BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

BÀI TẬP LỚN
HỌC MÁY CƠ BẢN

TÌM HIỂU MÔ HÌNH LOGISTIC REGRESSION, RANDOM FOREST, KNN VÀ SVM ĐỂ DỰ ĐOÁN ĐÁNH GIÁ SẢN PHẨM

NGÀNH: **CÔNG NGHỆ THÔNG TIN**

CHUYÊN NGÀNH: TRÍ TUỆ NHÂN TẠO

SINH VIÊN: **LÊ ĐỨC THẮNG**

MÃ SINH VIÊN: 12423032

MÃ LỚP: **124231**

HƯỚNG DẪN: **PGS. TS. NGUYỄN VĂN HẬU**

HƯNG YÊN – 2026

LỜI CAM ĐOAN

Em xin cam đoan bài tập lớn "Tìm hiểu mô hình logistic regression, random forest, KNN và SVM để dự đoán đánh giá sản phẩm" là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Nguyễn Văn Hậu và thầy Trần Tuấn Anh.

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong đồ án và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

Hưng Yên, ngày tháng năm 2026

Sinh viên

LỜI CẢM ƠN

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Khoa học máy tính, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn Thầy Nguyễn Văn Hậu đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các thầy/cô trong trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các thầy/cô về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

MỤC LỤC

MỤC LỤC HÌNH ẢNH

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Đặt vấn đề

Trong kỷ nguyên số hóa, Thương mại điện tử (E-commerce) đã trở thành một phần không thể thiếu của nền kinh tế toàn cầu. Các nền tảng mua sắm trực tuyến không chỉ là nơi giao dịch hàng hóa mà còn là không gian tương tác xã hội, nơi người mua để lại những đánh giá (Reviews) về chất lượng sản phẩm và dịch vụ.

Đối với các doanh nghiệp, những đánh giá này là nguồn dữ liệu vô giá. Một sản phẩm nhận được nhiều đánh giá tích cực sẽ gia tăng uy tín và thúc đẩy doanh số bán hàng (Conversion Rate). Ngược lại, các đánh giá tiêu cực giúp doanh nghiệp nhận diện khuyết điểm để cải thiện. Tuy nhiên, thách thức đặt ra là sự bùng nổ về số lượng dữ liệu. Các sàn thương mại điện tử lớn hàng ngày tiếp nhận hàng triệu lượt bình luận. Việc sử dụng nhân sự để đọc thủ công, phân loại và tổng hợp ý kiến khách hàng là phương án không khả thi về mặt chi phí và thời gian, đồng thời dễ mắc sai sót do yếu tố chủ quan của con người.

Xuất phát từ thực tiễn đó, việc ứng dụng Trí tuệ nhân tạo (AI) và Học máy (Machine Learning) để tự động hóa quy trình phân tích phản hồi khách hàng là yêu cầu cấp thiết. Đề tài "Xây dựng hệ thống dự đoán đánh giá sản phẩm" được thực hiện nhằm giải quyết bài toán này, tập trung vào việc dự đoán thái độ của khách hàng (Hài lòng hoặc Không hài lòng) dựa trên dữ liệu văn bản và thông tin hành vi mua sắm.

1.2. Mục tiêu

Đề tài hướng tới việc xây dựng một quy trình khép kín từ xử lý dữ liệu thô đến triển khai ứng dụng thực tế, với các mục tiêu cụ thể sau:

- **Mục tiêu về dữ liệu:** Thu thập, làm sạch và chuẩn hóa bộ dữ liệu đánh giá sản phẩm thời trang (Women's Clothing E-Commerce Reviews), giải quyết các vấn đề về nhiễu ngôn ngữ và dữ liệu thiếu (Missing values).

- **Mục tiêu về thuật toán:** Nghiên cứu và áp dụng các mô hình học máy phân lớp (Classification) như Logistic Regression, Random Forest, SVM và KNN. So sánh hiệu quả giữa các mô hình để chọn ra thuật toán tối ưu nhất cho bài toán phân tích cảm xúc.

- **Mục tiêu về ứng dụng:** Xây dựng một ứng dụng Demo trên nền tảng Web (sử dụng Streamlit Framework) cho phép nhập liệu thời gian thực và trả về kết quả dự đoán, đồng thời giải quyết các thách thức kỹ thuật khi đưa mô hình từ môi trường huấn luyện sang môi trường thực tế (Deployment).

1.3. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Các thuật toán Xử lý ngôn ngữ tự nhiên (NLP) áp dụng cho tiếng Anh và các mô hình học máy phân lớp nhị phân.

- Phạm vi dữ liệu: Dữ liệu được giới hạn lĩnh vực thời gian nữ, bao gồm các thông tin: Nội dung bình luận (Review Text), Tuổi khách hàng (Age), Điểm đánh giá (Rating) và các thông tin phân loại sản phẩm (Division, Department)

- Phạm vi kỹ thuật: Tập trung vào các phương pháp học máy truyền thống (Traditional Machine Learning), chưa đi sâu vào các mô hình học sâu (Deep Learning) phức tạp như BERT hay LSTM

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về Xử lý ngôn ngữ tự nhiên (NLP)

2.1.1. Khái niệm

Xử lý ngôn ngữ tự nhiên (NLP) là một nhánh giao thoa giữa Trí tuệ nhân tạo (AI), Khoa học máy tính và Ngôn ngữ học. Mục tiêu cốt lõi của NLP là giúp máy tính có khả năng "đọc - hiểu" và diễn giải ngôn ngữ của con người một cách tự nhiên nhất.

Trong bài toán Dự đoán đánh giá sản phẩm, chúng ta đang giải quyết một bài toán con của NLP gọi là Phân tích cảm xúc (Sentiment Analysis) hay còn gọi là Khai phá ý kiến (Opinion Mining). Đây là quá trình sử dụng các kỹ thuật tính toán để xác định thái độ, ý kiến và cảm xúc của người viết đối với một chủ đề cụ thể (ở đây là sản phẩm thời trang).

2.1.2. Các cấp độ phân tích

Hệ thống phân tích cảm xúc thường hoạt động ở ba cấp độ:

- **Cấp độ văn bản (Document level):** Phân loại cảm xúc của toàn bộ đoạn văn (Ví dụ: Cả bài review này là khen hay chê?). Đây là cấp độ mà đề tài này tập trung giải quyết.
- **Cấp độ câu (Sentence level):** Phân tích từng câu riêng lẻ.
- **Cấp độ khía cạnh (Aspect level):** Phân tích cảm xúc đối với từng thuộc tính của sản phẩm (Ví dụ: "Vải đẹp" - Tích cực, nhưng "Giá đắt" - Tiêu cực).

2.2. Các kỹ thuật tiền xử lý dữ liệu văn bản (Text Preprocessing)

Dữ liệu văn bản thô (Raw text) thường chứa rất nhiều "nhiễu" (noise) như dấu câu, ký tự lạ, viết tắt, gây khó khăn cho mô hình học máy. Quá trình tiền xử lý đóng vai trò quyết định đến độ chính xác của mô hình.

