	20
53765	113712	Arthrotec	Sciatica	"It works!!!"	9.0	September 13, 2009	46

215063 rows x 7 columns

Hình 1. 1: Tổng quan tập dữ liệu

1.2.1. Tên thuốc (drugName)

```

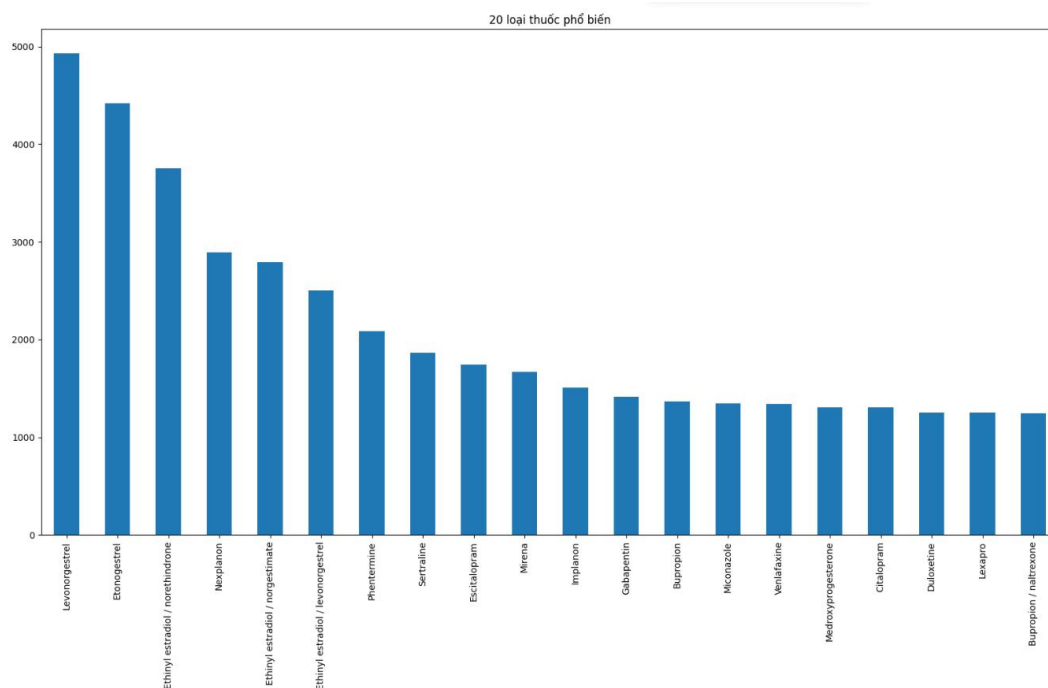
▶ num = len(data['drugName'].unique().tolist())
print('We number of Drugs are -',num )

⇒ We number of Drugs are - 3671

```

Hình 1. 2: Số lượng thuốc có trong tập dữ liệu

- Trong tập dữ liệu ở cột drugName có tổng cộng 3671 loại thuốc khác nhau.



Hình 1. 3: 20 loại thuốc phổ biến

- Trong 3671 loại thuốc khác nhau, có khoảng 20 loại thuốc chiếm đa số trên 1000 lần được đánh giá. Thuốc Levonorgestrel chiếm tỉ lệ cao nhất với khoảng gần 5000 lượt đề cập, theo sau đó là Etonogestrel, Estradiol và Norethindrome.

1.2.2. Condition (Tình trạng)

- Thuộc tính “condition” có 1194 dòng dữ liệu với giá trị NaN:

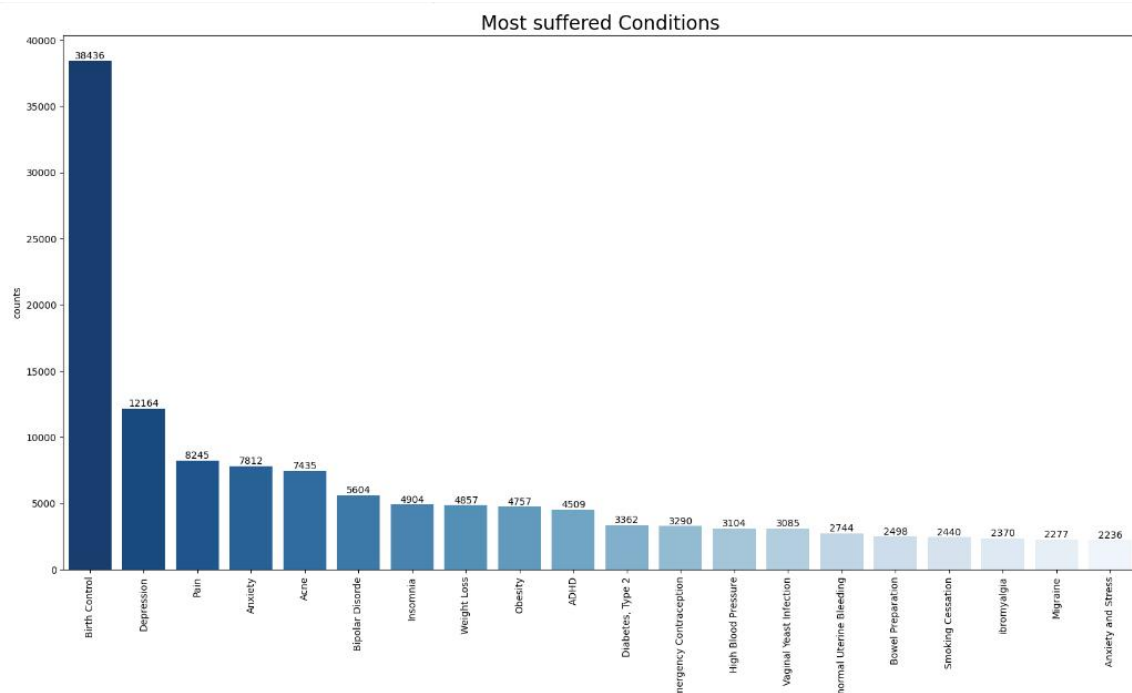
```
data.isnull().sum()
```

Thuộc tính	Số lượng giá trị null
Unnamed: 0	0
drugName	0
condition	1194
review	0
rating	0
date	0
usefulCount	0

dtype: int64

Hình 1. 4: Dữ liệu null của thuộc tính "condition"

- Biểu đồ dưới đây thể hiện top 20 tình trạng mà người bệnh thường gặp phải, trong số đó, người bệnh sử dụng thuốc ngừa thai chiếm tỉ lệ cao và vượt trội nhất với 30,436 đánh giá liên quan.

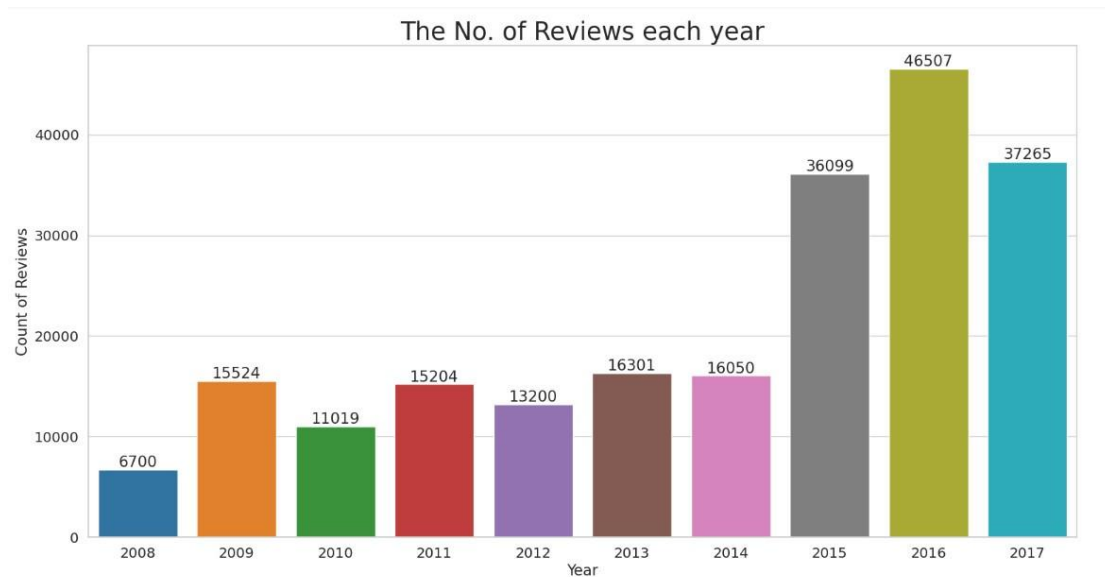


Hình 1. 5: 20 tình trạng phổ biến trong tập dữ liệu

- Việc có tỉ lệ vượt trội hơn so với các loại thuốc trong tập dữ liệu, thuốc ngừa thai (Birth Control) có ý nghĩa trong chiến lược phân phối thuốc của những doanh nghiệp y tế.

