ĐẠI HỌC UEH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ
KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH

ĐỒ ÁN KẾT THÚC HỌC PHẦN
MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Dự đoán đề xuất sản phẩm từ đánh giá khách hàng về quần áo nữ trên trang thương mại điện tử

Giảng viên	:	Ts. Đặng Ngọc Hoàng Thành
Mã lớp học phần	:	23C1INF50907602
Nhóm	:	5
Danh sách thành viên	:	Dương Mỹ Quỳnh
		Trương Vũ Phương Quỳnh
		Đinh Công Thành

TP Hồ Chí Minh, ngày 16 tháng 12 năm 2023

LỜI CẢM ƠN

Chúng em chân thành cảm ơn khoa Công nghệ thông tin kinh doanh – Trường Công nghệ và Thiết kế, Đại học Kinh tế TP. Hồ Chí Minh vì đã tích hợp học phần Xử lý ngôn ngữ tự nhiên vào chương trình đào tạo. Điều này đã mang lại cho chúng em cơ hội học tập và tiếp cận với những kiến thức mới, quan trọng cho sự phát triển trong ngành nghề sau này.

Không kém phần quan trọng, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến các nguồn học liệu mà chúng em đã sử dụng trong quá trình thực hiện đồ án. Những tài liệu này đã là nguồn thông tin quý báu, giúp chúng em hiểu rõ hơn về đề tài và cung cấp nền tảng cho quá trình nghiên cứu và triển khai.

Chúng em cũng muốn gửi lời cảm ơn tới tất cả các bạn học cùng nhóm, những người đã chia sẻ, đóng góp ý kiến và hỗ trợ chúng em trong suốt quá trình thực hiện đồ án. Sự hỗ trợ này không chỉ giúp chúng em vượt qua những khó khăn mà còn tạo ra một môi trường học tập tích cực và hữu ích.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn chân thành nhất đến TS. Đặng Ngọc Hoàng Thành, người đã là nguồn động viên và người hướng dẫn tận tâm trong quá trình thực hiện đồ án. Sự kiến thức sâu rộng và nhiệt huyết của thầy đã giúp chúng em tiếp cận và ứng dụng các khái niệm và kỹ thuật một cách hiệu quả nhất.

Mặc dù chúng em đã cố gắng hết sức, nhưng chắc chắn rằng đồ án vẫn còn những khía cạnh có thể được cải thiện. Chúng em mong nhận được sự góp ý chân thành từ thầy để có thể hoàn thiện hơn, và chúng em sẽ tiếp tục nỗ lực để phát triển đề tài này trong tương lai.

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1. GIỚI THIỆU VẤN ĐỀ NGHIÊN CỨU

1.1. Tính cấp thiết của đề tài

Trong bối cảnh công nghệ phát triển ngày nay, thương mại điện tử đã thâm nhập sâu vào cuộc sống hàng ngày, trở thành một phần quan trọng không thể thiếu. Người tiêu dùng ngày càng phụ thuộc vào mua sắm trực tuyến, đặc biệt là trong lĩnh vực thời trang nữ. Để hiểu rõ hơn về ý kiến của người tiêu dùng đối với sản phẩm thời trang nữ, việc áp dụng kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) là quan trọng. Mục tiêu là phân tích đánh giá khách hàng để cung cấp cái nhìn khách quan, xác định mức độ hài lòng, và khuyến khích trong việc giới thiệu sản phẩm.

Tuy nhiên, đánh giá khách hàng trên các trang thương mại điện tử thường đối mặt với một số vấn đề. Số lượng đánh giá không đủ, đặc biệt là cho những sản phẩm không có đánh giá, khó khăn trong việc đánh giá chính xác nhu cầu và mong muốn của khách hàng. Chất lượng đánh giá cũng là một thách thức, khi một số đánh giá không độc lập, thiếu trung thực, thậm chí mang tính xúc phạm. Ngoài ra, việc phân tích và xử lý dữ liệu từ các đánh giá cũng gặp khó khăn do sự đa dạng và phi cấu trúc của ngôn ngữ tự nhiên, làm cho quá trình này trở nên phức tạp và đòi hỏi nhiều thời gian.

Đồ án của nhóm bao gồm các bước chính như xử lý dữ liệu, phân tích ngôn ngữ tự nhiên để trích xuất thông tin quan trọng từ nhận xét, và tạo mô hình học máy, mô hình học sâu để dự đoán và phân loại đánh giá. Ngoài ra, nghiên cứu có thể tập trung vào phân tích xu hướng, ý kiến phổ biến và các yếu tố ảnh hưởng đến sự hài lòng của khách hàng đối với sản phẩm thời trang. Đề xuất sản phẩm cho người thân, bạn bè từ phía khách hàng là biểu hiện của sự hài lòng và cũng tạo động lực cho doanh nghiệp cải thiện dịch vụ và sản phẩm.

1.2. Mục tiêu đề tài

Mục tiêu đồ án được nhóm đem đến sẽ xử lý ngôn ngữ tự nhiên các đánh giá của khách hàng từ đó đưa ra kết luận liệu khách hàng có mua lại hay khuyến khích các nhà tiêu dùng sau mua sản phẩm đó hay không. Đồ án giúp người tiêu dùng có cái nhìn khách quan về sản phẩm mà họ dự tính mua trên các sàn thương mại điện tử ngoài ra nghiên cứu không chỉ hướng tới việc nâng cao trải nghiệm của người tiêu dùng mà còn

hỗ trợ doanh nghiệp thương mại điện tử và nhãn hiệu thời trang để cải thiện chất lượng sản phẩm, dịch vụ khách hàng và chiến lược tiếp thị dựa trên ý kiến của người tiêu dùng.

Đối với các khách hàng đánh giá khuyến khích mua lại nghĩa là họ hài lòng với cả sản phẩm lẫn dịch vụ của nhãn hàng vì thế nên các nhà bán hàng cũng cần chăm sóc và tập trung tương tác, duy trì các dịch vụ tốt đối với tệp khách hàng này để có được các khách hàng trung thành với thương hiệu. Đối với các khách hàng đánh giá không khuyến khích mua lại thì các nhà cung cấp cần tìm hiểu xem sản phẩm của họ đang gặp vấn đề gì và cần được cải thiện ở những điểm nào từ đó sửa đổi và cung cấp lại dịch vụ tốt nhất.

Vì vậy, đồ án nhóm đem đến không chỉ là một nghiên cứu đơn thuần về đánh giá sản phẩm, mà đóng vai trò quan trọng trong sự phát triển của thương mại điện tử và phân tích dữ liệu, mang lại lợi ích to lớn cho cả người tiêu dùng và doanh nghiệp thời trang.

2. THIẾT LẬP MÔ HÌNH TỔNG QUÁT

Đối với xây dựng mô hình phân lớp, nhóm đặt biến mục tiêu là Recommended IND để tìm mô hình dự đoán xem điểm đánh giá của khách hàng dựa trên các biến thu thập được về bài đánh giá và người mua hàng. Các phương pháp sử dụng bao gồm: Naive Bayes, Decision Tree, Logistic Regression và LSTM. Sử dụng các điểm: Accuracy score, MSE, MAE và R-square tính toán độ chính xác của từng mô hình.

3. CƠ SỞ LÝ THUYẾT

3.1. Mô hình Naive Bayes

Naïve Bayes là một thuật toán học máy đơn giản sử dụng quy tắc Bayes và giả định mạnh rằng các thuộc tính là độc lập có điều kiện trên lớp. Mặc dù giả định về sự độc lập này thường không phản ánh đúng thực tế, nhưng naïve Bayes thường mang lại kết quả phân loại có độ chính xác cao. Sự đơn giản của thuật toán, kết hợp với hiệu suất tính toán nhanh và nhiều ưu điểm khác, giúp naïve Bayes trở thành một lựa chọn phổ biến trong thực tế.

Ưu điểm	Hạn chế
Nhanh chóng: Huấn luyện nhanh, ngay cả với bộ dữ liệu lớn.	Yêu cầu dữ liệu: Cần một lượng dữ liệu lớn để đạt độ chính xác cao.

Đơn giản: Có thể tự động loại bỏ các tính năng không liên quan.	Độ chính xác: Có thể kém chính xác hơn so với một số thuật toán khác trên một số tập dữ liệu hoặc tác vụ phức tạp.
Hiệu quả: Hiệu suất phân loại tốt trên nhiều loại dữ liệu, đặc biệt là các tác vụ đơn giản.	Giả định: Giả định các thuộc tính độc lập có thể ảnh hưởng đến độ chính xác.