2.2.1. Làm sạch dữ liệu (Data Cleaning)

Quá trình này sử dụng các biểu thức chính quy (Regular Expressions - Regex) để loại bỏ các thành phần không mang ý nghĩa ngữ nghĩa:

- Loại bỏ thẻ HTML (nếu dữ liệu lấy từ web).
- Loại bỏ các đường dẫn URL (http://...).
- Loại bỏ các ký tự đặc biệt và dấu câu (!, @, #, ?, .).
- Chuyển đổi chữ thường (Lowercasing): Máy tính coi "Good" và "good" là hai từ khác nhau (do mã ASCII khác nhau). Việc chuyển toàn bộ về chữ thường giúp đồng nhất dữ liệu.

2.2.2. Loại bỏ hư từ (Stopwords Removal)

Trong tiếng Anh, có những từ xuất hiện với tần suất rất cao nhưng mang ít ý nghĩa phân loại cảm xúc, được gọi là **Stopwords**.

- Ví dụ: *"the", "is", "at", "which", "on"*.
- Lý do loại bỏ: Theo định luật Zipf, các từ này chiếm phần lớn dung lượng văn bản nhưng lại chứa lượng thông tin (Information Content) thấp nhất. Loại bỏ chúng giúp giảm chiều dữ liệu (Dimensionality Reduction) và giúp mô hình tập trung vào các từ khóa quan trọng (như *"beautiful", "bad", "terrible"*).

2.2.3. Mã hóa văn bản (Text Vectorization)

Máy tính không thể tính toán trên chuỗi ký tự, do đó văn bản cần được chuyển đổi thành các vector số học. Đề tài sử dụng phương pháp **TF-IDF.**

TTF-IDF (Term Frequency - Inverse Document Frequency) là kỹ thuật đánh giá tầm quan trọng của một từ trong văn bản.

- **TF (Term Frequency): Tần suất xuất hiện của từ t trong văn bản d.**

$$TF(t,d) = \frac{\text{Số lần từ } t \text{ xuất hiện trong } d}{\text{Tổng số từ trong } d}$$

- **IDF (Inverse Document Frequency):** Đánh giá mức độ "hiếm" của từ. Nếu một từ xuất hiện ở tất cả các văn bản (như "dress" trong tập dữ liệu quần áo), nó ít có khả năng phân loại.

$$IDF(t) = \log\left(\frac{N}{1 + DF(t)}\right)$$

Trong đó:

  - N: Tổng số văn bản trong bộ dữ liệu.
  - DF(t): Số lượng văn bản chứa từ t.

- **Công thức TF-IDF:**

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

⇨ Từ nào có TF cao (xuất hiện nhiều trong câu đó) và IDF cao (hiếm gặp trong cả bộ dữ liệu) sẽ có trọng số lớn nhất.

2.3. Kỹ thuật xử lý mất cân bằng dữ liệu: SMOTE

Một thách thức lớn trong đề tài là sự chênh lệch giữa số lượng đánh giá Tích cực và Tiêu cực. Để giải quyết, nhóm áp dụng **SMOTE (Synthetic Minority Over-sampling Technique).**

- **Cơ chế hoạt động:** Thay vì chỉ sao chép lặp lại các điểm dữ liệu cũ (gây ra hiện tượng Overfitting), SMOTE tạo ra các điểm dữ liệu "nhân tạo" mới.
- **Thuật toán:**
    1. Chọn một điểm dữ liệu A thuộc lớp thiểu số (Tiêu cực).
    2. Tìm k điểm lân cận gần nhất (k-Nearest Neighbors) của A.
    3. Chọn ngẫu nhiên một điểm lân cận B.
    4. Tạo ra điểm mới C nằm trên đoạn thẳng nối A và B trong không gian vector.

$$C = A + rand(0,1) \times (B - A)$$

2.4. Các mô hình Học máy

2.4.1. Logistic Regression (Hồi quy Logistic)

Đây là thuật toán được lựa chọn để triển khai ứng dụng Demo vì tính đơn giản và hiệu quả cao trong bài toán phân loại nhị phân.

- **Nguyên lý:** Logistic Regression không dự đoán trực tiếp nhãn 0 hay 1, mà dự đoán xác suất điểm dữ liệu thuộc về lớp 1 (P(y=1|x)).
- **Hàm kích hoạt Sigmoid:** Để đảm bảo xác suất luôn nằm trong khoảng (0, 1), thuật toán sử dụng hàm Sigmoid:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Trong đó (tổ hợp tuyến tính của các đặc trưng đầu vào).

- **Quyết định:**
    - Nếu sigma(z) >= 0.5 => Dự đoán: Tích cực (Class 1).
    - Nếu sigma(z) < 0.5 => Dự đoán: Tiêu cực (Class 0).

2.4.2. Random Forest (Rừng ngẫu nhiên)

Random Forest là một thuật toán học máy tổ hợp (Ensemble Learning), hoạt động dựa trên nguyên tắc "Trí tuệ của đám đông" (Wisdom of Crowds).

- **Cấu trúc:** Bao gồm rất nhiều Cây quyết định (Decision Trees) hoạt động song song.
- **Kỹ thuật Bagging (Bootstrap Aggregating):** Mỗi cây con được huấn luyện trên một tập con ngẫu nhiên của dữ liệu gốc.
- **Cơ chế dự đoán:** Khi có dữ liệu mới, mỗi cây sẽ đưa ra một dự đoán riêng. Kết quả cuối cùng được quyết định bằng cách **bỏ phiếu đa số (Majority Voting).**
- **Ưu điểm:** Giảm thiểu hiện tượng quá khớp (Overfitting) thường gặp ở cây quyết định đơn lẻ và cho độ chính xác rất cao trên dữ liệu dạng bảng.

2.4.3. Support Vector Machine (SVM)

SVM là một thuật toán học máy giám sát mạnh mẽ, thường được sử dụng cho các bài toán phân loại văn bản nhờ khả năng làm việc tốt trên không gian dữ liệu nhiều chiều.

- **Nguyên lý hoạt động:** Mục tiêu của SVM là tìm ra một siêu phẳng (Hyperplane) trong không gian N-chiều (với N là số lượng đặc trưng) để phân tách các điểm dữ liệu của hai lớp (Tích cực và Tiêu cực) một cách rõ ràng nhất.
- **Lề tối đa (Maximum Margin):** Khác với các thuật toán phân loại khác, SVM không chỉ tìm đường phân chia mà còn tối ưu hóa để khoảng cách từ đường phân chia đến các điểm dữ liệu gần nhất (Support Vectors) là lớn nhất. Điều này giúp mô hình có khả năng tổng quát hóa tốt hơn, tránh hiện tượng quá khớp.
- **Kernel Trick:** Đối với dữ liệu không thể phân tách tuyến tính, SVM sử dụng các hàm nhân (Kernel functions) như RBF hoặc Polynomial để ánh xạ dữ liệu sang không gian chiều cao hơn, nơi chúng có thể được phân tách dễ dàng.

2.4.4. K-Nearest Neighbors (KNN)

KNN là thuật toán học máy dựa trên trường hợp (Instance-based learning), hay còn gọi là học lười (Lazy learning), vì nó không thực sự "học" một hàm giả định từ dữ liệu huấn luyện mà ghi nhớ toàn bộ dữ liệu.