1.2.3. Đánh giá (review)

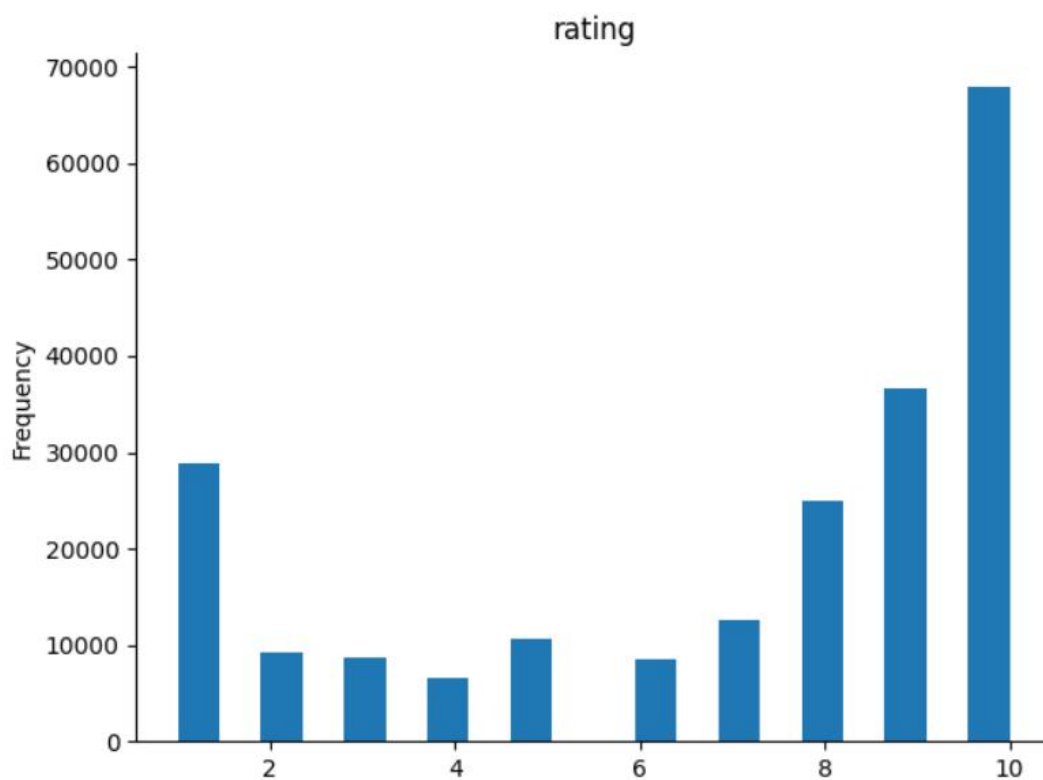
- Vì là dữ liệu dạng chữ nên thuộc tính đánh giá sẽ mang những nội dung không mong muốn, những ký tự đặc biệt như: ''', ký tự đặc biệt, ký tự ASCII, ký tự khoảng trắng ở đầu và cuối, những dấu chấm liên tục,... Do đóng vai trò quan trọng trong việc đánh giá dữ liệu nên sẽ yêu cầu chuẩn hóa dữ liệu cho thuộc tính này.



Hình 1. 6: Số lượng đánh giá qua các năm

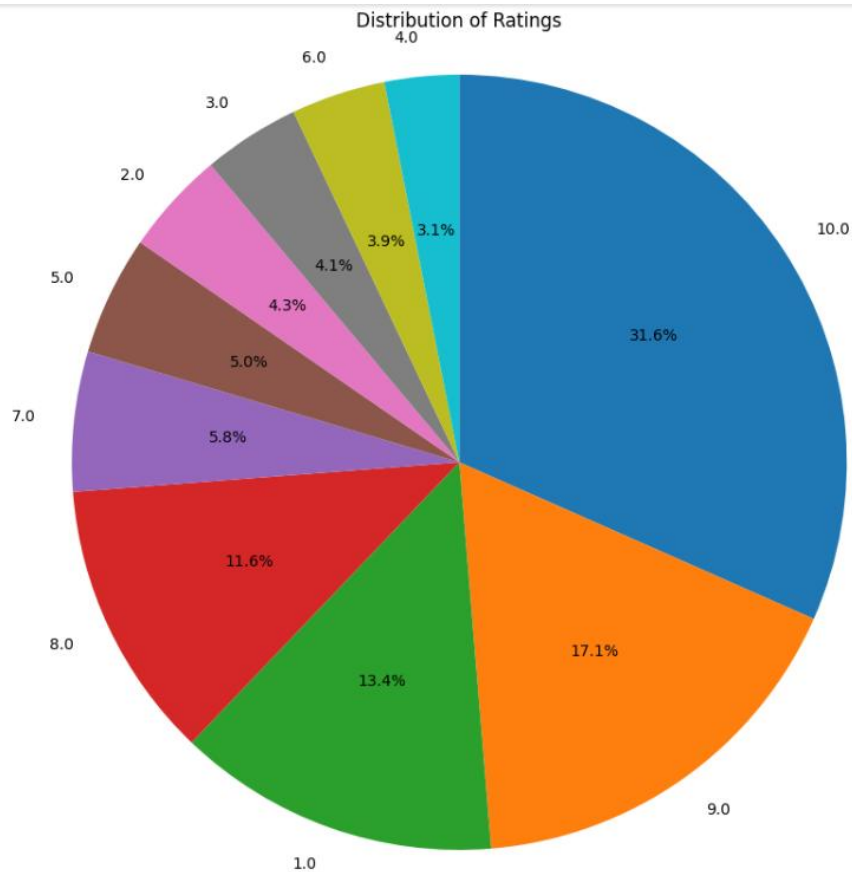
- Biểu đồ trên thể hiện số lượng đánh giá tăng lên theo từng năm, cho thấy mức độ sử dụng thuốc của người dùng tăng đáng kể, cao nhất là 46507 đánh giá năm 2016.

1.2.4. Xếp hạng đánh giá (Rating)



Hình 1. 7: Trục quan thuộc tính rating

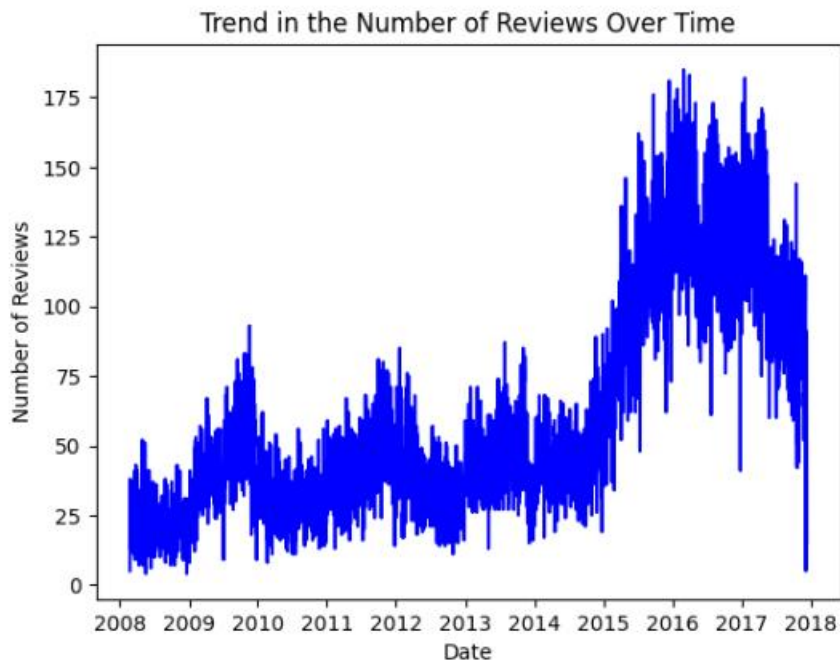
- Xếp hạng của một đánh giá nằm trong khoảng từ 1 - 10 sao thể hiện mức độ hài lòng của người dùng đối với một loại thuốc cụ thể. Trong biểu đồ trên, đánh giá 10 sao chiếm đa số, còn đối với khoảng từ 2 - 7 sao có số lượng gần tương đồng nhau. Phải kể đến những đánh giá 1 sao cũng có số lượng đáng kể. Đây là những thông tin quan trọng trong bài toán phân loại đánh giá người dùng.



Hình 1. 8: Sự phân bố các đánh giá từ 1 - 10 sao

- Xếp hạng 9-10 sao chiếm gần 50% những đánh giá cho biết được mức độ hài lòng về sản phẩm y tế của người dùng vẫn khá cao.