Bảng 1: Ưu điểm và hạn chế của mô hình Naive Bayes

3.2. Mô hình Decision Tree

Mô hình cây quyết định là một công cụ quan trọng trong lĩnh vực máy học, nó hoạt động bằng cách phân tách dữ liệu thành các nhóm con dựa trên giá trị của các thuộc tính. Cây bao gồm các nút quyết định, nút lá và nút gốc. Mỗi nút quyết định đại diện cho một quy tắc hoặc điều kiện, chia dữ liệu thành các nhánh. Quá trình chọn thuộc tính để phân chia sử dụng các phương pháp như Information Gain, Gini Index. Cây quyết định có thể dễ bị quá mức chính xác trên dữ liệu huấn luyện (overfitting), và pruning là kỹ thuật được sử dụng để ngăn chặn hiện tượng này. Cây quyết định thường được áp dụng trong bài toán phân loại và dự đoán, như việc quyết định xem một email có phải là spam hay không dựa trên các thuộc tính như từ vựng xuất hiện, độ dài email, số lượng liên kết, và các yếu tố khác.

Ưu điểm	**Hạn chế**
Nhanh chóng và đơn giản: Huấn luyện nhanh, dễ hiểu và giải thích.	Huấn luyện lâu: Trên tập dữ liệu lớn, thời gian huấn luyện có thể dài hơn một số thuật toán khác.
Kết quả chính xác: Có thể đạt được độ chính xác cao trên nhiều loại dữ liệu.	Vấn đề "sao chép": Cây có thể phát triển phức tạp không cần thiết do lặp lại các điểm chia dữ liệu.
Dễ hiểu: Cấu trúc cây đơn giản, dễ dàng trực quan hóa và giải thích kết quả.	Dễ gặp overfitting: Xu hướng dễ học thuộc lòng dữ liệu huấn luyện, dẫn đến hiệu suất kém trên dữ liệu mới.
Ít tốn bộ nhớ: Cấu trúc cây tiết kiệm bộ nhớ so với một số thuật toán khác.	

Bảng 2: Ưu điểm và hạn chế của mô hình Decision Tree

3.3. Mô hình Logistic Regression

Mô hình Logistic Regression là một phương pháp thống kê được sử dụng để dự đoán xác suất của một biến phụ thuộc nhị phân dựa trên các biến độc lập. Đối với mỗi quan sát, mô hình tính toán giá trị xác suất sử dụng hàm Logistic (hoặc Sigmoid), chuyển đổi đầu ra thành một giá trị nằm trong khoảng từ 0 đến 1. Các hệ số của mô hình được học từ dữ liệu để tối ưu hóa dự đoán xác suất. Khi áp dụng một ngưỡng quyết định, mô hình Logistic Regression có thể dự đoán xem một sự kiện xảy ra hay không, thường được sử dụng trong các bài toán như dự đoán chuyển đổi khách hàng, mắc bệnh, và nhiều ứng dụng khác.

Ưu điểm	Hạn chế
Dễ sử dụng: Dễ cài đặt, diễn giải và huấn luyện hiệu quả.	Yêu cầu dữ liệu: Không phù hợp khi số lượng quan sát ít hơn số lượng features, có thể dẫn đến overfitting.
Giải thích: Cung cấp thông tin về tầm quan trọng và hướng ảnh hưởng của các biến dự đoán.	Biến phụ thuộc rời rạc: Chỉ dự đoán được các hàm rời rạc, không phù hợp với biến liên tục.
Chính xác: Đạt độ chính xác tốt với nhiều bộ dữ liệu đơn giản, đặc biệt khi dữ liệu phân tách tuyến tính.	Mối quan hệ phức tạp: Khó nắm bắt các mối quan hệ phức tạp giữa các biến, có thể kém hiệu quả hơn các thuật toán như Neural Networks trong các trường hợp này.

Bảng 3: Ưu điểm và hạn chế của mô hình Logistic Regression

3.4. Mô hình LSTM

Mô hình Long Short-Term Memory (LSTM) là một loại mô hình Recurrent Neural Networks (RNN) trong học máy. Với cấu trúc phức tạp, LSTM sử dụng các cổng như Forget Gate, Input Gate, và Output Gate để kiểm soát quá trình lưu trữ và truyền thông tin qua thời gian. Điều này giúp LSTM giải quyết được vấn đề mất thông tin và học được các phụ thuộc dài hạn trong dữ liệu chuỗi. Thường được ứng dụng trong dự đoán chuỗi thời gian, xử lý ngôn ngữ tự nhiên và các bài toán khác liên quan đến dữ liệu chuỗi, LSTM là một công cụ mạnh mẽ trong lĩnh vực học máy.

Ưu điểm	Hạn chế
Mô hình ngôn ngữ đa cấp độ: LSTM cho phép xây dựng mô hình ngôn ngữ ở nhiều cấp độ khác nhau, từ ký tự đến đoạn văn, giúp nó phù hợp cho nhiều ứng dụng văn bản khác nhau.	Yêu cầu tài nguyên lớn: LSTM đòi hỏi nhiều tài nguyên và thời gian đào tạo, gây ra vấn đề hiệu suất và không hiệu quả về mặt phần cứng.
Xử lý chuỗi dữ liệu dài hạn: LSTM có khả năng ghi nhớ thông tin trong thời gian dài, làm cho chúng phù hợp cho các ứng dụng như dịch ngôn ngữ và tạo văn bản có ý nghĩa dài.	Vấn đề biến mất độ dốc chưa được hoàn toàn giải quyết: Mặc dù LSTM giải quyết vấn đề này, nhưng vẫn còn những thách thức liên quan đến việc truyền thông tin qua các ô.
Khả năng áp dụng cho nhiều lĩnh vực: LSTM không chỉ giải quyết vấn đề biến mất độ dốc mà còn linh hoạt và đa dạng, áp dụng rộng rãi từ xử lý ngôn ngữ tự nhiên đến xử lý ảnh và âm nhạc.	Khả năng bị quá mức (overfitting): LSTM dễ bị quá mức, và việc áp dụng thuật toán dropout để giảm overfitting là một thách thức.

Bảng 4: Ưu điểm và hạn chế của mô hình LSTM

3.5. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là quá trình chuẩn bị và làm sạch dữ liệu trước khi áp dụng các phương pháp phân tích hoặc xây dựng mô hình. Mục tiêu của tiền xử lý là tối ưu hóa chất lượng dữ liệu để đảm bảo rằng nó đủ chính xác, đầy đủ và có thể phục vụ mục đích phân tích hay mô hình hóa dữ liệu. Các bước tiền xử lý thường bao gồm loại bỏ dữ liệu trùng lặp, điền giá trị thiếu, chuẩn hóa định dạng, và xử lý các ngoại lệ. Quá trình này không chỉ giúp cải thiện hiệu suất của mô hình, mà còn giúp người phân tích hiểu rõ hơn về tính chất và cấu trúc của dữ liệu.

CHƯƠNG 2: TỔNG QUAN BỘ DỮ LIỆU

1. MÔ TẢ BỘ DỮ LIỆU

Xem xét bộ dữ liệu, bao gồm 23486 hàng tương ứng với các đánh giá của khách hàng và 10 biến đặc trưng:

Tên Biến	Ý Nghĩa
Biến phụ thuộc	
Recommended IND	Khách hàng đề xuất cho khách hàng khác mua sản phẩm không? 1- có. 0- không
Biến độc lập	
Clothing ID	Mã sản phẩm
Age	Tuổi người đánh giá
Title	Tiêu đề bài đánh giá
Review Text	Nội dung bài đánh giá
Rating	Điểm đánh giá trên thang từ 1 Tệ nhất đến 5 Tốt nhất.
Positive Feedback Count	Số lượng phản hồi thấy đánh giá này là tích cực.
Division Name	Cấp độ sản phẩm.
Department Name	Tên bộ phận của sản phẩm
Class Name	Loại sản phẩm

Bảng 5: Mô tả bộ dữ liệu

2. PHÂN TÍCH KHÁM PHÁ

Trước khi đưa dữ liệu vào mô hình, nhóm tiến hành phân tích khám phá các biến của bộ dữ liệu bằng các câu lệnh sau:

```
data.head()
```

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses

Hình 1: 5 hàng đầu của bộ dữ liệu

→ Lệnh này cho phép chúng ta xem một số hàng đầu tiên của bộ dữ liệu. Bằng cách sử dụng lệnh này, chúng ta có thể nhanh chóng xem qua các dòng đầu tiên và có cái nhìn tổng quan về dữ liệu.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23486 entries, 0 to 23485
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Unnamed: 0               23486 non-null  int64 
 1   Clothing ID              23486 non-null  int64 
 2   Age                      23486 non-null  int64 
 3   Title                    19676 non-null  object
 4   Review Text              22641 non-null  object
 5   Rating                   23486 non-null  int64 
 6   Recommended IND          23486 non-null  int64 
 7   Positive Feedback Count  23486 non-null  int64 
 8   Division Name            23472 non-null  object
 9   Department Name          23472 non-null  object
 10  Class Name               23472 non-null  object
dtypes: int64(6), object(5)
memory usage: 2.0+ MB
```

Hình 2: Thông tin tóm tắt của bộ dữ liệu

→ Lệnh này cung cấp thông tin tóm tắt về dữ liệu, bao gồm số lượng hàng, số lượng cột, kiểu dữ liệu của từng cột, và nhiều thông tin khác. Bằng cách sử dụng lệnh này, chúng ta có thể hiểu cấu trúc của bộ dữ liệu và kiểm tra kiểu dữ liệu của từng cột.

- Các cột và số lượng giá trị không null trong mỗi cột như sau:
    - Review Text: 22,641 giá trị không null, kiểu dữ liệu là object (chuỗi).
    - Recommended IND: 23,486 giá trị không null, kiểu dữ liệu là int64.