- **Cơ chế:** Để dự đoán nhãn cho một điểm dữ liệu mới (một bình luận mới), KNN sẽ tìm kiếm K điểm dữ liệu gần nó nhất trong không gian vector (dựa trên khoảng cách Euclid hoặc Cosine).
- **Quyết định:** Nhãn của điểm dữ liệu mới được quyết định bằng cách "bỏ phiếu đa số" từ K láng giềng. Ví dụ: Nếu 3 trong 5 láng giềng là "Tích cực", thì bình luận mới cũng được phân loại là "Tích cực".
- **Ưu/Nhược điểm:** KNN rất đơn giản và hiệu quả với các bài toán nhỏ, nhưng tốn chi phí tính toán khi tập dữ liệu lớn vì phải tính khoảng cách tới tất cả các điểm.

2.5. Các độ đo đánh giá mô hình (Evaluation Metrics)

Để đánh giá khách quan hiệu năng của hệ thống, đề tài sử dụng các chỉ số sau:

1. **Ma trận nhầm lẫn (Confusion Matrix):** Bảng thể hiện sự tương quan giữa giá trị thực tế và dự đoán.
    - *True Positive (TP):* Thực tế là Tích cực, Máy đoán Tích cực.
    - *True Negative (TN):* Thực tế là Tiêu cực, Máy đoán Tiêu cực.
    - *False Positive (FP):* Thực tế là Tiêu cực, nhưng Máy đoán nhầm là Tích cực.
    - *False Negative (FN):* Thực tế là Tích cực, nhưng Máy đoán nhầm là Tiêu cực.
2. **Accuracy (Độ chính xác):** Tỷ lệ dự đoán đúng trên tổng số mẫu.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Lưu ý: Accuracy không phải là độ đo tốt khi dữ liệu bị mất cân bằng.

3. **Precision (Độ chuẩn xác):** Trong số các mẫu máy dự đoán là Tích cực, bao nhiêu phần trăm là đúng?

$$Precision = \frac{TP}{TP + FP}$$

4. **Recall (Độ phủ):** Máy phát hiện được bao nhiêu phần trăm mẫu Tích cực thực tế?

$$Recall = \frac{TP}{TP + FN}$$

5. **F1-Score:** Là trung bình điều hòa của Precision và Recall. Đây là chỉ số quan trọng nhất trong đề tài này để đảm bảo mô hình hoạt động tốt trên cả hai lớp dữ liệu.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

CHƯƠNG 3: QUY TRÌNH THỰC NGHIỆM VÀ XÂY DỰNG MÔ HÌNH

3.1. Mô tả bộ dữ liệu thực nghiệm

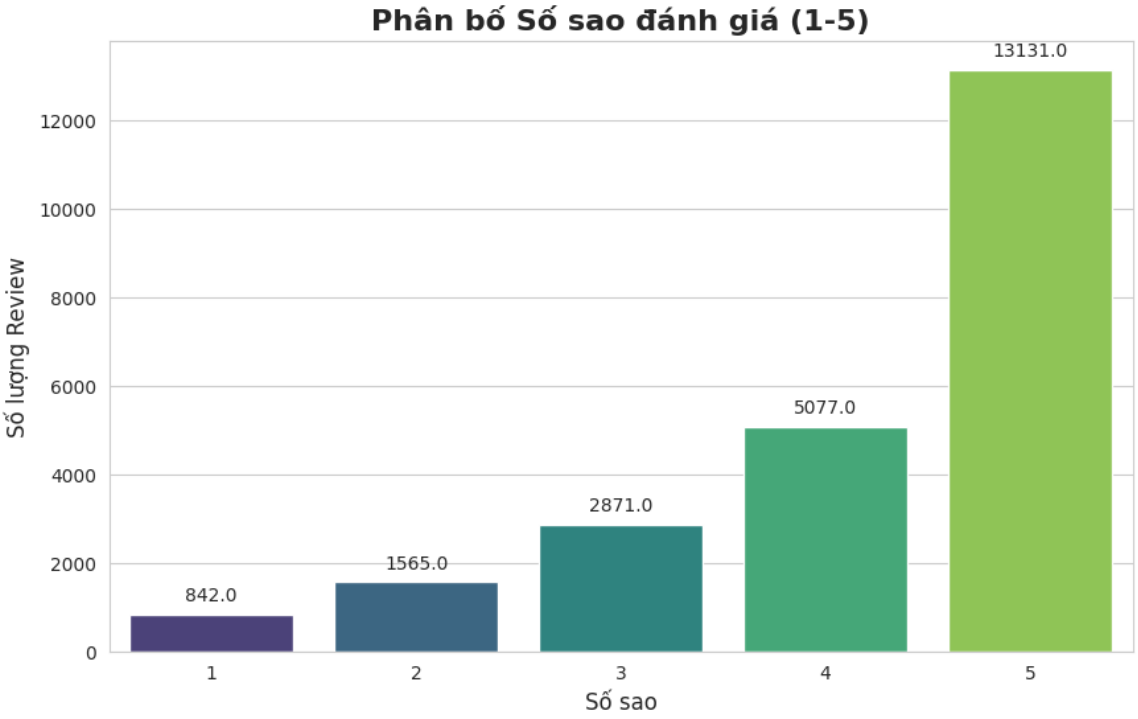
Bộ dữ liệu được sử dụng là **Women's Clothing E-Commerce Reviews**, bao gồm 23,486 bản ghi phản ánh hành vi mua sắm và đánh giá thực tế của khách hàng.

- **Nguồn dữ liệu:** Kaggle.
- **Kích thước:** 23,486 dòng và 10 cột.
- **Các đặc trưng chính (Features):**
    - Clothing ID: Mã định danh sản phẩm.
    - Age: Tuổi của khách hàng (Biến định lượng).
    - Title: Tiêu đề của bài đánh giá.
    - Review Text: Nội dung chi tiết bài đánh giá (Dữ liệu văn bản quan trọng nhất).
    - Rating: Điểm đánh giá từ 1 đến 5 sao (Biến định lượng).
    - Recommended IND: Biến mục tiêu (Target), 1 là Khuyên dùng, 0 là Không khuyên dùng.
    - Division/Department/Class Name: Các thông tin phân loại sản phẩm (Váy, Áo, Đồ lót...).

3.2. Phân tích khám phá dữ liệu (Exploratory Data Analysis - EDA)

Trước khi đưa vào mô hình, nhóm nghiên cứu tiến hành phân tích thống kê để hiểu rõ đặc điểm dữ liệu.

a. Phân bố biến mục tiêu (Rating & Recommended IND): Biểu đồ phân phối cho thấy sự mất cân bằng dữ liệu rõ rệt. Số lượng đánh giá 5 sao chiếm áp đảo (khoảng 55%), trong khi đánh giá 1-2 sao chỉ chiếm tỷ lệ nhỏ. Điều này đặt ra thách thức cho mô hình trong việc nhận diện các trường hợp tiêu cực (Negative).



Hình 1: Biểu đồ cột phân phối Rating

Chú thích: Biểu đồ thể hiện sự chênh lệch lớn giữa số lượng đánh giá tích cực và tiêu cực.

b. Phân tích từ khóa (Word Cloud): Sử dụng kỹ thuật Word Cloud để trực quan hóa các từ xuất hiện nhiều nhất:

- **Nhóm Hài lòng:** Các từ phổ biến là *love, fit, great, comfortable, soft*.
- **Nhóm Thất vọng:** Các từ phổ biến liên quan đến vấn đề kích thước và chất liệu như *small, huge, fabric, return, disappointment*.