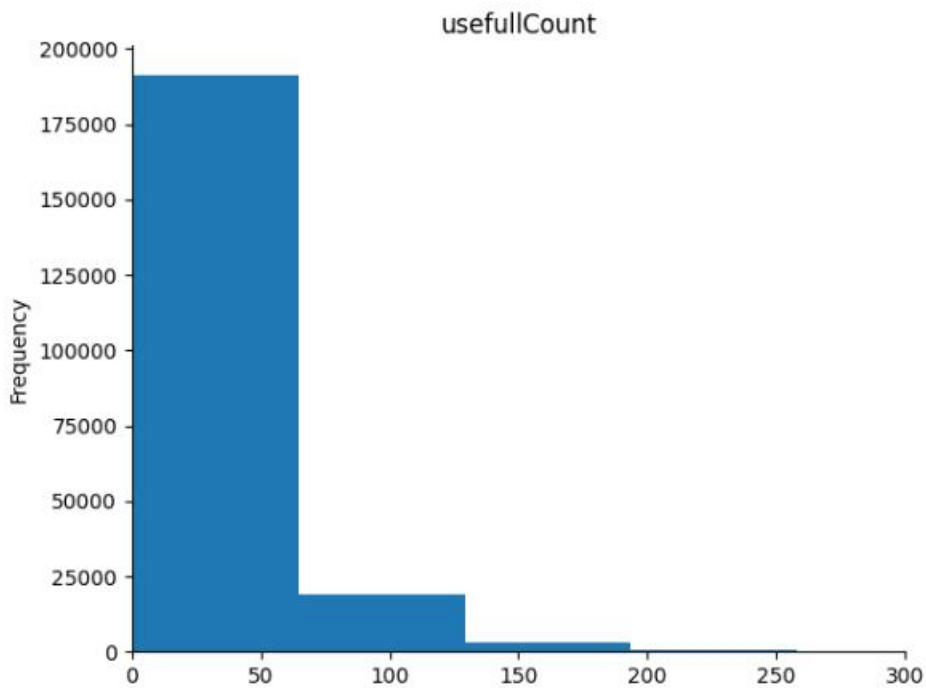
1.2.5. Thời gian (date)



Hình 1. 9: Xu hướng đánh giá qua các năm

- Việc phân bổ thời gian đánh giá như biểu đồ cho thấy đã có những cải thiện về khả năng tiếp cận đến thuốc do tính sẵn có hoặc chi phí dẫn đến tăng số lượng bệnh nhân sử dụng thuốc và viết đánh giá.
- Việc tăng cường sử dụng những nền tảng trực tuyến để đánh giá thuốc và tình trạng bệnh cũng đã góp phần làm tăng số lượng đánh giá từ năm 2015 trở đi.
- Những thay đổi trong chiến lược tiếp thị: Những nhà sản xuất và các ngành chăm sóc sức khỏe có thể đã thay đổi chiến lược tiếp thị của họ để tăng đề xuất của thuốc dẫn đến bệnh nhân sử dụng thuốc và viết đánh giá tăng lên.

1.2.6. Mức độ hữu ích của đánh giá (usefullCount)



Hình 1. 10: Phân bố giá trị về độ hữu ích của các đánh giá

- Khoảng giá trị từ 0 đến dưới 60 người chiếm phần đa số trong việc bình chọn một đánh giá là hữu ích.

1.3. Mối tương quan giữa các biến

1.3.1. Review và rating

- Sử dụng thư viện plotly để tạo biểu đồ Ngram - Bigram biểu diễn tần suất xuất hiện của những cụm từ có 2 từ phổ biến trong cột “review”, được phân chia riêng biệt theo số đánh giá từ 1 - 5 sao và 6 - 10 sao xếp hạng.

```

from plotly import tools
import plotly.offline as py
import plotly.graph_objs as go

from collections import defaultdict
rating_6_10 = data[data["rating"]>5]
rating_1_5 = data[data["rating"]<6]

```

Hình 1. 11: Cài đặt thư viện plotly

```

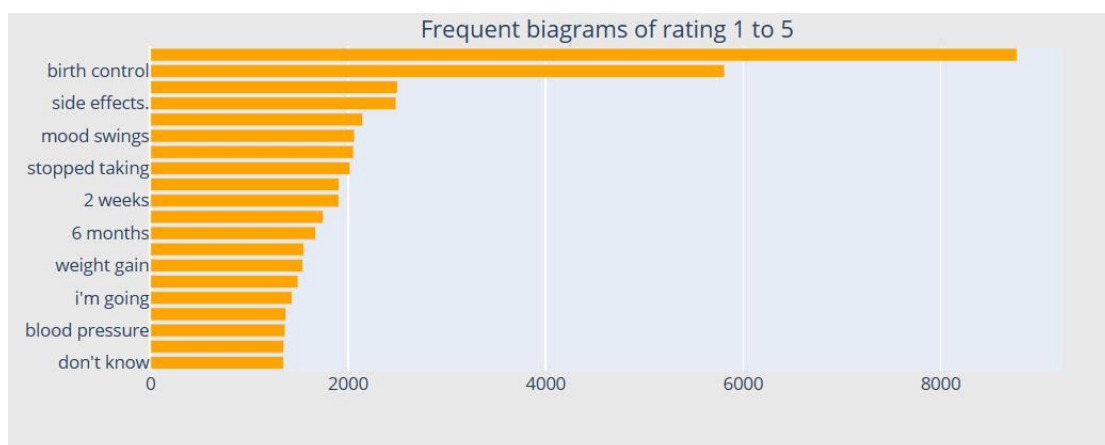
freq_dict = defaultdict(int)
# Tạo một từ điển lưu số lần xuất hiện của một
# từ trong đoạn review mà có đánh giá 1-5 sao
for sent in rating_1_5["review"]:
    for word in generate_ngrams(sent,2):
        freq_dict[word] += 1

for sent in rating_6_10["review"]:
    for word in generate_ngrams(sent,2):
        freq_dict[word] += 1

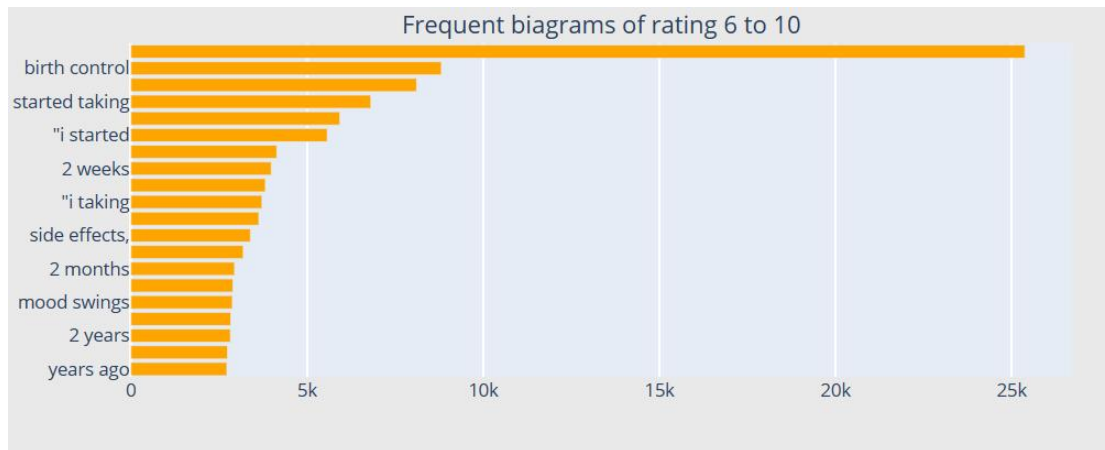
```

Hình 1. 12: Phân loại các từ thuộc 2 nhóm đánh giá

- Có thể thay đổi số lượng từ mong muốn với tham số **n** ở hàm `generate_ngrams(sent, n)`, ở đây chúng ta lấy cụm từ có 2 từ.



Hình 1. 13: Các từ xuất hiện trong các đánh giá 1-5 sao



Hình 1. 14: Các từ xuất hiện trong các đánh giá từ 6 - 10 sao

- Có thể thấy có những từ xuất hiện ở cả 2 loại xếp hạng ví dụ như: *birth control*, *side effects*, *2 weeks*, ... hoặc có những từ chỉ xuất hiện ở một loại xếp hạng như: *stopped taking* chỉ xuất hiện ở những xếp hạng từ 1 - 5 sao. Điều này có vai trò quan trọng trong việc dự đoán và phân loại một đánh giá là tích cực hay tiêu cực.