```
data.describe()
```

	Unnamed: 0	Clothing ID	Age	Rating	Recommended IND	Positive Feedback Count
count	23486.000000	23486.000000	23486.000000	23486.000000	23486.000000	23486.000000
mean	11742.500000	918.118709	43.198544	4.196032	0.822362	2.535936
std	6779.968547	203.298980	12.279544	1.110031	0.382216	5.702202
min	0.000000	0.000000	18.000000	1.000000	0.000000	0.000000
25%	5871.250000	861.000000	34.000000	4.000000	1.000000	0.000000
50%	11742.500000	936.000000	41.000000	5.000000	1.000000	1.000000
75%	17613.750000	1078.000000	52.000000	5.000000	1.000000	3.000000
max	23485.000000	1205.000000	99.000000	5.000000	1.000000	122.000000

Hình 3:Tóm tắt thống kê về các giá trị số

→ Lệnh này cung cấp một tóm tắt thống kê về các giá trị số của dữ liệu, bao gồm min, max, mean, median, và các phần centile khác. Điều này giúp chúng ta có cái nhìn tổng quan về phân bố và đặc tính của dữ liệu số.

- Dựa vào bảng kết quả trên, nhóm rút ra một số kết luận về phân phối và xu hướng của biến Recommended IND trong tập dữ liệu:
    - Giá trị trung bình (mean) của cột này là 0.822017, cho thấy khoảng 82% người dùng đã đồng ý đề xuất sản phẩm đã mua.
    - Độ lệch chuẩn (std) là 0.382507, cho thấy sự biến động tương đối lớn trong việc đề xuất sản phẩm.

3. NHẬN DIỆN CÁC BIẾN

Cột ‘Unnamed: 0’ là cột định danh về số thứ tự của sản phẩm không có ý nghĩa trong việc phân tích và dự đoán nên nhóm cũng sẽ bỏ cột này ra khỏi bộ dữ liệu.

```
data = data.drop(['Unnamed: 0'], ], axis=1)
```

Nhóm sử dụng câu lệnh sau để xem số danh mục của từng biến:

```
print('Số danh mục của mỗi biến')
print(data.nunique())
```

Tên biến	Danh mục của mỗi biến
Clothing ID	1206
Age	77
Title	13993
Review Text	22634
Rating	5

Recommended IND	2
Division Name	3
Department Name	6
Class Name	20

Bảng 6: Danh mục của mỗi biến trong bộ dữ liệu

Từ bảng trên, nhóm nhận thấy cần chuyển đổi kiểu dữ liệu của Recommended IND trong bảng dữ liệu từ số sang object, nhằm biểu diễn chúng như các biến phân loại. Cụ thể, các bước trong đoạn code như sau:

```
data["Recommended IND"] = data["Recommended IND"].astype(object)
```

Đoạn code này tương tự chuyển đổi kiểu dữ liệu của cột "Recommended IND" từ số sang object. Cột "Recommended IND" chỉ bao gồm các giá trị 0 và 1, và việc chuyển đổi sang kiểu object giúp biểu diễn nó như một biến phân loại.

→ Bằng cách chuyển đổi kiểu dữ liệu này, cột "Recommended IND" được xử lý như các biến phân loại trong bảng dữ liệu, cho phép thực hiện các phân tích và trực quan hóa phù hợp với tính chất của nó.

4. THỐNG KÊ DỮ LIỆU

Nhóm sử dụng câu lệnh sau để xem dataframe có bao nhiêu dòng, bao nhiêu cột:

```
print('Bộ dữ liệu có: ', data.shape[0], 'dòng', data.shape[1], 'cột')
print('Các cột dữ liệu:\n', '\n'.join(data.columns[0:]))
```

```
Bộ dữ liệu có:  23486 dòng 10 cột
Các cột dữ liệu:
 Clothing ID
Age
Title
Review Text
Rating
Recommended IND
Positive Feedback Count
Division Name
Department Name
Class Name
```

CHƯƠNG 3: TIỀN XỬ LÝ DỮ LIỆU

1. LÀM SẠCH DỮ LIỆU

1.1. Xử lý dữ liệu bị thiếu

Nhóm sử dụng câu lệnh sau để kiểm tra những cột dữ liệu bị thiếu:

```
print('Dữ liệu bị thiếu:')
data.isnull().sum().sort_values(ascending=False)
```

```
Dữ liệu bị thiếu:
Title                        3810
Review Text                   845
Division Name                  14
Department Name                14
Class Name                     14
Clothing ID                     0
Age                             0
Rating                          0
Recommended IND                 0
Positive Feedback Count         0
dtype: int64
```

→ Từ kết quả trên, nhóm nhìn thấy các cột 'Department Name', 'Division Name', 'Class Name' có 14 dòng dữ liệu bị khuyết thiếu chỉ chiếm gần 0.06% so với tổng số 23486 dòng dữ liệu thu thập được. Số liệu thiếu chiếm rất ít nên nhóm quyết định xóa các dòng bị khuyết khiếu dữ liệu.

→ Ngoài ra các cột 'Title', 'Review Text' thiếu lần lượt 3810 và 845 chiếm 16% và 3% so với tổng số 23486 dòng dữ liệu thu thập được. Với số phần trăm này là khá nhiều, nhóm phải xử lý bằng cách lấy trung vị hoặc trung bình. Nhưng đây là đánh giá của từng cá nhân nên việc lấy trung vị hay trung bình đều rất nhạy cảm, nên nhóm quyết định bỏ các dòng bị thiếu này đi.

Nhóm sử dụng câu lệnh sau để xoá các dòng dữ liệu bị thiếu:

```
data = data.dropna(subset=['Division Name', 'Department Name', 'Class Name', 'Title', 'Review Text'])
```

1.2. Xử lý dữ liệu không nhất quán

Nhóm sử dụng hàm unique() để kiểm tra danh sách giá trị của các cột, để có thể kiểm tra bộ dữ liệu có tồn tại dữ liệu không nhất quán hay không.

```
# Recommended IND
print('Recommended IND: ',data['Recommended IND'].unique()) # 0 với 1
```

```
Recommended IND:  [0 1]
```

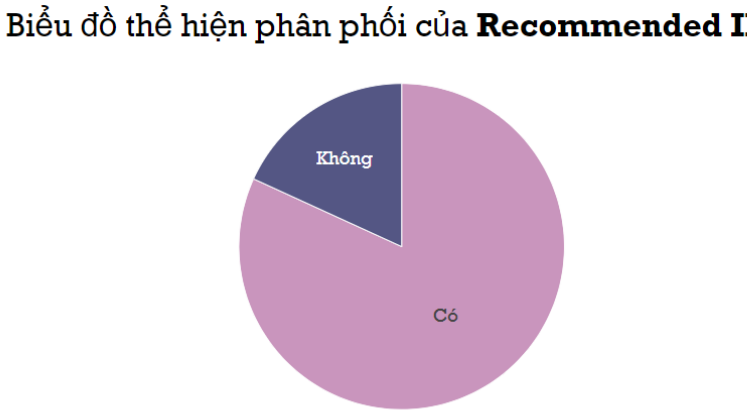
→ Từ kết quả trên ta có thể thấy không có xảy ra trường hợp dữ liệu không nhất quán nên nhóm sẽ không tiến hành xử lý dữ liệu không nhất quán này.

2. PHÂN TÍCH ĐƠN BIẾN

2.1. Biểu đồ thể hiện phân phối của Recommended IND