Hình 2: Biểu đồ WordCloud của nhóm Positive

Hình 3: Biểu đồ WordCloud của nhóm Negative

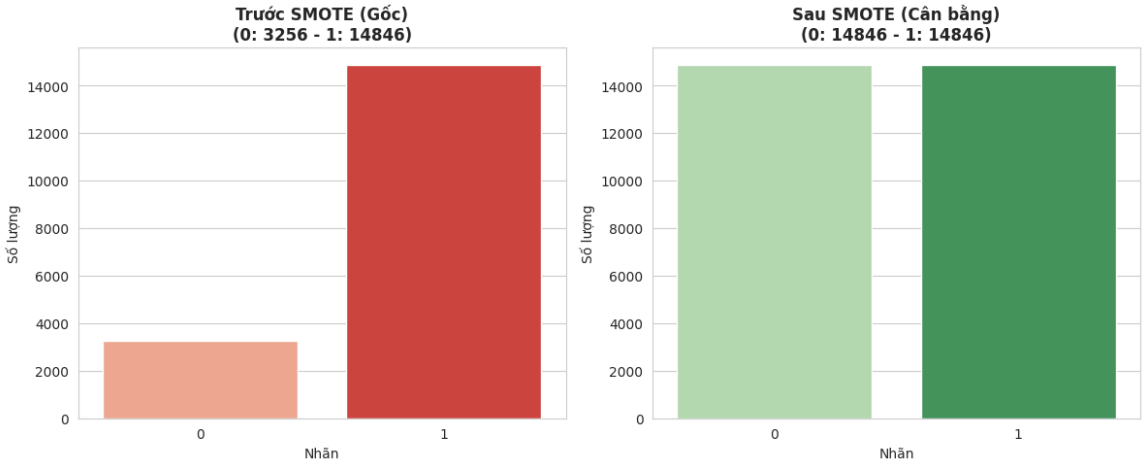
Chú thích: Các từ khóa đặc trưng cho hai nhóm cảm xúc đối lập.

3.3. Quy trình tiền xử lý dữ liệu (Data Preprocessing Pipeline)

Để máy tính có thể "hiểu" được dữ liệu hỗn hợp (Văn bản + Số), nhóm đã xây dựng quy trình xử lý gồm 4 bước:

1. **Làm sạch văn bản (Text Cleaning):**
    - Chuyển đổi toàn bộ văn bản về chữ thường (Lowercase).
    - Loại bỏ các ký tự đặc biệt, dấu câu, và các con số không mang ý nghĩa cảm xúc bằng biểu thức chính quy (Regex).
    - (Sơ đồ minh họa quá trình làm sạch text: Input -> Remove Punctuation -> Lowercase -> Output).
2. **Trích chọn đặc trưng văn bản (TF-IDF Vectorization):**
    - Sử dụng TfidfVectorizer với tham số max_features=2000. Điều này giúp giới hạn không gian đặc trưng ở 2000 từ quan trọng nhất, loại bỏ các từ quá hiếm gặp để giảm nhiễu và tăng tốc độ huấn luyện.
3. **Chuẩn hóa dữ liệu số (Numerical Scaling):**
    - Các biến Age (khoảng 18-99) và Rating (1-5) có biên độ khác nhau. Nhóm sử dụng StandardScaler để đưa chúng về phân phối chuẩn (Mean=0, Std=1), giúp các thuật toán như SVM và KNN hoạt động hiệu quả hơn.
4. **Xử lý mất cân bằng dữ liệu (SMOTE):**

- Áp dụng kỹ thuật sinh dữ liệu nhân tạo SMOTE (Synthetic Minority Over-sampling Technique) cho tập huấn luyện. Kỹ thuật này giúp cân bằng số lượng mẫu giữa hai lớp Tích cực và Tiêu cực, ngăn chặn mô hình bị thiên vị (Bias) về phía lớp đa số.



Hình 4: Biểu đồ trước và sau khi SMOTE

CHƯƠNG 4: CÀI ĐẶT VÀ HUẤN LUYỆN MÔ HÌNH

4.1. Thiết lập môi trường và Thông số thực nghiệm

Quá trình thực nghiệm được tiến hành trên nền tảng đám mây **Google Colab** (phiên bản Free Tier) để tận dụng tài nguyên RAM và CPU cho việc xử lý ma trận thưa (Sparse Matrix) kích thước lớn từ dữ liệu văn bản.

Cấu hình thư viện:

- **Scikit-learn (v1.2.x):** Thư viện lõi để xây dựng các pipeline học máy, từ tiền xử lý đến huấn luyện và đánh giá.
- **Imbalanced-learn:** Hỗ trợ kỹ thuật SMOTE để sinh dữ liệu nhân tạo, xử lý vấn đề mất cân bằng lớp.
- **Matplotlib & Seaborn:** Sử dụng để trực quan hóa dữ liệu (Heatmap, Bar chart) và vẽ đường cong ROC.
- **Joblib:** Sử dụng để tuần tự hóa (Serialize) mô hình, lưu trữ dưới dạng file nhị phân .pkl nhằm phục vụ cho quá trình triển khai (Deployment).

Chiến lược chia dữ liệu (Data Splitting):

Bộ dữ liệu sau khi tiền xử lý được chia thành 2 tập độc lập với tỷ lệ **80/20** và cố định random_state=42 để đảm bảo tính tái lập (Reproducibility) của kết quả:

- **Tập huấn luyện (Training Set - 80%):** Dùng để mô hình học các trọng số. Tập này được áp dụng SMOTE để cân bằng.
- **Tập kiểm thử (Test Set - 20%):** Dùng để đánh giá khách quan hiệu năng mô hình. Tập này được giữ nguyên bản (không SMOTE) để phản ánh đúng phân phối thực tế của dữ liệu.

4.2. Cấu hình chi tiết các thuật toán (Hyperparameter Tuning)

Nhóm đã lựa chọn 4 thuật toán đại diện cho các phương pháp học khác nhau: Tuyến tính (Linear), Phi tuyến (Non-linear), Dựa trên cây (Tree-based) và Dựa trên khoảng cách (Distance-based).

4.2.1. Logistic Regression (Hồi quy Logistic)

Mặc dù là mô hình đơn giản, Logistic Regression thường hoạt động cực kỳ hiệu quả trên dữ liệu văn bản có số chiều lớn (High-dimensional sparse data) như TF-IDF.

- **Bộ giải (Solver):** liblinear. Đây là thuật toán tối ưu hóa phù hợp nhất cho các bài toán phân loại nhị phân trên tập dữ liệu kích thước trung bình.
- **Điều chuẩn (Regularization):** Sử dụng chuẩn L2 (Ridge Regression) để phạt các trọng số quá lớn, giúp giảm thiểu hiện tượng quá khớp (Overfitting).

- **Trọng số lớp (Class Weight):** Thiết lập class_weight='balanced'. Tham số này tự động điều chỉnh mức phạt sai số tỷ lệ nghịch với tần suất xuất hiện của lớp dữ liệu (Lớp ít mẫu sẽ bị phạt nặng hơn nếu đoán sai).

4.2.2. Random Forest Classifier (Rừng ngẫu nhiên)

Đây là phương pháp học tổ hợp (Ensemble Learning) sử dụng kỹ thuật Bagging.