1.3.2. Loại thuốc(drugName) và xếp hạng (rating)

```

import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(20,10))
rating = dict(data.loc[data.rating == 10, "drugName"].value_counts())
drugname = list(rating.keys())
drug_rating = list(rating.values())

sns_rating = sns.barplot(x = drugname[0:20], y = drug_rating[0:20], palette = 'BuPu_r')

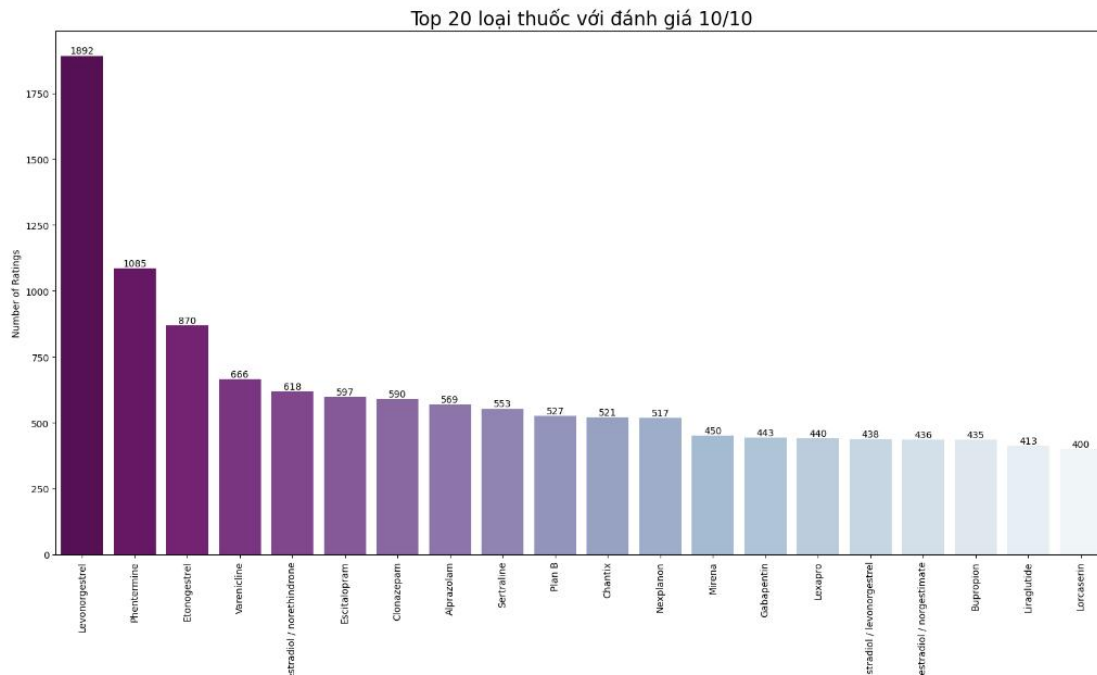
for i in sns_rating.containers:
    sns_rating.bar_label(i,)

plt.setp(sns_rating.get_xticklabels(), rotation=90)
plt.show()

```

Hình 1. 15: Cài đặt thư viện để biểu diễn sự tương quan dữ liệu

- Sử dụng thư viện matplotlib và seaborn để minh họa sự tương quan giữa drugName và rating, ở đây chúng ta chỉ hiển thị 20 loại thuốc được xếp hạng 10 sao nhiều nhất.



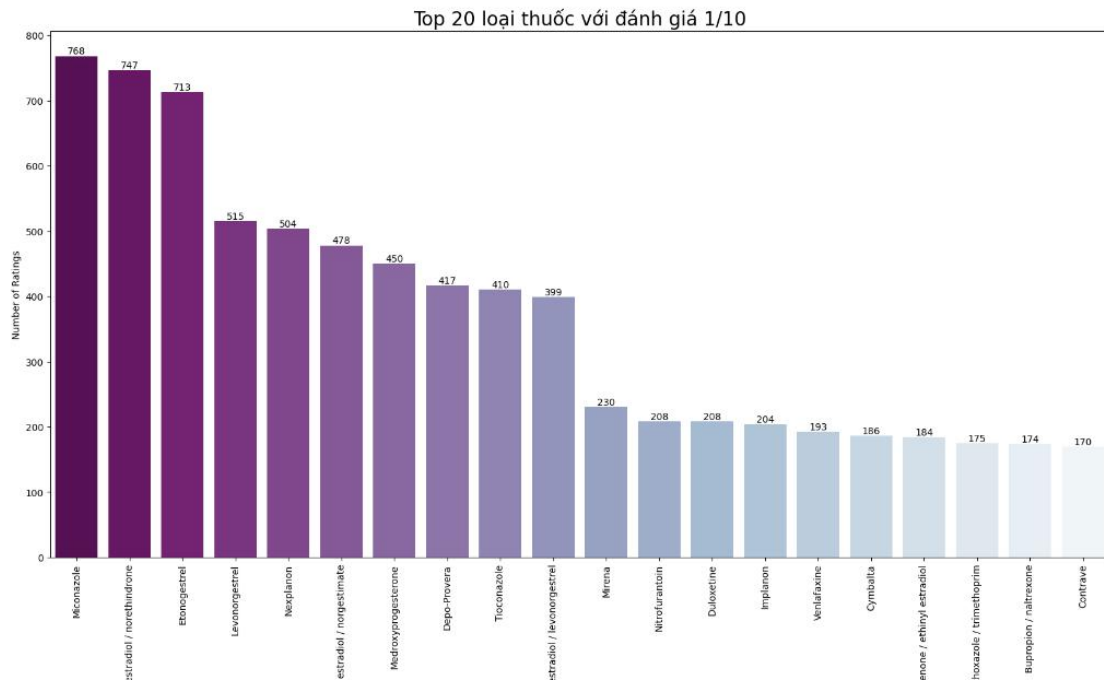
Hình 1. 16: Top 20 loại thuốc với đánh giá 10/10 sao

- Các loại thuốc như **Levonorgestrel** (1892 votes), **Phentemine** (1085 votes), **Etonogestrel** (870 votes) là những loại thuốc có số lượt đánh giá 10 sao cao nhất.

```
rating = dict(data.loc[data.rating == 1, "drugName"].value_counts())
```

Hình 1. 17: Lấy những tên thuốc được đánh giá 1 sao

- Thay đổi giá trị data.rating để được Top 20 đánh giá được xếp hạng 1 sao nhiều nhất.



Hình 1. 18: Top 20 loại thuốc được đánh giá 1/10 sao

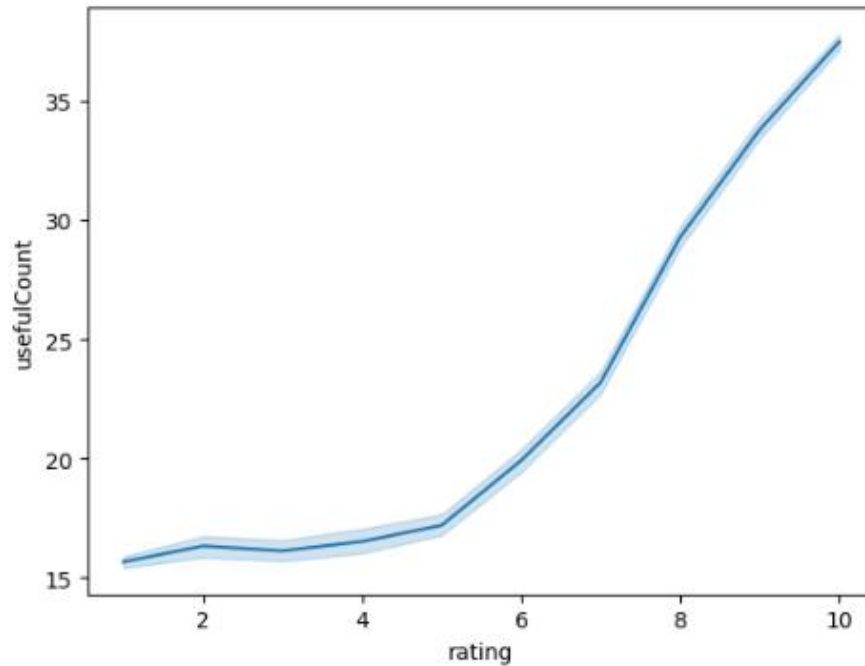
- Bên cạnh đó những loại thuốc như **Miconzole**(768), **Norethindrone**(747), ... là những loại thuốc được người dùng đánh giá 1 sao nhiều nhất.

1.3.3. Xếp hạng (rating) và Độ hữu ích (usefulCount)

- Sử dụng thư viện seaborn để mô tả mối liên hệ giữa 2 biến rating và usefulCount.

```
sns.lineplot(data=data,x='rating',y='usefulCount')
```

Hình 1. 19: Sử dụng thư viện seaborn



Hình 1. 20: Tương quan giữa rating và usefulCount

- Biểu đồ trên thể hiện mức độ người dùng bình chọn một đánh giá là hữu ích có tỉ lệ với xếp hạng của một đánh giá. Đa phần người dùng có xu hướng coi một đánh giá là hữu ích nếu người đăng đánh giá cho rằng loại thuốc đó có tác dụng tích cực và cho nó số sao từ 6 - 10 sao.