```
recommended = (
    data['Recommended IND']
    .value_counts()
    .to_frame()
    .reset_index()
    .rename(columns={'index':'Recommended', 'Recommended
IND':'Count'})
    .sort_values(by=['Recommended'], ascending=True)
    .replace([0, 1], ['Có', 'Không'])
          )
colors = ['#545684','#C995BD']
fig = go.Figure(data=[go.Pie(labels=recommended['Recommended'],
                             values=recommended['Count'])])
fig.update_traces(hoverinfo='percent',
                  textinfo='label',
                  textfont_size=20,
                  marker=dict(colors=colors,
                              line=dict(color='white', width=1)))
fig.update_layout(showlegend=False,
                  title_text="Biểu đồ thể hiện phân phối của
<b>Recommended IND<b>",
                  title_x=0.5,
                  font=dict(family="Rockwell, sans-serif", size=25,
color='#000000'))

fig.show()
```



Hình 4: Biểu đồ thể hiện phân phối của Recommended IND

→ Nhận xét:

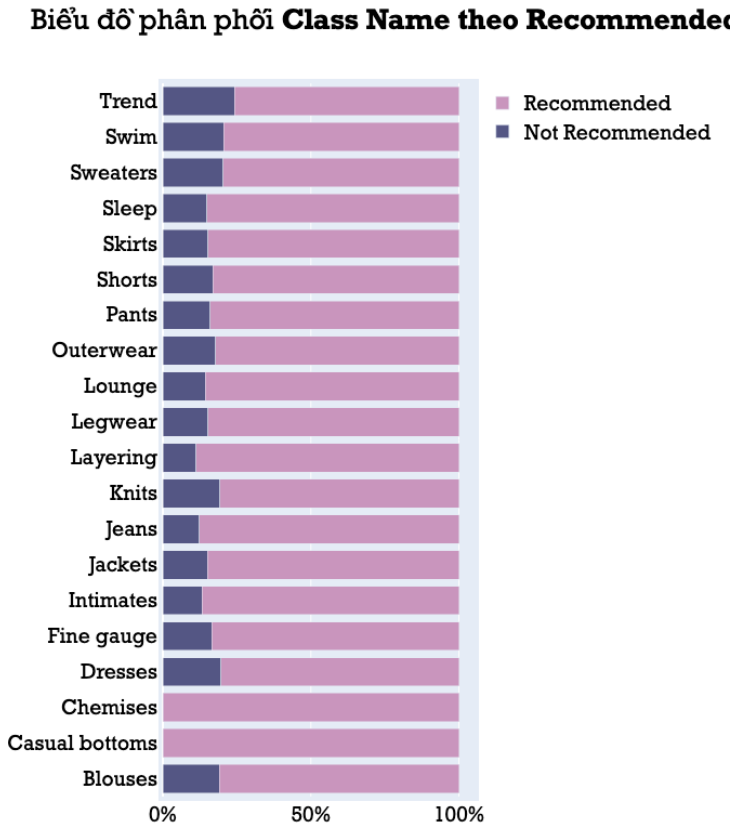
- Biểu đồ tròn thể hiện phân phối của Recommended IND cho thấy có 81,8% khách hàng sẽ giới thiệu sản phẩm cho bạn bè và gia đình. 18,2% khách hàng sẽ không giới thiệu sản phẩm.
- Đây là một kết quả rất tích cực, cho thấy khách hàng hài lòng với sản phẩm và dịch vụ của doanh nghiệp. Điều này có thể giúp doanh nghiệp thu hút thêm khách hàng mới và tăng doanh số bán hàng.

2.2. Biểu đồ phân phối phân loại sản phẩm theo Recommended IND

```
classes = (
    data
    .groupby(['Recommended IND', 'Class Name'])
    .size()
    .to_frame()
    .rename(columns={0:'Count'})
    .reset_index()
          )
a = classes.groupby('Class Name')['Count'].transform('sum')
classes['Count'] = classes['Count'].div(a)
classes = classes.pivot(index='Class Name', columns='Recommended IND')
fig = go.Figure()
fig.add_trace(go.Bar(
    y=classes.index,
    x=classes.iloc[:,0],
    name='Not Recommended',
    orientation='h',
    marker=dict(
        color='#545684')
    ))
fig.add_trace(go.Bar(
    y=classes.index,
    x=classes.iloc[:,1],
    name='Recommended',
    orientation='h',
    marker=dict(
        color='#C995BD')
    ))
fig.update_layout(barmode='stack')

fig.update_layout(
                title = 'Biểu đồ phân phối <b>Class Name<b> theo
<b>Recommended<b> ',
                barmode='stack',
                autosize=False,
                width=680,
                height=800,
                font=dict(family="Rockwell, sans-serif", size=18,
color='#000000'),
```

```
                margin=dict(
                  l=150,
                  r=100,
                   b=30,
                   t=100,
                   pad=4
                          ))
fig.layout.xaxis.tickformat = ',.0%'
fig.show()
```



Hình 5: Biểu đồ phân phối Class Name theo Recommended

→ Nhận xét:

- Tỷ lệ khách hàng đề xuất sản phẩm cao cho thấy khách hàng hài lòng với sản phẩm và dịch vụ của doanh nghiệp. Điều này có thể giúp doanh nghiệp thu hút thêm khách hàng mới và tăng doanh số bán hàng.
- Các phân loại sản phẩm được đề xuất nhiều nhất là các sản phẩm thời trang, thể thao và ngoài trời. Điều này cho thấy khách hàng có xu hướng đề xuất các sản phẩm có thiết kế đẹp, chất lượng tốt và phù hợp với nhu cầu của họ.
- Các phân loại sản phẩm được đề xuất ít nhất là các sản phẩm nội y và áo sơ mi. Điều này có thể là do các sản phẩm này là sản phẩm cá nhân, khách hàng thường không muốn đề xuất cho người khác.

2.3. WordCloud thể hiện tần suất xuất hiện của các từ trong Review Text

- Trích xuất các từ có độ dài lớn hơn 3 từ trong cột "Review Text"

```
count_words = data["Review Text"].apply(lambda x: " ".join([w for w in
x.split() if len(w)>3]))
```

- Sau khi đã trích xuất, ta sẽ đếm số lần xuất hiện của mỗi từ

```
count_words = pd.Series(" ".join(count_words).split()).value_counts()
print(count_words)
```

- Tạo WordCloud từ tần suất xuất hiện của các từ

```
this                        18556
with                        11345
that                         8247
have                         7124
dress                        6941
                            ...
(located                        1
sites!)                         1
didn&amp;#39;t                  1
5&amp;#39;6&amp;quot;,          1
platinum                        1
Length: 34217, dtype: int64
```

```
wordcloud = WordCloud(width=800, height=400,
background_color='white').generate_from_frequencies(frequencies=count_
words)

fig, ax = plt.subplots(figsize=(14, 4))
ax.imshow(wordcloud, interpolation='bilinear')
ax.axis('off')
ax.set_title("WordCloud\n")

plt.show()
```

Hình 6: WordCloud thể hiện tần suất xuất hiện của các từ trong Review Text

→ Nhận xét:

- Kết quả trên cho thấy khách hàng đánh giá cao chất lượng của sản phẩm quần áo nữ. Điều này được thể hiện qua việc các từ và cụm từ như "this", "very", "love", "great", "really", "like" được sử dụng nhiều. Các nhà bán hàng có thể sử dụng thông tin này để tiếp tục cải thiện chất lượng sản phẩm của mình, chẳng hạn như:
  + Sử dụng chất liệu tốt hơn
  + Thiết kế sản phẩm đẹp hơn
  + Hoàn thiện sản phẩm tốt hơn

- Ngoài ra, khách hàng cũng quan tâm đến kích thước và kiểu dáng của sản phẩm quần áo nữ. Điều này được thể hiện qua việc các từ và cụm từ như "size", "dress", "fit" được sử dụng nhiều. Các nhà bán hàng có thể sử dụng thông tin này để cải thiện kích thước và kiểu dáng sản phẩm của mình, chẳng hạn như:
  + Cung cấp nhiều kích thước hơn cho sản phẩm
  + Thiết kế sản phẩm phù hợp với nhiều kiểu dáng cơ thể hơn

- Cuối cùng, khách hàng có xu hướng đánh giá sản phẩm quần áo nữ sau khi đã sử dụng. Điều này được thể hiện qua việc các từ và cụm từ như "wear", "ordered" được sử dụng nhiều. Các nhà bán hàng có thể sử dụng thông tin này để cải thiện dịch vụ chăm sóc khách hàng của mình, chẳng hạn như:
  + Tạo điều kiện thuận lợi cho khách hàng đánh giá sản phẩm
  + Trả lời các câu hỏi của khách hàng về sản phẩm
  + Giải quyết các vấn đề của khách hàng sau khi mua sản phẩm

CHƯƠNG 4: MÔ HÌNH DỰ ĐOÁN

1. CÁC MÔ HÌNH DỰ ĐOÁN

Trước khi đưa dữ liệu vào mô hình, nhóm tiến hành một vài bước như sau:

- Tạo một bộ dữ liệu mới chỉ có biến mục tiêu là Recommended IND và biến Review Text

```
df=data[['Review Text','Recommended IND']]
```

- Tiếp theo nhóm thực hiện công việc xử lý văn bản.
  + Chuyển đổi văn bản thành chữ thường.
  + Loại bỏ các ký tự nằm trong dấu ngoặc vuông (ví dụ: [abc]).
  + Loại bỏ các đường dẫn web (ví dụ: https://example.com).
  + Loại bỏ các thẻ HTML (ví dụ: <p>).
  + Loại bỏ các ký tự đặc biệt (ví dụ: dấu câu).
  + Loại bỏ ký tự xuống dòng.
  + Loại bỏ các từ kết hợp chữ và số (ví dụ: abc123).
  + Chuyển từ quá khứ về dạng gốc (ví dụ: went → go)

```
def clean_text(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(text)
    tokens = [lemmatizer.lemmatize(token, pos='v') for token in
tokens]
    text = ' '.join(tokens)
    return text
```

- Áp dụng hàm 'clean_text' lên cột 'Review Text' của DataFrame 'df' để làm sạch dữ liệu văn bản trong cột đó.