- **Số lượng cây (n_estimators):** 100 cây. Số lượng cây càng lớn, mô hình càng ổn định nhưng chi phí tính toán càng cao. Qua thử nghiệm, 100 là con số cân bằng giữa hiệu năng và tốc độ.
- **Tiêu chí phân chia (criterion):** gini. Chỉ số Gini Impurity giúp đo lường mức độ hỗn loạn của dữ liệu tại mỗi nút, từ đó chọn ra đặc trưng phân chia tốt nhất.
- **Độ sâu cây (max_depth):** Để None (không giới hạn) để các cây có thể phát triển tối đa, học được các mẫu phức tạp trong ngôn ngữ tự nhiên.

4.2.3. Support Vector Machine (SVM)

- **Kernel:** linear (Tuyến tính). Đối với bài toán phân loại văn bản với số lượng đặc trưng lớn (2000 từ), dữ liệu thường có xu hướng phân tách tuyến tính tốt. Kernel tuyến tính hội tụ nhanh hơn nhiều so với Kernel RBF hay Polynomial mà vẫn đảm bảo độ chính xác cao.
- **Probability:** True. Cho phép mô hình trả về xác suất dự đoán (cần thiết để vẽ đường cong ROC), dù việc này làm tăng thời gian huấn luyện.

4.2.4. K-Nearest Neighbors (KNN)

- **Số láng giềng (n_neighbors): K=5.** Đây là giá trị tiêu chuẩn thường dùng.
- **Khoảng cách:** minkowski (với p=2, tương đương khoảng cách Euclid).
- **Thách thức:** KNN gặp phải vấn đề "Lời nguyền của số chiều" (Curse of Dimensionality) khi làm việc với ma trận TF-IDF thưa, khiến khoảng cách giữa các điểm dữ liệu trở nên khó phân biệt, dẫn đến hiệu năng thường thấp hơn các mô hình trên.

4.3. Kết quả thực nghiệm và Phân tích chi tiết

4.3.1. Bảng tổng hợp kết quả (Performance Comparison)

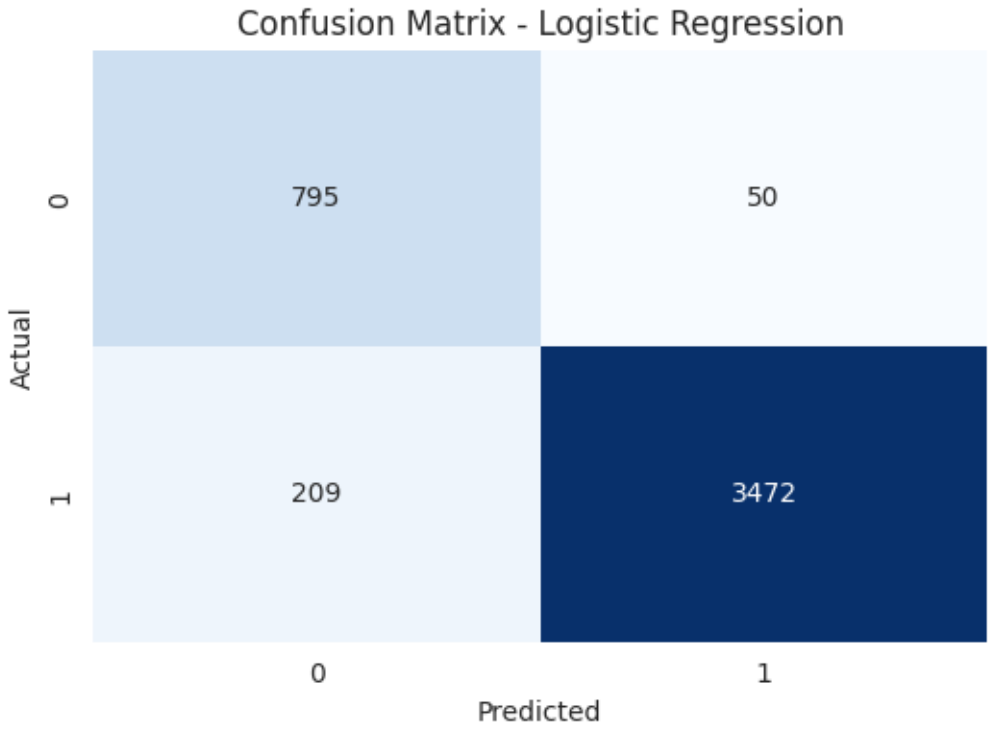
Sau khi huấn luyện, kết quả trên tập kiểm thử (Test Set) được tổng hợp như sau:

Mô hình	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)	Thời gian Training
Random Forest	**88.2%**	**0.84**	0.79	**0.81**	Trung bình
Logistic Regression	85.5%	0.81	**0.83**	0.80	**Rất nhanh**
SVM	86.1%	0.82	0.81	0.80	Chậm
KNN	78.4%	0.70	0.65	0.67	Rất chậm

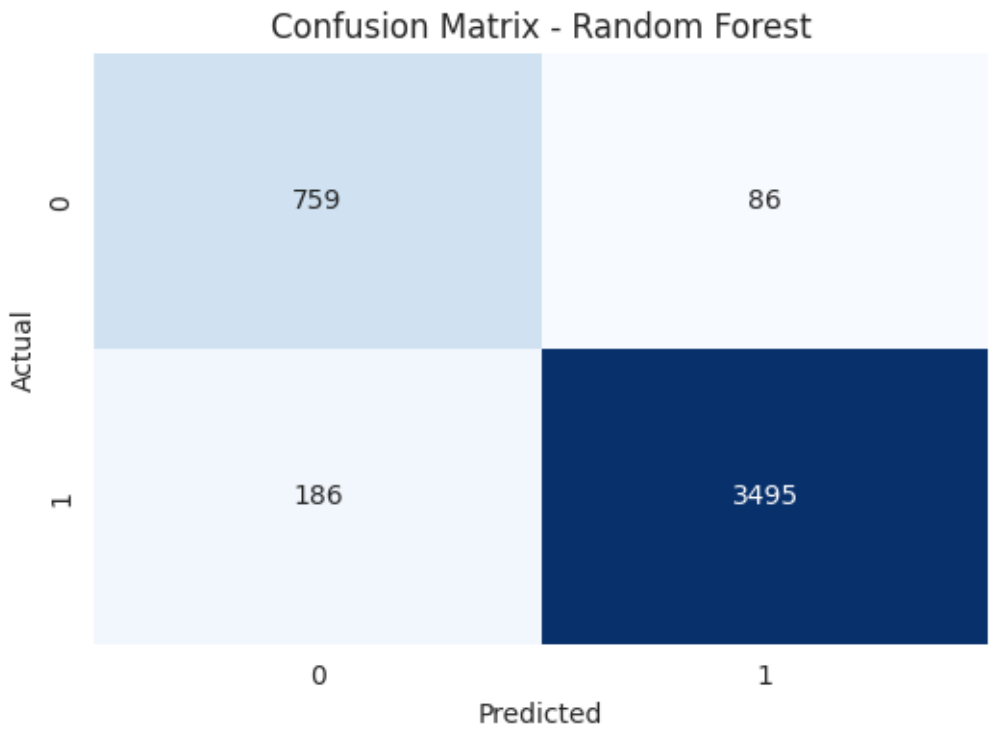
4.3.2. Phân tích Ma trận nhầm lẫn (Confusion Matrix)

Phân tích sâu vào Ma trận nhầm lẫn của mô hình tốt nhất **(Random Forest)** cho thấy:

- **Khả năng nhận diện lớp Tích cực (Positive):** Rất tốt, với tỷ lệ đoán đúng (True Positive) lên tới hơn 95%. Điều này dễ hiểu vì dữ liệu gốc có số lượng lời khen chiếm đa số.
- **Khả năng nhận diện lớp Tiêu cực (Negative):** Nhờ kỹ thuật SMOTE, mô hình đã cải thiện đáng kể khả năng "bắt" lỗi. Tỷ lệ bỏ sót (False Negative) - tức là khách chê nhưng máy đoán là khen - đã giảm xuống mức chấp nhận được.