1.3.4. DrugName và condition

Sử dụng biểu đồ Bar Chart để biểu diễn số lượng “drugName” cho mỗi “condition”.

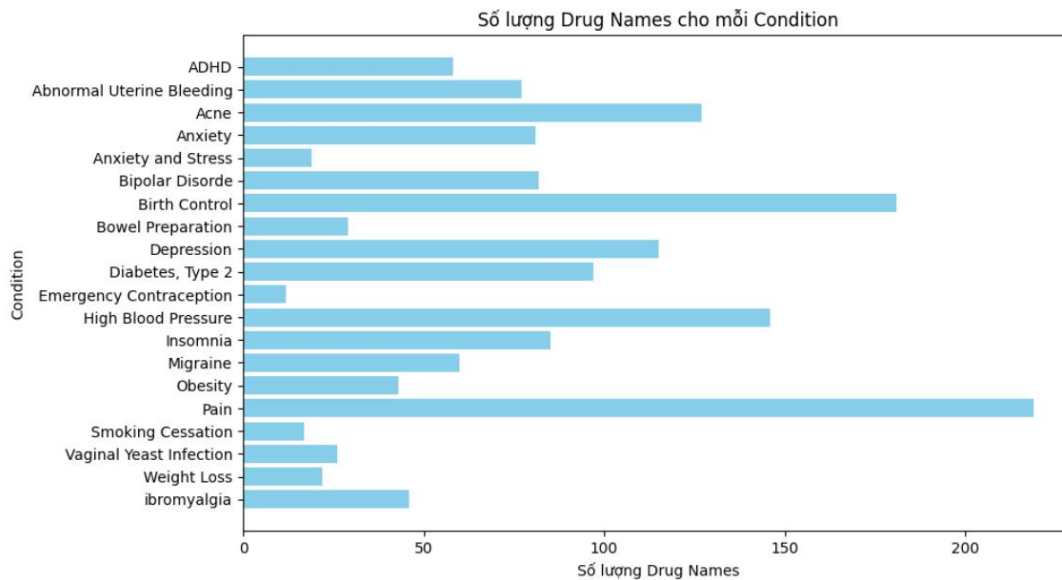
```
# Lọc ra 20 condition phổ biến
top_20_conditions = data['condition'].value_counts().nlargest(20).index
data_filtered = data[data['condition'].isin(top_20_conditions)]

# Đếm số lượng drugName cho mỗi condition
drug_count_by_condition = data_filtered.groupby(['condition'])['drugName'].nunique().reset_index()

# Vẽ biểu đồ
plt.figure(figsize=(10, 6))
plt.barh(drug_count_by_condition['condition'], drug_count_by_condition['drugName'], color='skyblue')
plt.xlabel('Số lượng Drug Names')
plt.ylabel('Condition')
plt.title('Số lượng Drug Names cho mỗi Condition')
plt.gca().invert_yaxis() # Đảo ngược trục y để hiển thị condition phổ biến ở phía trên
plt.show()
```

Hình 1. 21: Lọc 20 condition phổ biến và biểu diễn tương quan giữa drugName và condition

- Do có một lượng lớn giá trị “condition” nên ở phần minh họa này chỉ biểu diễn số lượng thuốc dùng cho 20 tình trạng bệnh phổ biến nhất. Cụ thể như sau:



Hình 1. 22: Tương quan giữa 2 biến drugName và condition

- Có thể thấy 2 tình trạng có thuốc hỗ trợ nhiều nhất là *Pain* và *Birth Control*, với khoảng 400 loại thuốc cho 2 tình trạng này.

2. Xử lý dữ liệu

- Trong bài toán phân loại một đánh giá tự động có 6 thuộc tính, trong đó chúng em chọn ra thuộc tính “drugName”, “review” và “usefulCount” để xây dựng mô hình với nhãn là “rating”.

- Thuộc tính “review” có kiểu dữ liệu là dạng chữ, do đó cần phải có bước chuẩn hóa dữ liệu kỹ càng trước khi sử dụng cho huấn luyện mô hình.

- Xóa chuỗi ký tự “'” xuất hiện nhiều lần:

Unnamed: 0	drugName	condition	review
53762	140714	Escitalopram	Anxiety
			"I've been taking Lexapro (escitalopram) for a while and it's been helping me with my anxiety."
53763	130945	Levonorgestrel	Birth Control
			"I'm married, 34 years old and I have no plans of having more children, so I decided to start taking birth control."

Hình 2. 1: Minh họa dữ liệu dư thừa

Quan sát có thể thấy chuỗi ký tự `'` xuất hiện nhiều lần trong các đánh giá và không có tác dụng trong việc dự đoán nên cần phải lọc những chuỗi ký tự này và đồng thời cũng chuyển tất cả text về dạng chữ thường.

```
def review_clean(review):  
    # changing to lower case  
    lower = review.lower()  
  
    # Thay thế chuỗi &#039 xuất hiện nhiều lần bằng ký tự rỗng;  
    pattern_remove = lower.replace("&#039;", "")
```

Hình 2. 2: Xóa ký tự `'`

- Loại bỏ những ký tự đặc biệt:

53764	47656	Tapentadol	Pain	"I was prescribed Nucynta for severe neck/shou...
53765	113712	Arthrotec	Sciatica	"It works!!!"

Hình 2. 3: Ký tự đặc biệt

Những ký tự không phải chữ cái từ a-z, chữ số từ 0-9 hay dấu khoảng cách sẽ bị loại bỏ.

```
# Xóa tất cả các ký tự đặc biệt  
special_remove = pattern_remove.replace(r'^\w\d\s', ' ')
```

Hình 2. 4: Xóa ký tự đặc biệt

- Loại bỏ những ký tự trong bảng mã ASCII:

```
# Xóa các ký tự trong bảng mã ASCII  
ascii_remove = special_remove.replace(r'^\x00-\x7F+', ' ')
```

Hình 2. 5: Xóa ký tự ASCII

- Xóa những ký tự khoảng trống:

```
# Xóa các ký tự hoặc chuỗi ký tự khoảng trống ở đầu câu  
whitespace_remove = ascii_remove.replace(r'^\s+|\s+?$', '')  
  
# Xóa những ký tự khoảng cách liên tục và thay bằng 1 khoảng cách  
multiw_remove = whitespace_remove.replace(r'\s+', ' ')
```

Hình 2. 6: Xóa ký tự khoảng trống

- Xóa những dấu chấm liên tục:

```
# Thay 2 dấu chấm bằng 1 dấu chấm
dataframe = multiw_remove.replace(r'\.{2,}', ' ')
```

Hình 2. 7: Xóa những dấu chấm liên tục

- Loại bỏ những stop_words xuất hiện trong câu:

```
# Removing the stopwords
import nltk
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
data['review_clean'] = data['review_clean'].apply(
    lambda x: ' '.join(word for word in x.split() if word not in stop_words))
```

Hình 2. 8: Loại bỏ những stop_words

- Bằng cách sử dụng thư viện nltk.corpus để loại bỏ những stopwords không mang nhiều ý nghĩa về ngữ nghĩa như *a, an, the, and, or, to, this, that, ...* .

Ví dụ: Loại bỏ những stopword trong đoạn text sau “*I was depressed and hopeless one second and then mean, irrational, and full of rage the next*”:

```
from nltk.corpus import stopwords
test = 'I was depressed and hopeless one second and then mean, irrational, and full of rage the next'
stop_words = set(stopwords.words('english'))
filtered_words = " ".join(word for word in test.split() if word.lower() not in stop_words)
print("filtered_words ==> ", filtered_words)

filtered_words ==> depressed hopeless one second mean, irrational, full rage next
```

Hình 2. 9: Ví dụ xóa stop_words

- Loại bỏ những stemwords trong câu:

```
from nltk.stem.snowball import SnowballStemmer
# Removing the word stems using the Snowball Stemmer
Snow_ball = SnowballStemmer("english")
data['review_clean'] = data['review_clean'].apply(
    lambda x: " ".join(Snow_ball.stem(word) for word in x.split()))
```

Hình 2. 10: Xóa những stem_words

Từ thư viện `nltk.stem.snowball` sử dụng hàm `SnowballStemmer` để loại bỏ những stemwords giúp giảm đi sự biến thể của từ giúp tập trung vào ý nghĩa cơ bản của câu.