```
df['Review Text'] = df['Review Text'].apply(lambda x:clean_text(x))
```

- In thông tin về DataFrame 'df', bao gồm số lượng hàng và cột, kiểu dữ liệu của từng cột và thông tin về bộ nhớ sử dụng. Lưu DataFrame 'df' thành file CSV có tên 'data_NLP.csv' trong thư mục 'folder' được định nghĩa trước đó.

```
df.info()
df.to_csv(folder+"/data_NLP.csv", index=False)
```

- Định nghĩa một hàm 'roc_auc' để tính toán giá trị ROC AUC dựa trên các dự đoán và nhãn mục tiêu. Hàm này sử dụng thư viện 'metrics' từ sklearn để tính toán giá trị ROC AUC.

```
def roc_auc(predictions,target):

    fpr, tpr, thresholds = metrics.roc_curve(target, predictions)
    roc_auc = metrics.auc(fpr, tpr)
    return roc_auc
```

- Sau đó gán cột 'Review Text' vào biến 'X' và cột 'Recommended IND' vào biến 'y' để sử dụng cho việc huấn luyện mô hình. Ta sử dụng Tokenizer để mã hóa các câu thành chuỗi các số nguyên để chuẩn bị dữ liệu cho việc huấn luyện mô hình. Chia dữ liệu thành hai tập huấn luyện và kiểm tra (train-test split) theo tỷ lệ 80:20. Tham số stratify=y đảm bảo rằng tỷ lệ các lớp trong cột 'Recommended IND' được giữ nguyên trong cả hai tập. Chuyển đổi dữ liệu nhãn từ dạng Series thành mảng NumPy với kiểu dữ liệu np.int32.

```
X = df['Review Text']
y = df['Recommended IND']

tokenizer = Tokenizer()
tokenizer.fit_on_texts(X.tolist())

word_index = tokenizer.word_index

X_seq = tokenizer.texts_to_sequences(X.tolist())

X_pad = pad_sequences(X_seq)
```

- Tiếp theo, ta chia dữ liệu thành hai tập huấn luyện và kiểm tra (train-test split) theo tỷ lệ 80:20. Tham số stratify=y đảm bảo rằng tỷ lệ các lớp trong cột 'Recommended IND' được giữ nguyên trong cả hai tập. Chuyển đổi dữ liệu nhãn từ dạng Series thành mảng NumPy với kiểu dữ liệu np.int32.

```
X_train, X_test, y_train, y_test = train_test_split(X_pad, y,
                                                    stratify=y,
                                                    test_size=0.2,
                                                    random_state=42,
                                                    shuffle=True)
y_train = np.array(y_train, dtype=np.int32)
```

```
y_test  = np.array(y_test, dtype=np.int32)
```

- Cuối cùng, ta sử dụng Word2Vec để tạo ma trận nhúng embedding_matrix để dùng cho mô hình học sâu Long Short Term Memory (LSTM).

```
word2vec_model = Word2Vec(sentences=X_pad.tolist(), vector_size=100,
window=5, min_count=1, workers=4)
word2vec_model.init_sims(replace=True)
embedding_matrix = word2vec_model.wv.vectors
vocab_size, embedding_dim = embedding_matrix.shape
```

1.1. Mô hình Naive Bayes

Chúng ta thực hiện các bước để huấn luyện một mô hình Naive Bayes và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

```
NB = MultinomialNB()
NB.fit(X_train, y_train)
print("Training Completed")
```

```
predicted=NB.predict(X_test)
y_pred = NB.predict(X_test)
print("AUC: %.2f%%" % (roc_auc(y_pred,y_test)))
```

```
AUC: 0.52%
```

→ Giá trị AUC là 0.52% cho hiệu suất của mô hình không cao vì thế mô hình trên chưa tốt để có thể phân loại dữ liệu.

1.2. Mô hình Decision Tree

Chúng ta thực hiện các bước để huấn luyện một mô hình Decision Tree và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

```
DT = DecisionTreeClassifier()
DT.fit(X_train, y_train)
print("Training Completed")
```

```
y_pred = DT.predict(X_test)
print("AUC: %.2f%%" % (roc_auc(y_pred,y_test)))
```

```
AUC: 0.52%
```

→ Giá trị AUC là 0.52% cho hiệu suất của mô hình không cao vì thế mô hình trên chưa tốt để có thể phân loại dữ liệu.

1.3. Mô hình Log Regression

Chúng ta thực hiện các bước để huấn luyện một mô hình Logistic Regression và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

```
logistic_regression = LogisticRegression()
logistic_regression.fit(X_train, y_train)
print("Training Completed")
```

```
y_pred = logistic_regression.predict(X_test)
print("AUC: %.2f%%" % (roc_auc(y_pred,y_test)))
```

```
AUC: 0.50%
```

→ Giá trị AUC là 0.50% cho hiệu suất của mô hình không cao vì thế mô hình trên chưa tốt để có thể phân loại dữ liệu.

1.4. Mô hình LSTM

Chúng ta thực hiện các bước để huấn luyện một mô hình LSTM và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

Trước tiên nhóm xây dựng một mô hình mạng neural sử dụng kiến trúc Sequential:

```
model = Sequential()
model.add(Embedding(vocab_size, embedding_dim,
weights=[embedding_matrix], input_length=X_train.shape[1],
trainable=False))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.fit(X_train, y_train, validation_data=(X_test, y_test),
epochs=10, batch_size=32)
```

```
model.fit(X_train, y_train)
print("Training Completed")
```

```
y_pred = model.predict(X_test)
print("AUC: %.3f%%" % (roc_auc(y_pred,y_test)))
```

```
123/123 [==============================] - 4s 32ms/step
AUC: 0.913%
```

→ **Nhận xét:** Tổng quan, LSTM có AUC cao nhất, cho thấy mô hình có khả năng dự đoán tốt nhất. Naive Bayes và Logistic Regression có AUC tương đối gần nhau và tương đối tốt. Trong khi đó, Decision Tree có AUC thấp hơn, cho thấy mô hình có khả năng

dự đoán kém hơn. Tuy nhiên, cần xem xét các yếu tố khác nữa, cho nên nhóm quyết định đánh giá thêm bằng những chỉ số khác ở mục tiếp theo.

2. ĐÁNH GIÁ MÔ HÌNH

Để thống kê tỷ lệ lỗi, ta sẽ tiến hành tính các chỉ số như MSE, R-squared, MAE, percentage error bằng các câu lệnh sau:

```
# Đánh giá mô hình Naive Bayes
naive_bayes_predictions = NB.predict(X_test)
naive_bayes_mse = mean_squared_error(y_test, naive_bayes_predictions)
naive_bayes_mae = mean_absolute_error(y_test, naive_bayes_predictions)
naive_bayes_r2 = r2_score(y_test, naive_bayes_predictions)
naive_bayes_percentage_error = np.mean(np.abs((y_test -
naive_bayes_predictions) / y_test)) * 100

print("Naive Bayes:")
print("MSE:", naive_bayes_mse)
print("MAE:", naive_bayes_mae)
print("R-square:", naive_bayes_r2)

# Đánh giá mô hình Logistic Regression
logistic_regression_predictions = logistic_regression.predict(X_test)
logistic_regression_mse = mean_squared_error(y_test,
logistic_regression_predictions)
logistic_regression_mae = mean_absolute_error(y_test,
logistic_regression_predictions)
logistic_regression_r2 = r2_score(y_test,
logistic_regression_predictions)
logistic_regression_percentage_error = np.mean(np.abs((y_test -
logistic_regression_predictions) / y_test)) * 100

print("\nLogistic Regression:")
print("MSE:", logistic_regression_mse)
print("MAE:", logistic_regression_mae)
print("R-square:", logistic_regression_r2)

# Đánh giá mô hình Decision Tree
decision_tree_predictions = DT.predict(X_test)
decision_tree_mse = mean_squared_error(y_test,
decision_tree_predictions)
decision_tree_mae = mean_absolute_error(y_test,
decision_tree_predictions)
decision_tree_r2 = r2_score(y_test, decision_tree_predictions)
decision_tree_percentage_error = np.mean(np.abs((y_test -
decision_tree_predictions) / y_test)) * 100

print("\nDecision Tree:")
print("MSE:", decision_tree_mse)
print("MAE:", decision_tree_mae)
print("R-square:", decision_tree_r2)