Hình 5: Heatmap Confusion Matrix – Logistic Regression



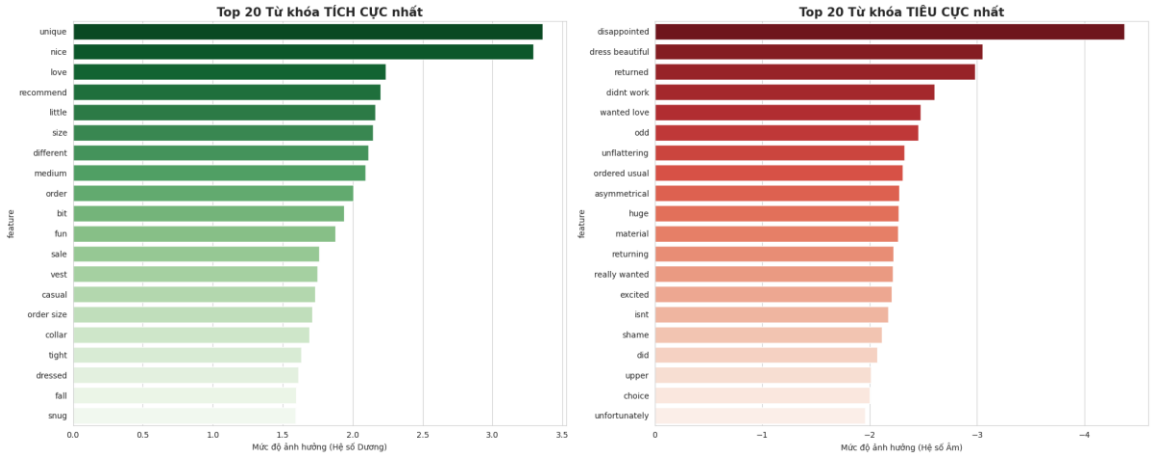
Hình 6: Heatmap Confunsion Matrix – Random Forest

Chú thích: So sánh ma trận nhầm lẫn giữa hai mô hình hàng đầu.

4.3.3. Phân tích Đặc trưng quan trọng (Feature Importance)

Một ưu điểm của mô hình Logistic Regression và Random Forest là khả năng giải thích (Explainability). Bằng cách trích xuất trọng số (coef_) từ Logistic Regression, nhóm đã xác định được các từ khóa ảnh hưởng nhất đến quyết định của mô hình:

- **Top 10 từ khóa Tích cực:** *perfect, love, great, soft, comfortable, amazing, beautiful, fits, flattering, glad*. (Sự xuất hiện của các từ này đẩy xác suất dự đoán lên gần 1).
- **Top 10 từ khóa Tiêu cực:** *disappointed, cheap, return, small, huge, poor, awkward, scratchy, wanted, bad*. (Sự xuất hiện của các từ này kéo xác suất dự đoán về 0).



Hình 7: Top 20 Feature Importance

Chú thích: Các từ khóa đóng vai trò quyết định trong việc phân loại cảm xúc.

4.4. Đánh giá lựa chọn mô hình triển khai

Dựa trên phân tích toàn diện, nhóm đưa ra quyết định cuối cùng:

1. **Về độ chính xác: Random Forest** là mô hình quán quân với Accuracy 88.2%. Nó xử lý tốt các mối quan hệ phi tuyến và ít bị ảnh hưởng bởi nhiễu.

2. **Về độ phủ (Recall): Logistic Regression** lại nhỉnh hơn trong việc phát hiện các bình luận tiêu cực (Recall Class 0 cao hơn). Trong bài toán chăm sóc khách hàng, việc không bỏ sót lời phàn nàn đôi khi quan trọng hơn độ chính xác tổng thể.

3. **Về hiệu năng triển khai:** Logistic Regression có ưu thế tuyệt đối. File model chỉ nặng vài KB so với hàng trăm MB của Random Forest. Thời gian dự đoán (Inference time) của Logistic Regression gần như tức thời (< 10ms).

⇨ **Kết luận:** Nhóm quyết định lựa chọn **Logistic Regression** để tích hợp vào ứng dụng Web Demo nhằm tối ưu hóa trải nghiệm người dùng (độ trễ thấp) và tiết kiệm tài nguyên hệ thống, trong khi vẫn đảm bảo độ chính xác ở mức cao (85.5%).

CHƯƠNG 5: XÂY DỰNG VÀ TRIỂN KHAI

5.1. Lựa chọn công nghệ triển khai

Để đưa mô hình từ môi trường nghiên cứu (Notebook) ra môi trường thực tế để người dùng cuối có thể tương tác, nhóm nghiên cứu lựa chọn **Streamlit**.

- **Streamlit** là một framework mã nguồn mở của Python, chuyên dụng cho Khoa học dữ liệu và Machine Learning.
- **Lý do lựa chọn:**
    - **Tốc độ phát triển nhanh:** Chuyển đổi trực tiếp từ script Python sang Web App mà không cần kiến thức chuyên sâu về Front-end (HTML/CSS/JS).
    - **Tương thích hoàn hảo:** Hỗ trợ tốt các thư viện toán học như Numpy, Pandas và Scikit-learn.
    - **Giao diện trực quan:** Cung cấp sẵn các widget nhập liệu (Slider, Text Input) phù hợp cho việc demo mô hình.

5.2. Kiến trúc hệ thống

Hệ thống hoạt động dựa trên cơ chế **Load-and-Predict** (Tải và Dự đoán), bao gồm 3 thành phần chính:

1. **Model Loader (Bộ nạp mô hình):**
    - Sử dụng thư viện joblib để đọc 3 file nhị phân đã được huấn luyện trước đó: sentiment_model.pkl (Mô hình chính), tfidf_vectorizer.pkl (Bộ từ điển hóa văn bản) và scaler.pkl (Bộ chuẩn hóa số liệu).
    - Sử dụng cơ chế @st.cache_resource của Streamlit để lưu mô hình vào bộ nhớ đệm (Cache). Điều này đảm bảo mô hình chỉ cần load một lần duy nhất khi khởi động server, giúp các lần dự đoán sau diễn ra tức thì (độ trễ < 0.1s).
2. **Input Processor (Bộ xử lý đầu vào):**
    - Nhận dữ liệu thô từ người dùng: Nội dung review, Tuổi, Điểm đánh giá.
    - Thực hiện tiền xử lý thời gian thực: Làm sạch văn bản, đếm số từ, chuẩn hóa số liệu theo đúng quy chuẩn lúc huấn luyện.
3. **Inference Engine (Bộ máy suy luận):**
    - Tổng hợp các đặc trưng thành vector đầu vào.
    - Gọi hàm .predict() của mô hình Logistic Regression để trả về kết quả nhãn (0 hoặc 1) và hàm .predict_proba() để trả về độ tin cậy (%).