Ví dụ: chuẩn hóa chuỗi *“i have been taking lyrica for 1 1/2 yrs now”*:

```
from nltk.stem.snowball import SnowballStemmer
test = 'i have been taking lyrica for 1 1/2 yrs now'
snowball_stemmer = SnowballStemmer('english')
stemmed_words = " ".join(snowball_stemmer.stem(word) for word in test.split())
print("stemmed_word ==> ",stemmed_words)
```

stemmed_word ==> i have been take lyrica for 1 1/2 yrs now

Hình 2. 11: Minh họa xóa stem_words

Sau tất cả các bước chuẩn hóa trên, từ dữ liệu ban đầu như sau:

```
data['review']
```

0 "It has no side effect, I take it in combinati...

1 "My son is halfway through his fourth week of ...

2 "I used to take another oral contraceptive, wh...

3 "This is my first time using any form of birth...

4 "Suboxone has completely turned my life around...

...

53761 "I have taken Tamoxifen for 5 years. Side effe...

53762 "I've been taking Lexapro (escitaploprgra...

53763 "I'm married, 34 years old and I have no ...

53764 "I was prescribed Nucynta for severe neck/shou...

53765 "It works!!!"

Name: review, Length: 215063, dtype: object

Hình 2. 12: Dữ liệu chưa chuẩn hóa

```
data['review']
```

0 "it side effect, take combin bystol 5 mg fish ...

1 "mi son halfway fourth week intuniv. becam con...

2 "i use take anoth oral contraceptive, 21 pill ...

3 "this first time use form birth control. im gl...

4 "suboxon complet turn life around. feel health...

...

53761 "i taken tamoxifen 5 years. side effect sever ...

53762 "ive take lexapro (escitaploprgram) sinc febru...

53763 "im married, 34 year old kids. take pill hassl...

53764 "i prescrib nucynta sever neck/should pain. ta...

53765 "it works!!!"

Name: review, Length: 215063, dtype: object

Hình 2. 13: Dữ liệu sau chuẩn hóa

Sau khi chuẩn hóa dữ liệu hoàn tất sẽ tiến hành vector hóa dữ liệu để đưa vào mô hình phân lớp.

```
[13] from sklearn.feature_extraction.text import TfidfVectorizer
```

Hình 2. 14: Cài đặt TfidfVectorizer

Sử dụng thư viện TfidfVectorizer để thực hiện quá trình này.

```
[16] # Create the feature matrix
vectorizer = TfidfVectorizer(lowercase=True, stop_words="english")
reviews = vectorizer.fit_transform(data[['review', 'drugName']])
```

```
[17] data['combined_text'] = data['review'] + ' ' + data['drugName']
```

```
[18] reviews = vectorizer.fit_transform(data['combined_text'])
```

Hình 2. 15: Vector hóa cột review và drugName

Nhóm 2 cột drugName và review lại thành một combined_text và đưa vào mô hình vectorizer.

Kết quả có dạng như sau:

```
print(reviews)

(0, 49675)    0.48237558298774347
(0, 33437)    0.3924687643353888
(0, 19871)    0.4687708405842555
(0, 30188)    0.21846199283378143
(0, 9758)     0.4726797918509412
(0, 12129)    0.3180127952388603
(0, 17221)    0.145780654078066
(1, 22252)    0.18982384939904834
(1, 17225)    0.1285230338260505
(1, 19274)    0.0857602423191681
```

Hình 2. 16: Dữ liệu sau khi vector hóa

CHƯƠNG 3: XÂY DỰNG MÔ HÌNH PHÂN LỚP

1. Thuật toán phân lớp

- Phân lớp là một kỹ thuật trong máy học dùng để phân loại dữ liệu nhị phân hoặc đa lớp, trong đó đầu ra của thuật toán là dữ liệu rời rạc.
- Mô hình phân lớp thường được sử dụng trong các bài toán như: phân loại thư rác, phân loại hình ảnh, dự báo thời tiết,... . Trong bài báo này mô hình phân lớp được sử dụng để phân loại một đánh giá về thuốc là tích cực hay tiêu cực.

2. Tiêu chí đánh giá mô hình phân lớp

- Đối với mô hình phân lớp, có rất nhiều tiêu chí để kiểm định, đánh giá. Có thể kể đến những tiêu chí phổ biến như: Accuracy Score, F1-Score, Confusion Matrix,...

- Trong bài toán này chúng em sử dụng:

Accuracy: đo lường tỷ lệ dự đoán đúng trên tổng số mẫu

$$Arc = \frac{True\ Positives + True\ Negatives}{Total\ Samples}$$

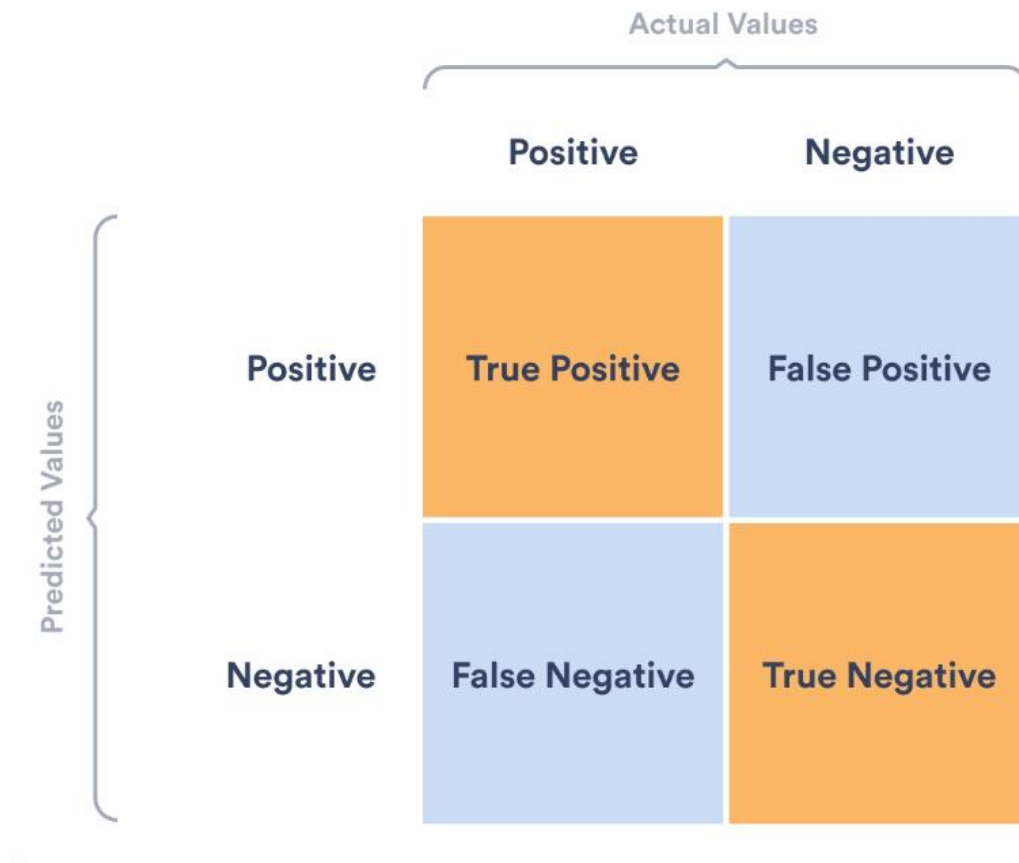
- True Positives: Số lượng mẫu thực tế thuộc lớp dương (positive) và mô hình đã dự đoán đúng là lớp dương.

- True Negative: Số lượng mẫu thực tế thuộc lớp âm (negative) và mô hình đã dự đoán đúng là lớp âm.

- Total Samples: Tổng số mẫu trong tập dữ liệu.

- Confusion Matrix: Cách tính sử dụng accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại

như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Để có thể đánh giá được các giá trị này, chúng ta sử dụng một ma trận được gọi là confusion matrix.



Hình 2. 17: Ma trận nhầm lẫn

Ở đây có thêm 2 chỉ số là False Positive, False Negative:

- False Positive: Số lượng mẫu thực tế thuộc lớp âm nhưng mô hình đã dự đoán sai là lớp dương.
- False Negative: Số lượng mẫu thực tế thuộc lớp dương nhưng mô hình đã dự đoán sai là âm.