# Đánh giá mô hình LSTM
lstm_predictions = model.predict(X_test)
lstm_mse = mean_squared_error(y_test, lstm_predictions)
```

```
lstm_mae = mean_absolute_error(y_test, lstm_predictions)
lstm_r2 = r2_score(y_test, lstm_predictions)
lstm_percentage_error = np.mean(np.abs((y_test - lstm_predictions) /
y_test)) * 100

print("\nLSTM:")
print("MSE:", lstm_mse)
print("MAE:", lstm_mae)
print("R-square:", lstm_r2)
```

Kết quả thu được:

```
Naive Bayes:
MSE: 0.547165013984236
MAE: 0.547165013984236
R-square: -2.6785285565894634

Logistic Regression:
MSE: 0.18281210272056955
MAE: 0.18281210272056955
R-square: -0.22902510789397046

Decision Tree:
MSE: 0.2957030256801424
MAE: 0.2957030256801424
R-square: -0.9879780257033206
123/123 [==============================] - 2s 15ms/step

LSTM:
MSE: 0.08811651278422267
MAE: 0.16404286198407514
R-square: 0.40760264216530817
```

Nhận xét:

- Đầu tiên, chúng ta đã áp dụng **mô hình Naive Bayes** bằng cách sử dụng hàm predict() của nó trên tập dữ liệu X_test. Sau đó, chúng ta tính toán MSE bằng cách so sánh các giá trị dự đoán của mô hình với giá trị thực tế, và tính toán MAE bằng cách tính trung bình giá trị tuyệt đối của sự sai khác giữa các giá trị dự đoán và giá trị thực tế. Cuối cùng, chúng ta tính toán R-square, một độ đo phổ biến được sử dụng để đánh giá khả năng giải thích của mô hình. Giá trị R-square nằm trong khoảng từ -∞ đến 1, với 1 cho thấy mô hình phù hợp hoàn hảo với dữ liệu và 0 cho thấy mô hình không có khả năng giải thích hơn so với trung bình.
    - MSE (Mean Squared Error): 0.547
    - MAE (Mean Absolute Error): 0.547
    - R-square: -2.624

- Kết quả này cho thấy mô hình Naive Bayes không phù hợp với dữ liệu hoặc có thể có những vấn đề trong việc đánh giá và dự đoán dữ liệu. Giá trị MSE và MAE lớn, R-square âm chỉ ra rằng mô hình không tốt trong việc giải thích biến phụ thuộc.

- Tiếp theo, chúng ta đã áp dụng **mô hình Logistic Regression** bằng cách sử dụng hàm predict() của nó trên tập dữ liệu X_test. Tương tự như mô hình Naive Bayes, chúng ta tính toán MSE, MAE và R-square cho mô hình Logistic Regression để đánh giá hiệu suất của nó.
    - MSE: 0.182
    - MAE: 0.182
    - R-square: -0.229
    - Mô hình Logistic Regression cho kết quả tốt hơn so với Naive Bayes, với giá trị MSE, MAE thấp hơn và R-square gần 0. Tuy nhiên, R-square âm chỉ ra rằng mô hình không thể giải thích tốt biến phụ thuộc.

- Tiếp theo, chúng ta đã sử dụng **mô hình Decision Tree** bằng cách sử dụng hàm predict() của nó trên tập dữ liệu X_test. Chúng ta tính toán MSE, MAE và R-square để đánh giá mức độ chính xác và khả năng giải thích của mô hình Decision Tree.
    - MSE: 0.295
    - MAE: 0.295
    - R-square: -0.987
    - Mô hình Decision Tree có kết quả tương đối kém, với giá trị MSE, MAE cao và R-square âm. Điều này cho thấy mô hình không tốt trong việc dự đoán và giải thích biến phụ thuộc.

- Cuối cùng, chúng ta đã áp dụng **mô hình LSTM** bằng cách sử dụng hàm predict() của nó trên tập dữ liệu X_test. Chúng ta tính toán MSE, MAE và R-square để đánh giá hiệu suất của mô hình LSTM.
    - MSE: 0.088
    - MAE: 0.164

- R-square: 0.407
- Kết quả MSE và MAE của LSTM thấp nhất trong số các mô hình đã đánh giá, cho thấy mô hình có độ chính xác cao hơn. R-square đạt khoảng 0.428, cho thấy mô hình có khả năng giải thích một phần lớn biến động của dữ liệu.

→ Tổng quan, LSTM cho kết quả tốt nhất trong số các mô hình đã đánh giá, với MSE và MAE thấp nhất và R-square cao nhất. Do đó, nhóm quyết định chọn mô hình LSTM là mô hình để dự đoán. Biết rằng:

- MSE là phép đo độ lệch giữa giá trị dự đoán và giá trị thực tế bằng cách lấy trung bình của bình phương sự chênh lệch của chúng, chỉ số này càng thấp thì càng tốt.
- MAE là trung bình giá trị tuyệt đối của chênh lệch giữa giá trị dự đoán và giá trị thực tế, cũng tương tự như MSE khi chỉ số càng thấp, mô hình càng đưa ra kết quả dự đoán chính xác.