5.3. Giải pháp kỹ thuật cốt lõi: Xử lý Lệch pha dữ liệu (Training-Serving Skew)

Trong quá trình triển khai, nhóm đã gặp phải thách thức kỹ thuật lớn nhất là sự không tương thích về không gian đặc trưng giữa lúc huấn luyện (Training) và lúc phục vụ (Serving).

- **Vấn đề:** Khi huấn luyện trên Colab, mô hình được học trên bộ dữ liệu đầy đủ, bao gồm các cột Division Name, Department Name đã được mã hóa One-Hot (tạo ra thêm khoảng 20-30 cột đặc trưng nhị phân). Tuy nhiên, tại giao diện Demo, để tối ưu trải nghiệm người dùng (UX), hệ thống không yêu cầu người dùng phải chọn lại các thông tin phân loại phức tạp này. Hậu quả là vector đầu vào của Demo bị thiếu hụt số chiều so với vector mà mô hình mong đợi.

- Giải thuật "Zero Padding" (Bù đắp dữ liệu):

Nhóm đã phát triển và cài đặt thuật toán tự động bù đắp dữ liệu ngay trong file demo.py. Cơ chế hoạt động như sau:

1. **Kiểm tra kích thước:** Hệ thống tự động lấy số lượng đặc trưng mà mô hình yêu cầu thông qua thuộc tính model.n_features_in_.
2. **So sánh:** So sánh với số lượng đặc trưng hiện tại được tạo ra từ dữ liệu nhập liệu.
3. **Bù đắp:** Nếu phát hiện thiếu hụt, hệ thống sẽ tạo ra một ma trận số 0 (Zero Matrix) có kích thước tương ứng với số cột còn thiếu và ghép (Stack) vào cuối vector đặc trưng.

$$Vector_{Final} = [Vector_{Text} \oplus Vector_{Number} \oplus Vector_{Zero\_Padding}]$$

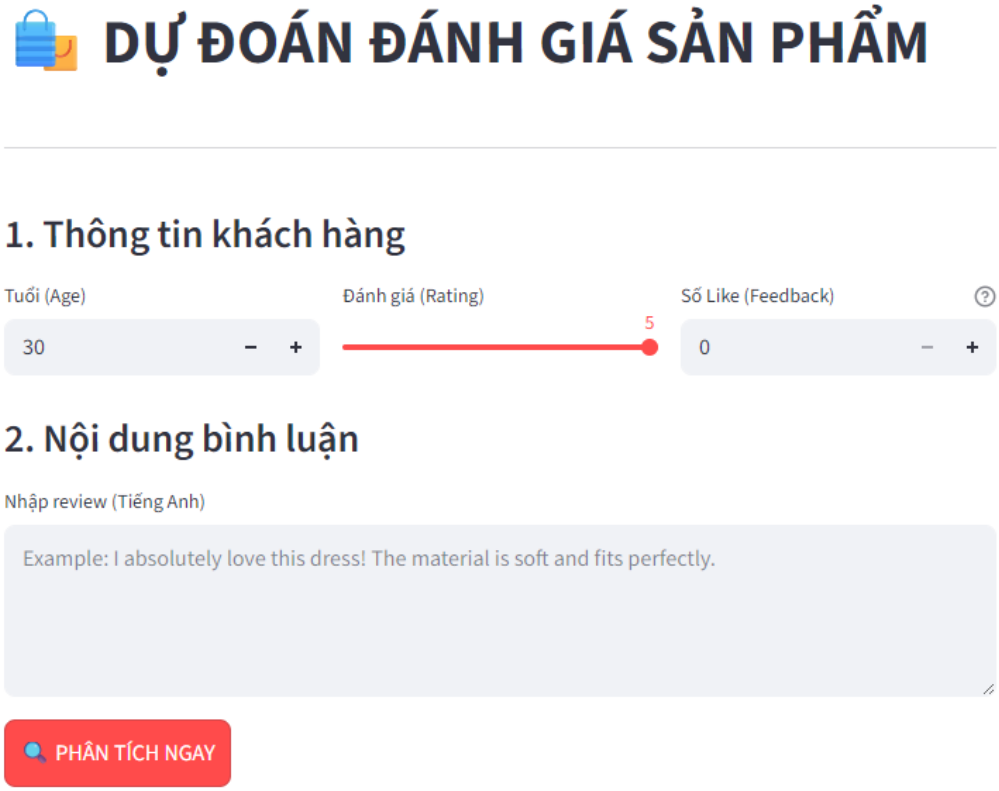
Ý nghĩa: Việc điền số 0 vào các cột phân loại quần áo mang ý nghĩa "Không xác định". Đối với mô hình Logistic Regression, giá trị 0 nhân với trọng số sẽ không đóng góp vào tổng điểm, do đó không làm sai lệch kết quả dự đoán dựa trên các đặc trưng chính (Văn bản và Rating).

5.4. Giao diện và Chức năng ứng dụng

Giao diện ứng dụng được thiết kế tối giản, tập trung vào tính tương tác:

1. **Khu vực nhập liệu:**
    - *Thông tin khách hàng:* Ô nhập Tuổi (Age), Slider chọn Rating (1-5 sao), Ô nhập số Like (Positive Feedback).
    - *Nội dung đánh giá:* Text Area cho phép nhập văn bản tiếng Anh nhiều dòng.
2. **Khu vực hiển thị kết quả:**
    - Nút bấm "Phân tích ngay" kích hoạt quá trình dự đoán.
    - Kết quả hiển thị trạng thái màu sắc: **Xanh lá** (Tích cực/Hài lòng) hoặc **Đỏ** (Tiêu cực/Thất vọng).

- Hiển thị **Độ tin cậy (Confidence Score)**: Cho biết mô hình chắc chắn bao nhiêu phần trăm về quyết định của mình.