3. Xây dựng mô hình phân lớp

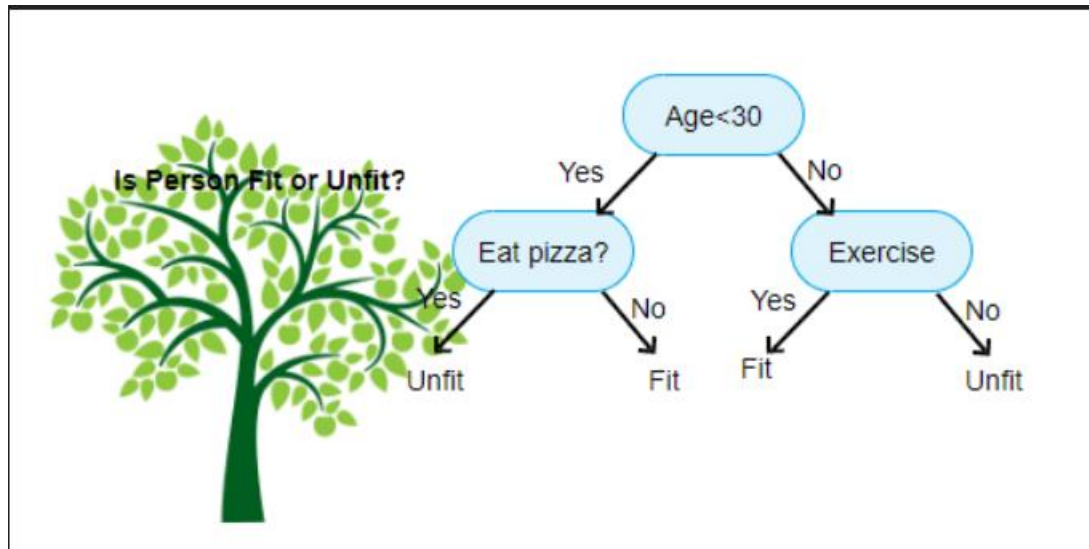
- import accuracy_score và confusion_matrix để đánh giá hiệu quả của mô hình.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

Hình 2. 18: Cài đặt chỉ số đánh giá và ma trận nhầm lẫn

3.1. Cây quyết định (DecisionTreeClassifier)

- Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary), Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.



Hình 2. 19: Cây quyết định

- Sử dụng thư viện sklearn.tree để gọi ra mô hình Cây quyết định, sau đó huấn luyện mô hình trên tập train.

```
from sklearn.tree import DecisionTreeClassifier
```

```
DT_model = DecisionTreeClassifier()
DT_model.fit(X_train, y_train)
```

Hình 2. 20: Cài đặt mô hình Cây quyết định

- Dự đoán mô hình bằng dữ liệu từ tập test.

```
# Evaluate the model
y_pred = DT_model.predict(X_test)
```

Hình 2. 21: Dự đoán mô hình

```
print('accuracy = ',accuracy_score(y_test, y_pred))
cnf_matrix = confusion_matrix(y_test, y_pred)
print('Confusion matrix:')
# Hiển thị ma trận nhầm lẫn bằng heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cnf_matrix, annot=True, cmap='Blues', fmt='d', cbar=False,
            xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

accuracy = 0.6157794888963285

Hình 2. 22: Kết quả dự đoán

- Sau cùng ta được độ chính xác của mô hình Cây quyết định là 61,5 %.

		Confusion Matrix										
Actual	Negative	4771	204	181	114	177	116	157	314	416	757	
	Positive	318	1231	68	48	64	41	46	132	142	236	
		233	73	1126	54	85	44	57	128	135	248	
		147	47	40	829	75	36	54	93	135	179	
		253	90	71	51	1369	60	102	178	222	352	
		180	50	35	40	60	1066	79	167	194	291	
		233	71	55	51	78	58	1562	253	330	469	
		351	104	118	98	136	99	193	3440	661	1175	
		403	128	112	99	166	128	260	557	5314	1911	
		694	202	161	139	245	188	364	923	1576	12400	
		Negative Positive		Predicted								

Hình 2. 23: Ma trận nhầm lẫn cho mô hình Cây quyết định

Với mô hình ta có thể dễ dàng nhìn thấy được:

- Với kết quả dự đoán 1 sao: ta có dự đoán đúng 1 sao là 4771 mẫu trong 7207 mẫu.
- Với kết quả dự đoán 2 sao: ta có dự đoán đúng 2 sao là 1231 mẫu trong 2296 mẫu.

- Với kết quả dự đoán 3 sao: ta có dự đoán đúng 3 sao là 1126 mẫu trong 2210 mẫu.
- Với kết quả dự đoán 4 sao: ta có dự đoán đúng 4 sao là 829 mẫu trong 1635 mẫu.
- Với kết quả dự đoán 5 sao: ta có dự đoán đúng 5 sao là 1369 mẫu trong 2800 mẫu.
- Với kết quả dự đoán 6 sao: ta có dự đoán đúng 6 sao là 1066 mẫu trong 2162 mẫu.
- Với kết quả dự đoán 7 sao: ta có dự đoán đúng 7 sao là 1562 mẫu trong 3160 mẫu.
- Với kết quả dự đoán 8 sao: ta có dự đoán đúng 8 sao là 3440 mẫu trong 6375 mẫu.
- Với kết quả dự đoán 9 sao: ta có dự đoán đúng 9 sao là 5314 mẫu trong 9078 mẫu.
- Với kết quả dự đoán 10 sao: ta có dự đoán đúng 10 sao là 12400 mẫu trong 16892 mẫu.

3.2. Rừng ngẫu nhiên (RandomForestClassifier)

- Là một hợp của các cây quyết định. Là một thuật toán học có giám sát, rừng ngẫu nhiên có thể được sử dụng cho cả hai bài toán phân lớp và hồi quy với hai biến thể là RandomForestClassifier và RandomForestRegressor. Một dữ liệu đầu vào sẽ được đi qua nhiều cây quyết định.

```
from sklearn.ensemble import RandomForestClassifier

RDF_model = RandomForestClassifier()
RDF_model.fit(X_train, y_train)
```

Hình 2. 24: Cài đặt mô hình Rừng ngẫu nhiên

Import thư viện RandomForestClassifier và tiến hành huấn luyện mô hình trên tập train.

```

print('accuracy = ',accuracy_score(y_test, y_pred))
cnf_matrix = confusion_matrix(y_test, y_pred)
print('Confusion matrix:')
# Hiển thị ma trận nhầm lẫn bằng heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cnf_matrix, annot=True, cmap='Blues', fmt='d', cbar=False,
            xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

```

accuracy = 0.7426812483725774

Hình 2. 25: Kết quả dự đoán của mô hình Rừng ngẫu nhiên

- Mô hình cho ra độ chính xác 74,2 %. Tương ứng ta có ma trận nhầm lẫn như bên dưới:

		Confusion Matrix										
Actual	Negative	5718	4	1	0	0	0	2	13	49	1420	
	Positive	302	1445	1	1	1	0	0	9	29	538	
		269	5	1315	0	0	0	0	13	31	550	
		142	0	1	1002	1	0	2	14	48	425	
		230	0	1	2	1606	1	2	19	60	827	
		130	0	0	0	0	1295	0	14	63	660	
		109	1	1	0	0	0	1793	26	83	1147	
		152	0	0	0	0	0	2	3743	168	2310	
		138	2	1	0	1	0	1	33	5500	3402	
		203	1	0	0	2	0	1	17	154	16514	
		Negative Positive		Predicted								

Hình 2. 26: Ma trận nhầm lẫn của mô hình Rừng ngẫu nhiên

Với mô hình ta có thể dễ dàng nhìn thấy được:

- Với kết quả dự đoán 1 sao: ta có dự đoán đúng 1 sao là 5718 mẫu trong 7207 mẫu.
- Với kết quả dự đoán 2 sao: ta có dự đoán đúng 2 sao là 1415 mẫu trong 2296 mẫu.

- Với kết quả dự đoán 3 sao: ta có dự đoán đúng 3 sao là 1315 mẫu trong 2210 mẫu.
- Với kết quả dự đoán 4 sao: ta có dự đoán đúng 4 sao là 1002 mẫu trong 1635 mẫu.
- Với kết quả dự đoán 5 sao: ta có dự đoán đúng 5 sao là 1606 mẫu trong 2800 mẫu.
- Với kết quả dự đoán 6 sao: ta có dự đoán đúng 6 sao là 1295 mẫu trong 2162 mẫu.
- Với kết quả dự đoán 7 sao: ta có dự đoán đúng 7 sao là 1793 mẫu trong 3160 mẫu.
- Với kết quả dự đoán 8 sao: ta có dự đoán đúng 8 sao là 3743 mẫu trong 6375 mẫu.
- Với kết quả dự đoán 9 sao: ta có dự đoán đúng 9 sao là 5500 mẫu trong 9078 mẫu.
- Với kết quả dự đoán 10 sao: ta có dự đoán đúng 10 sao là 16514 mẫu trong 16892 mẫu.

3.3. LightGBM()

- LightGBM (Light Gradient Boosting Machine) là một thuật toán học máy dựa trên kỹ thuật tăng cường (ensemble learning) sử dụng trong bài toán phân loại và hồi quy. Được phát triển bởi Microsoft, LightGBM nhanh và hiệu quả về mặt bộ nhớ, và thường được ưa chuộng trong các tình huống có tập dữ liệu lớn.