- R-squared thể hiện phần trăm phương sai của biến phụ thuộc hay nói cách khác nó thể hiện mức độ phù hợp của dữ liệu với mô hình hồi quy, khi R-squared càng gần 1, mô hình càng tốt.

CHƯƠNG 5: GIAO DIỆN CHƯƠNG TRÌNH

1. GIỚI THIỆU GIAO DIỆN

Dựa vào mô hình có kết quả dự đoán tốt nhất là Long Short Term Memory (LSTM), nhóm đã thiết kế một giao diện đơn giản để áp dụng mô hình này vào dự đoán sự mua lại (Recommended IND) dựa trên nội dung đánh giá sản phẩm (Review Text) của khánh hàng.

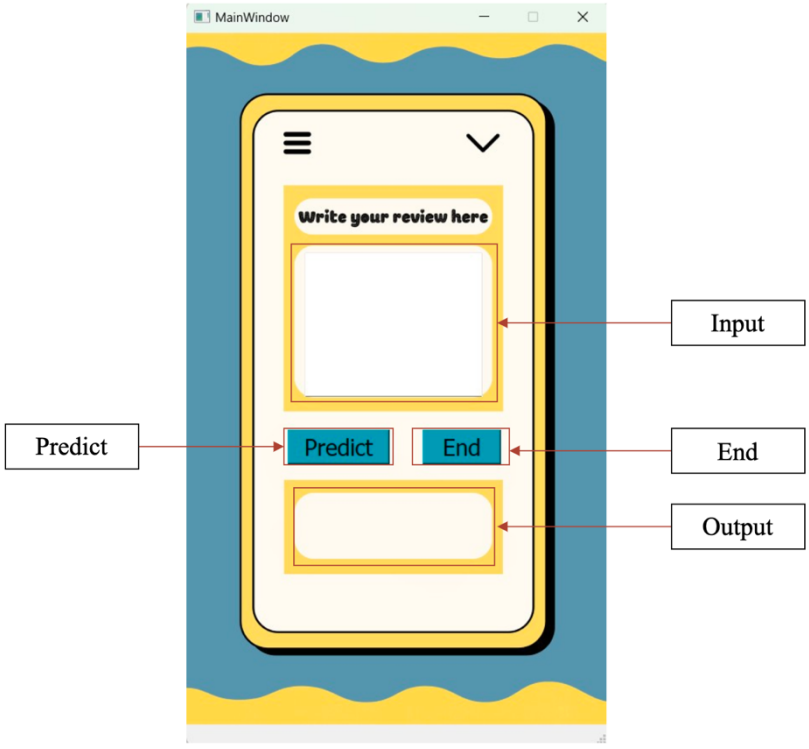
Giao diện gồm có 3 phần chính:

Input: Khu vực để nhập văn bản đánh giá về sản phẩm của khách hàng để thực hiện dự đoán về sự mua lại.

Output: Khu vực hiển thị kết quả dự đoán sự mua lại của khách hàng dựa trên văn bản đánh giá sản phẩm (Recommended: Có mua lại, No recommended: Không mua lại).

Buttons: Gồm có 2 buttons để tương tác với giao diện.

- Button "Predict": Button để thực hiện dự đoán sự mua lại để đưa kết quả ra khu vực Output sau khi đã có văn bản đánh giá ở khu vực Input. Nếu Input không chứa văn bản đánh giá thì Output sẽ hiển thị "Please enter a review"
- Button "End": Button để thực hiện kết thúc giao diện.



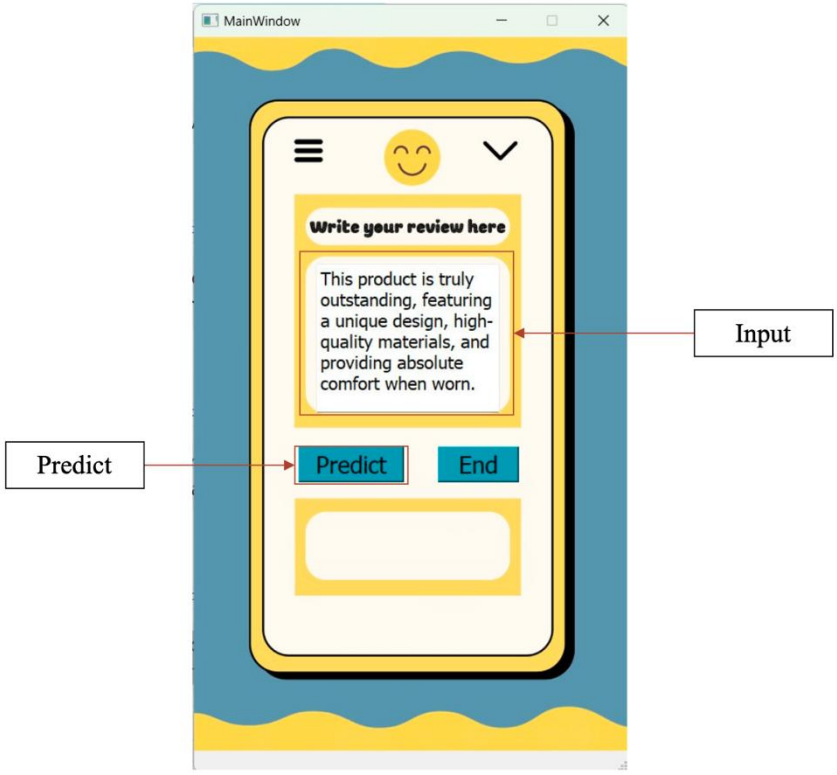
Hình 7: Giao diện chính của chương trình

2. HƯỚNG DẪN SỬ DỤNG GIAO DIỆN

Bước 1: Nhập đánh giá vào ô Input

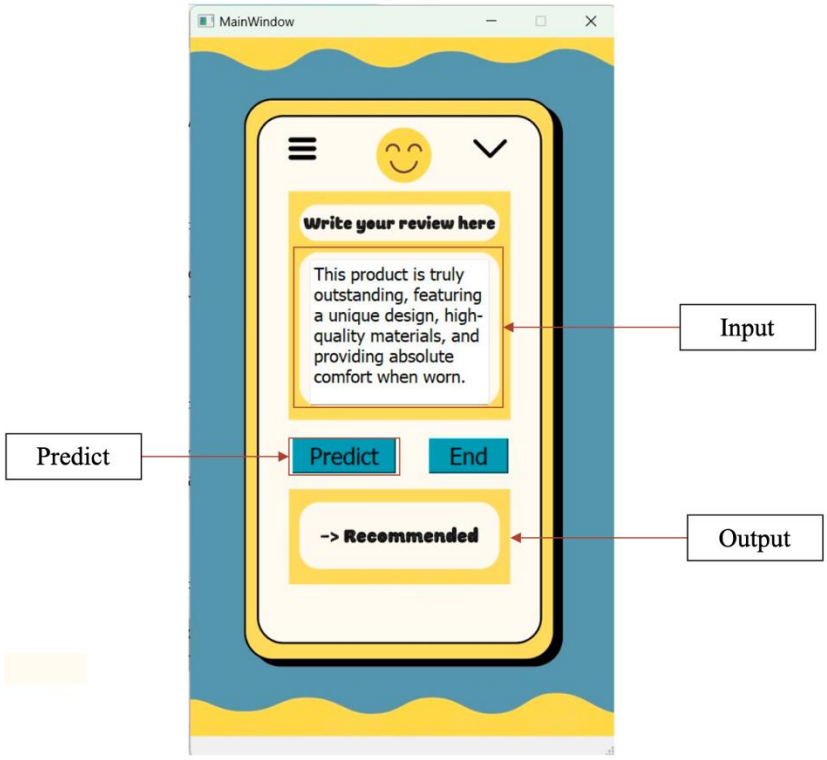
Hình 8: Nhập Input trong giao diện

Bước 2: Chọn Button "Predict" để dự đoán



Hình 9: Chọn "Predict" để dự đoán

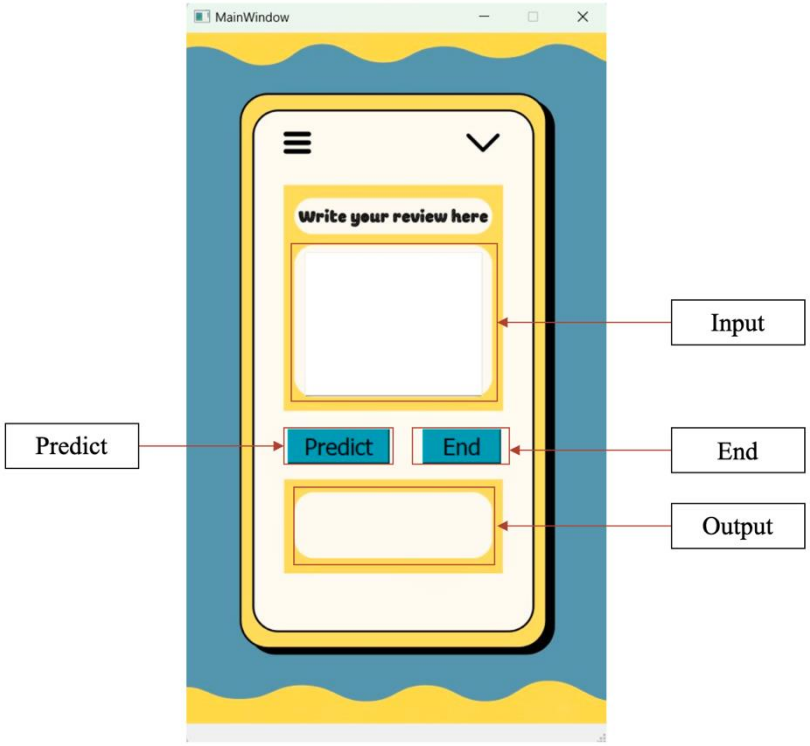
Bước 3: Kết quả được trả ra tại ô Output



Hình 10: Kết quả giao diện trả về

Để tiếp tục dự đoán 1 đánh giá khác ta tiến hành làm lại từ đầu bước 1 với đánh giá tiếp theo, nếu muốn kết thúc ta tiếp tục đến bước 4.

Bước 4: Kết thúc quá trình dự đoán ta chọn Button “End”.

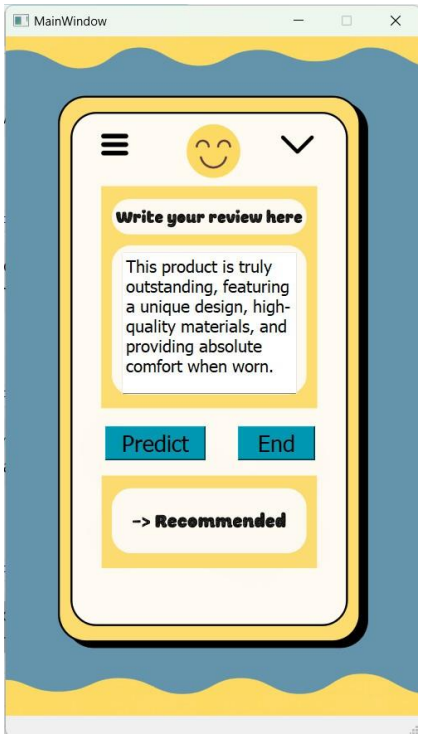


Hình 11: Kết thúc quá trình dự đoán

3. CÁC KẾT QUẢ GIAO DIỆN TRẢ VỀ

3.1. Đối với kết quả “Recommended” – đề xuất nên mua hàng

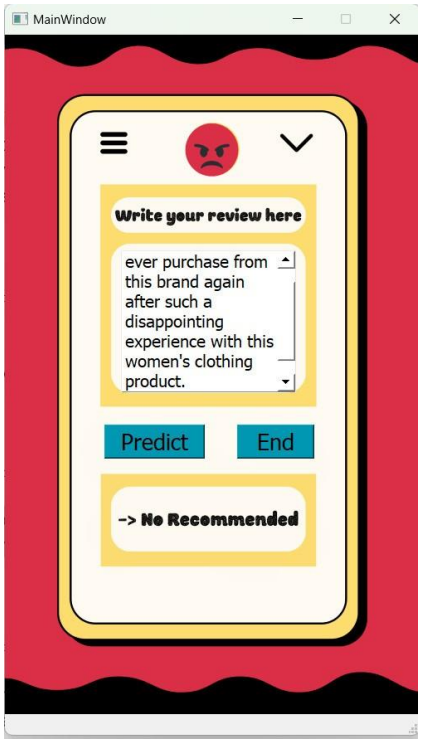
Đối với kết quả Recommended đồng nghĩa với việc sản phẩm này nhận được khuyến khích nên mua hàng đối với các khách hàng mới và khuyến khích mua lại sản phẩm đối với các khách hàng cũ.



Hình 12: Giao diện chương trình khi "Recommended"

3.2. Đối với “No Recommended” – không đề xuất nên mua hàng

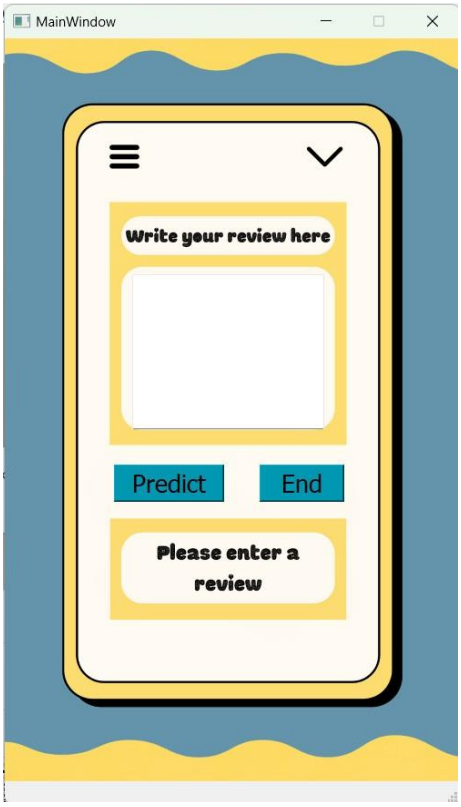
Đối với kết quả No Recommended đồng nghĩa với việc sản phẩm này không nhận được khuyến khích nên mua hàng đối với các khách hàng mới và cũng không khuyến khích mua lại sản phẩm đối với các khách hàng cũ.



Hình 13: Giao diện chương trình khi "No Recommeded"

3.3. Đối với người dùng chưa nhập đánh giá nhưng đã ấn "Predict"

Chương trình sẽ trả về kết quả "Please enter a review" đối với những người dùng muốn dự đoán mà chưa nhập đánh giá vào ô Input để lưu ý với khách hàng hãy nhập đánh giá.



Hình 14: Lưu ý khi người dùng không nhập đánh giá

CHƯƠNG 6: HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN

1. HẠN CHẾ

Hạn chế của đề tài Dự đoán đề xuất sản phẩm từ đánh giá khách hàng về quần áo nữ trên trang thương mại điện tử gồm các yếu tố sau đây:

- ***Dữ liệu không đồng đều:*** Dữ liệu đánh giá sản phẩm thời trang trực tuyến thường được thu thập từ nhiều nguồn khác nhau, bao gồm các trang web thương mại điện tử, mạng xã hội, và các diễn đàn trực tuyến. Điều này dẫn đến sự không đồng đều về chất lượng và tính toàn diện của dữ liệu. Sự không đồng đều của dữ liệu có thể làm giảm tính khách quan của kết quả đánh giá tổng thể. Điều này đặt ra thách thức trong việc tạo ra một bức tranh toàn diện về chất lượng sản phẩm trên thị trường.
- ***Khả năng tin cậy và chất lượng của người đánh giá:***
    + Người đánh giá sản phẩm thời trang trực tuyến có thể bao gồm cả khách hàng đã mua sản phẩm và các nhân viên của công ty bán sản phẩm. Điều này đặt ra câu hỏi về khả năng tin cậy và chất lượng của người đánh giá.