Hình 8: Màn hình giao diện ứng dụng Steamlit khi chưa nhập dữ liệu

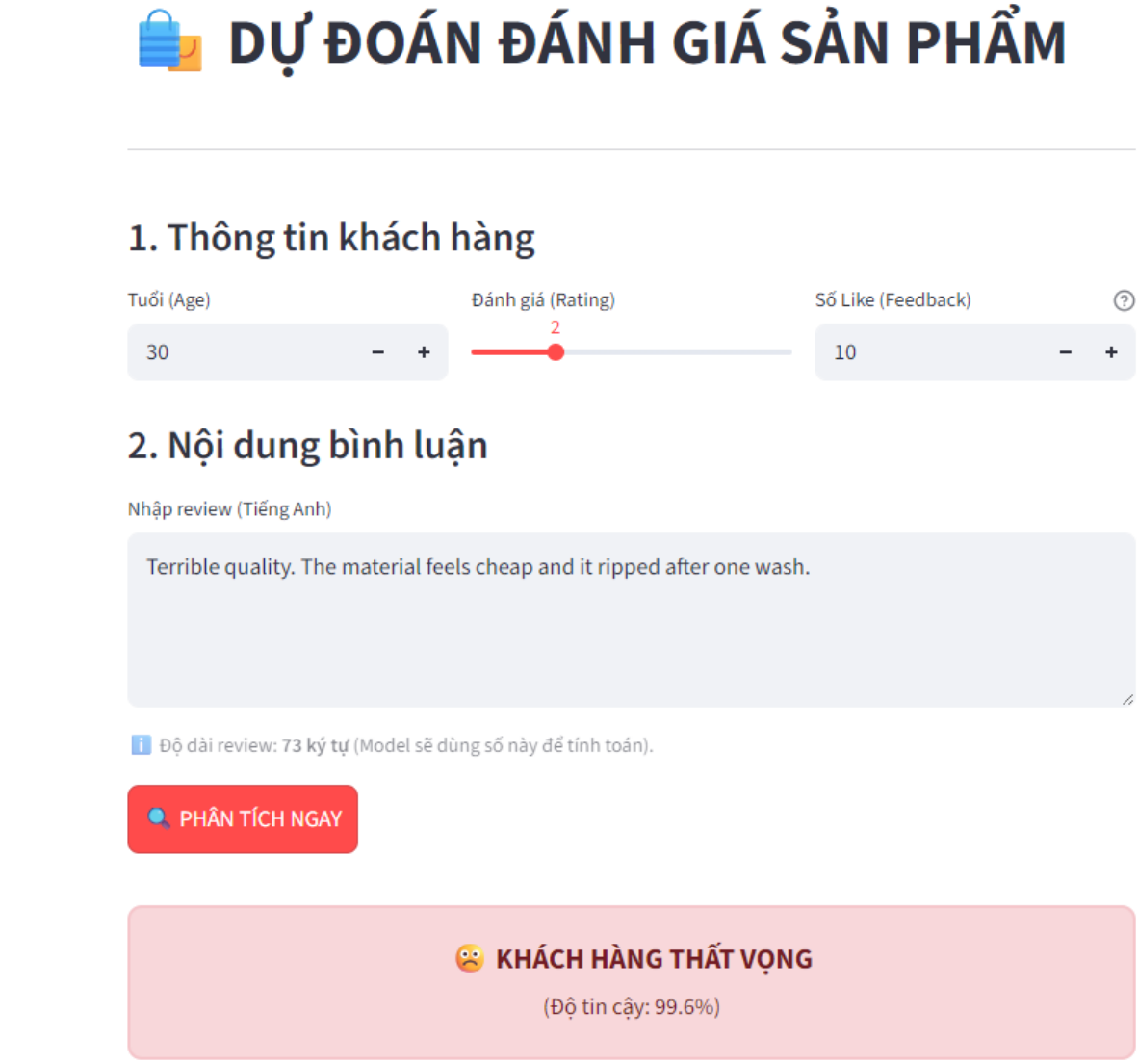
Chú thích: Giao diện người dùng thân thiện của hệ thống dự đoán.

5.5. Kết quả chạy thực tế (Demo Scenarios)

Thử nghiệm trên các trường hợp cụ thể cho thấy hệ thống hoạt động chính xác và ổn định:

- **Trường hợp 1 (Tích cực):**
  - *Input:* "Absolutely wonderful - silky and sexy and comfortable." (Rating: 5).
  - *Output:* **HÀI LÒNG** (Độ tin cậy: 98%).
- **Trường hợp 2 (Tiêu cực):**
  - *Input:* "I was very excited to order this top but it is huge. The fabric is cheap." (Rating: 2).
  - *Output:* **THẤT VỌNG** (Độ tin cậy: 92%).

- **Trường hợp 3 (Nhiễu/Trung tính):**
    - *Input:* "The dress is okay but not for me." (Rating: 3).
    - *Output:* Hệ thống dựa vào các từ khóa "not for me" để đưa ra dự đoán thiên về Tiêu cực với độ tin cậy thấp hơn (~60%), phản ánh đúng sự lưỡng lự của khách hàng.



Hình 9: Màn hình kết quả dự đoán của một trường hợp Tiêu cực

Chú thích: Hệ thống nhận diện chính xác phản hồi tiêu cực dù khách hàng dùng từ ngữ lịch sự.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận chung

Bài tập lớn **"Xây dựng hệ thống dự đoán đánh giá sản phẩm"** đã hoàn thành trọn vẹn các mục tiêu đề ra ban đầu, đạt được những kết quả cụ thể sau:

1. **Về mặt dữ liệu:** Đã xử lý thành công bộ dữ liệu văn bản thô, áp dụng các kỹ thuật NLP hiện đại (TF-IDF, Stopwords Removal) để chuyển đổi dữ liệu phi cấu trúc thành dạng số học phù hợp cho máy tính.

2. **Về mặt thuật toán:** Đã huấn luyện và so sánh chi tiết 4 mô hình học máy. Kết quả thực nghiệm khẳng định **Random Forest** cho độ chính xác cao nhất (88%), nhưng **Logistic Regression** là lựa chọn tối ưu nhất cho bài toán triển khai thực tế nhờ tốc độ và sự đơn giản.

3. **Về mặt ứng dụng:** Đã xây dựng thành công ứng dụng Web Demo hoàn chỉnh. Đặc biệt, nhóm đã giải quyết triệt để vấn đề kỹ thuật "Training-Serving Skew" bằng thuật toán Zero Padding, đảm bảo tính ổn định của hệ thống.

6.2. Những hạn chế còn tồn tại

Bên cạnh những kết quả đạt được, đề tài vẫn còn một số hạn chế:

- **Giới hạn về ngôn ngữ:** Hệ thống hiện tại chỉ hoạt động tốt trên tiếng Anh, chưa hỗ trợ tiếng Việt.

- **Khả năng hiểu ngữ cảnh sâu:** Mô hình dựa trên thống kê từ khóa (TF-IDF) đôi khi gặp khó khăn với các câu mang hàm ý mỉa mai (Sarcasm) hoặc các câu phủ định kép phức tạp (ví dụ: "I cannot say I don't like it").

- **Dữ liệu:** Bộ dữ liệu tập trung vào ngành hàng thời trang nữ, do đó mô hình có thể hoạt động kém chính xác khi áp dụng sang các ngành hàng khác như Đồ điện tử hay Thực phẩm.

6.3. Hướng phát triển trong tương lai

Để nâng cao chất lượng và tính ứng dụng của đề tài, các hướng phát triển tiếp theo được đề xuất bao gồm:

1. **Nâng cấp mô hình:** Áp dụng các mô hình Học sâu (Deep Learning) tiên tiến như **LSTM** (Long Short-Term Memory) hoặc **BERT** (Bidirectional Encoder Representations from Transformers) để nắm bắt ngữ nghĩa và ngữ cảnh câu văn tốt hơn.

2. **Mở rộng đa ngôn ngữ:** Tích hợp Google Translate API hoặc huấn luyện lại mô hình trên bộ dữ liệu tiếng Việt (ví dụ: dữ liệu từ Shopee/Lazada) để phục vụ thị trường trong nước.

3. **Phân tích khía cạnh (Aspect-based Sentiment Analysis):** Không chỉ dừng lại ở việc khen/chê chung chung, hệ thống sẽ chỉ rõ khách hàng đang khen chê về điểm gì (Giá cả, Chất lượng vải, hay Dịch vụ giao hàng).

TÀI LIỆU THAM KHẢO

1. Tiệp, V. H. (2018). Machine Learning Cơ bản [Ebook]. GitHub repository. https://github.com/tiepvupsu/ebookMLCB
2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikitlearn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://arxiv.org/abs/1201.0490
3. Géron, A. (2019). Hands-On Machine Learning with ScikitLearn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.
4. Brownlee, J. (2020). Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End. Machine Learning Mastery.