```

import lightgbm as lgb
# Similarly LGBMRegressor can also be imported for a regression model.
from lightgbm import LGBMClassifier

```

Hình 2. 27: Cài đặt mô hình LightGBM

```

# Creating an object for model and fitting it on training data set
LGBMC_model = LGBMClassifier()
LGBMC_model.fit(X_train, y_train)

```


Hình 2. 28: Huấn luyện mô hình LightGBM

Sử dụng thư viện lightgbm để sử dụng mô hình LGBMClassifier, sau đó tiến hành huấn luyện mô hình.

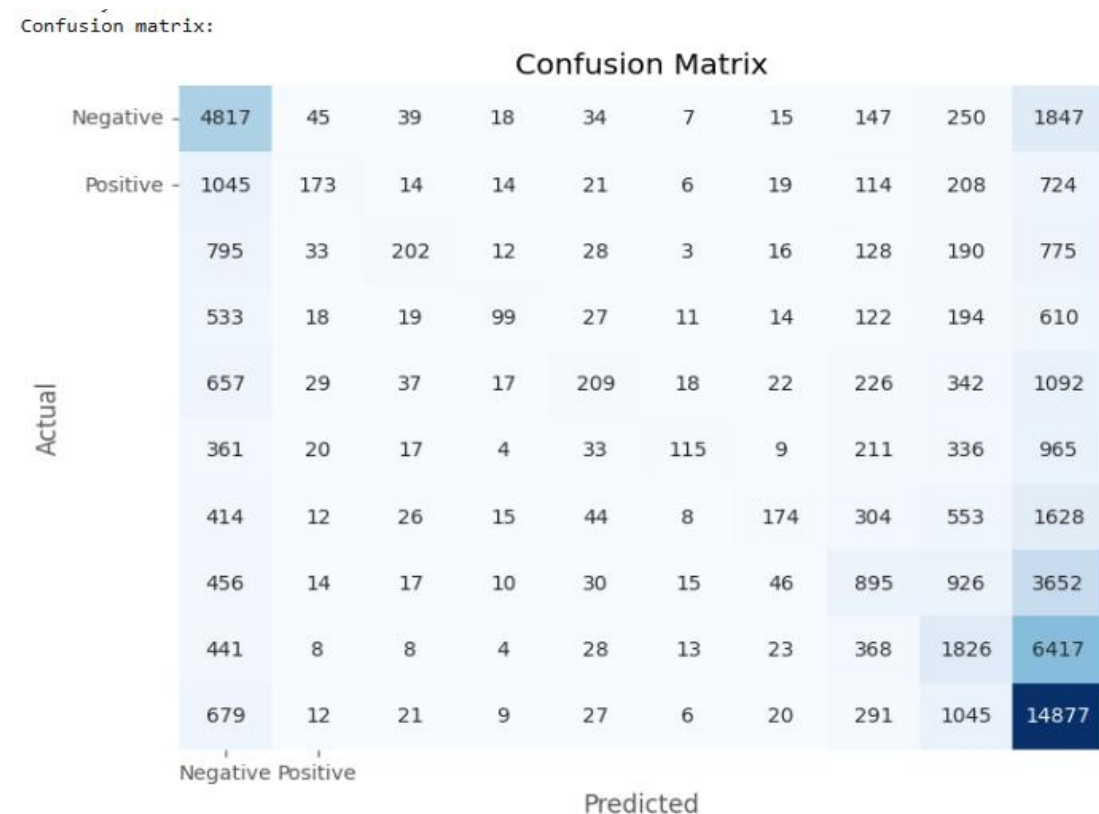
```
# Predicting the Target variable
y_pred = LGBM_model.predict(X_test)

[67] # making the confusion matrix
print('accuracy = ', accuracy_score(y_test, y_pred))
cnf_matrix = confusion_matrix(y_test, y_pred)
print('Confusion matrix:')
# Hiển thị ma trận nhầm lẫn bằng heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cnf_matrix, annot=True, cmap='Blues', fmt='d', cbar=False,
            xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

accuracy = 0.4374018104286676
```

Hình 2. 29: Kết quả dự đoán của mô hình LightGBM

Kết quả dự đoán có độ chính xác 43,7 %.



Hình 2. 30: Ma trận nhầm lẫn của mô hình LightGBM

Với mô hình ta có thể dễ dàng nhìn thấy được:

- Với kết quả dự đoán 1 sao: ta có dự đoán đúng 1 sao là 4817 mẫu trong 7207 mẫu.

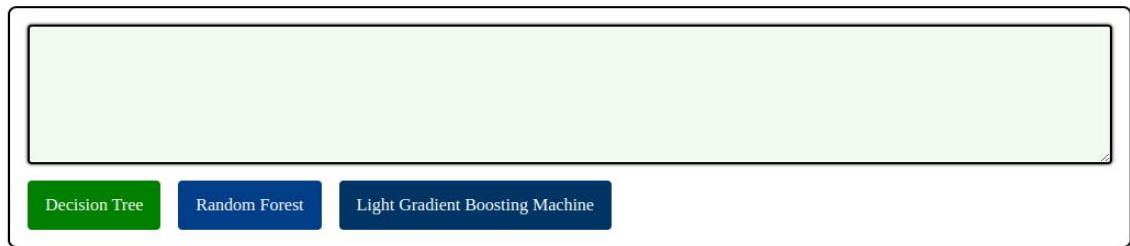
- Với kết quả dự đoán 2 sao: ta có dự đoán đúng 2 sao là 173 mẫu trong 2296 mẫu.
- Với kết quả dự đoán 3 sao: ta có dự đoán đúng 3 sao là 202 mẫu trong 2210 mẫu.
- Với kết quả dự đoán 4 sao: ta có dự đoán đúng 4 sao là 99 mẫu trong 1635 mẫu.
- Với kết quả dự đoán 5 sao: ta có dự đoán đúng 5 sao là 209 mẫu trong 2800 mẫu.
- Với kết quả dự đoán 6 sao: ta có dự đoán đúng 6 sao là 115 mẫu trong 2162 mẫu.
- Với kết quả dự đoán 7 sao: ta có dự đoán đúng 7 sao là 174 mẫu trong 3160 mẫu.
- Với kết quả dự đoán 8 sao: ta có dự đoán đúng 8 sao là 895 mẫu trong 6375 mẫu.
- Với kết quả dự đoán 9 sao: ta có dự đoán đúng 9 sao là 1826 mẫu trong 9078 mẫu.
- Với kết quả dự đoán 10 sao: ta có dự đoán đúng 10 sao là 14877 mẫu trong 16892 mẫu.

4. Đánh giá.

- Qua ba mô hình phân loại đánh giá thì với mô hình Random Forest cho kết quả dự đoán tốt nhất với độ chính xác là 74.5%.

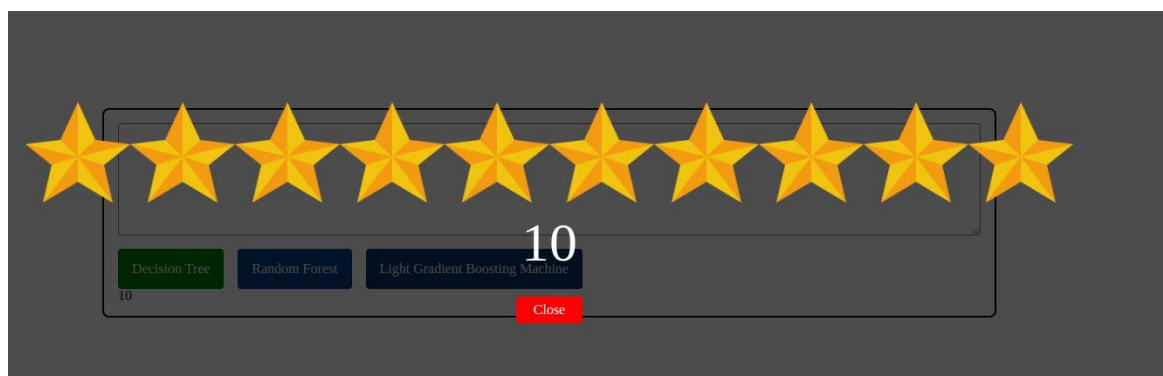
5. Xây dựng trang web phân loại đánh giá

- **BackEnd** : FastAPI + Model được lưu về sau khi đã được huấn luyện.
- **FrontEnd** : ReactJS.



Hình 5. 1: Giao diện dự đoán

- Người dùng nhập vào ô trống đoạn đánh giá sau đó chọn mô hình cần dự đoán, ngay lập tức cho ra kết quả dự đoán được kết quả.



Hình 5. 2: Kết quả dự đoán

- Kết quả đạt được là phân loại được đánh giá mà người dùng nhập.