+ Khách hàng thường đánh giá sản phẩm dựa trên trải nghiệm cá nhân của họ. Tuy nhiên, trải nghiệm này có thể bị ảnh hưởng bởi nhiều yếu tố, chẳng hạn như sở thích cá nhân, kích thước cơ thể, hoặc hoàn cảnh sử dụng. Do đó, đánh giá của khách hàng có thể không khách quan và không phản ánh đúng chất lượng thực tế của sản phẩm.
+ Nhân viên của công ty bán sản phẩm có thể có động cơ thiên vị trong đánh giá. Họ có thể đánh giá cao sản phẩm của công ty mình để tăng doanh số bán hàng hoặc để tạo dựng lòng tin của khách hàng.
+ Khả năng tin cậy và chất lượng của người đánh giá có thể làm ảnh hưởng đến độ chính xác của dữ liệu. Điều này đặt ra câu hỏi về sự đáng tin cậy của thông tin người mua có được từ đánh giá.

- ***Thông tin chi tiết về sản phẩm:***
    + Một số đánh giá sản phẩm thời trang trực tuyến có thể không cung cấp đủ thông tin chi tiết về sản phẩm, chẳng hạn như kích thước, chất liệu, hoặc cách sản phẩm phối hợp với các mặt hàng khác. Điều này có thể làm mất đi tính khách quan và khả năng so sánh giữa các sản phẩm.
    + Ví dụ, một đánh giá chỉ nói rằng sản phẩm "rất đẹp" hoặc "rất thoải mái" không cung cấp đủ thông tin cho người mua để đưa ra quyết định mua hàng. Người mua cần biết thêm thông tin về kích thước, chất liệu, hoặc cách sản phẩm phối hợp với các mặt hàng khác để có thể so sánh sản phẩm với các sản phẩm khác và lựa chọn sản phẩm phù hợp với nhu cầu của mình.
    + Thiếu thông tin chi tiết về sản phẩm có thể làm giảm tính khách quan của đánh giá. Điều này đặt ra thách thức trong việc hiểu đúng nhu cầu và sở thích của người mua.

2. HƯỚNG PHÁT TRIỂN

- Để đảm bảo tính nhất quán và chất lượng của dữ liệu đánh giá sản phẩm thời trang trực tuyến, việc thống nhất các tiêu chí đánh giá là vô cùng quan trọng. Những tiêu chí này nên bao gồm những yếu tố quan trọng mà người mua quan tâm, như chất lượng sản phẩm, phong cách thiết kế, mức giá và độ bền.

- Đối mặt với thách thức của việc đảm bảo sự đáng tin cậy của dữ liệu, việc tăng cường kiểm duyệt là không thể phủ nhận. Không phải tất cả các đánh giá đều đáng tin cậy và chất lượng cao, điều này có thể dẫn đến những đánh giá thiên vị hoặc không chính xác, làm suy giảm tính khách quan của thông tin. Điều này đặt ra thách thức về độ chính xác và đáng tin cậy của thông tin mà người mua thu nhận từ các đánh giá trực tuyến.
- Để nâng cao chất lượng thông tin trong đánh giá, quy trình thu thập đánh giá có thể được tối ưu hóa. Khuyến khích người đánh giá cung cấp thông tin chi tiết hơn về sản phẩm, đặc biệt là về kích thước, chất liệu và cách sản phẩm phối hợp với các mặt hàng khác. Điều này giúp người mua có cái nhìn chi tiết và đầy đủ hơn về sản phẩm mà họ quan tâm, từ đó tối ưu hóa trải nghiệm mua sắm trực tuyến của họ.

PHỤ LỤC

Link github source code: Tại đây.

TÀI LIỆU THAM KHẢO

Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Mitchell, T. M. (1997). Machine learning. McGraw-Hill Education.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. School of Engineering and Applied Science, Washington University in St. Louis, 1 Brookings Drive, St. Louis, MO, 63130-4899, USA.

Mạnh, P. V. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow. Đại học Khoa học Tự nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh.

Brownlee, J. (2017). Deep Learning for Natural Language Processing. Machine Learning Mastery.

Jadhav, S. D., & Channe, H. P. (2016). Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. International Journal of Science and Research (IJSR).

BẢNG PHÂN CÔNG

STT	Họ và tên	MSSV	Công việc
1	Dương Mỹ Quỳnh	31211027666	- Chương 2 (2): Phân tích khám phá. - Chương 2 (3): Nhận diện các biến. - Chương 2 (4): Thống kê dữ liệu. - Chương 3: Tiền xử lý dữ liệu. - Chương 4: Mô hình dự đoán - Làm slide thuyết trình.
2	Trương Vũ Phương Quỳnh	31211027668	- Chương 1: Tổng quan về đề tài. - Chương 2 (1): Mô tả bộ dữ liệu. - Chương 5: Trình bày word giao diện chương trình. - Chương 6: Hạn chế và hướng phát triển. - Tổng hợp word.
3	Đinh Công Thành	31211027670	- Chương 4: Mô hình dự đoán. - Chương 5: Trình bày word giao diện chương trình. - Giao diện chương trình.