BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC KINH TẾ THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ

UEH
UNIVERSITY

ĐỒ ÁN MÔN HỌC

PHÂN TÍCH QUAN ĐIỂM ĐÁNH GIÁ CHẤT LƯỢNG DỊCH VỤ KHÁCH SẠN TRÊN NỀN TẢNG TRIPADVISOR

Học Phần: Xử Lý Ngôn Ngữ Tự Nhiên

Nhóm Sinh Viên:

1. ĐỖ THỊ MỸ KHÁNH - 31221024404
2. TRẦN NGUYỄN THẢO NGUYÊN - 31221023350
3. NGUYỄN HỮU THANH - 31221021356
4. NGUYỄN VÂN PHI YẾN - 31221021785

Chuyên Ngành: KHOA HỌC DỮ LIỆU

Khóa: K48

Giảng Viên: TS. Đặng Ngọc Hoàng Thành

TP Hồ Chí Minh, ngày 16 tháng 12 năm 2024

LỜI CẢM ƠN

Để có thể hoàn thiện bài luận báo cáo cuối kỳ môn học *Xử lý ngôn ngữ tự nhiên* với đề tài *"Phân tích quan điểm đánh giá chất lượng dịch vụ khách sạn trên nền tảng TripAdvisor"*, nhóm chúng em xin gửi lời tri ân sâu sắc đến Thầy TS. Đặng Ngọc Hoàng Thành, giảng viên hướng dẫn môn học trực thuộc khoa Công nghệ Thông tin Kinh doanh của Đại học Kinh tế Thành phố Hồ Chí Minh.

Nhóm em xin gửi lời cảm ơn chân thành đến Thầy vì sự tận tâm trong giảng dạy và những kiến thức quý báu mà Thầy đã truyền đạt trong suốt quá trình học môn *Xử lý ngôn ngữ tự nhiên*. Thầy không chỉ cung cấp nền tảng vững chắc về lý thuyết mà còn khuyến khích chúng em mở rộng tư duy, tìm tòi và nghiên cứu các phương pháp mới. Nhờ đó, nhóm em có thêm động lực để áp dụng các kỹ thuật học máy và học sâu vào đề tài *"Phân tích quan điểm đánh giá chất lượng dịch vụ khách sạn trên nền tảng TripAdvisor"*. Những chỉ dẫn và sự hỗ trợ của Thầy đã giúp nhóm em hoàn thành tốt đề tài và nâng cao kỹ năng nghiên cứu cũng như ứng dụng thực tiễn.

Trong quá trình thực hiện bài tiểu luận, nhóm chúng em đã cố gắng vận dụng những kiến thức đã được học và nghiên cứu thêm nhiều tài liệu để hoàn thiện bài tiểu luận này. Tuy nhiên, do kiến thức còn hạn chế và thiếu kinh nghiệm thực tiễn nên nội dung bài tiểu luận khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ Thầy để nhóm có thể hoàn thiện bài làm hơn.

Nhóm chúng em xin chân thành cảm ơn Thầy vì sự đồng hành và hỗ trợ trong suốt quá trình học tập và nghiên cứu.

Trân trọng,
Nhóm thực hiện

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BIỂU ĐỒ

CHƯƠNG 1 - TỔNG QUAN ĐỀ TÀI

1.1. Lý Do Chọn Đề Tài

Tại Việt Nam, nhờ có văn hóa đa dạng và cảnh sắc thiên nhiên nổi bật, du lịch dần trở thành một trong những ngành kinh tế mũi nhọn, ảnh hưởng đến sự phát triển kinh tế - xã hội. Theo Tổng cục Du lịch Việt Nam, vào thời điểm trước đại dịch (2019), Việt Nam đã đón hơn 18 triệu lượt khách quốc tế, mang lại tổng doanh thu trên 720.000 tỷ đồng (tương đương khoảng 30 tỷ USD). Sau đại dịch, du lịch Việt Nam đang trên đà phục hồi mạnh mẽ; chỉ trong 10 tháng đầu năm 2023, Việt Nam đã đón hơn 10 triệu lượt khách quốc tế, vượt xa kỳ vọng (Tổng cục Du lịch, 2023). Do đó, chất lượng dịch vụ lưu trú là một vấn đề cần được quan tâm hơn nữa bởi đây cũng là một trong những yếu tố ảnh hưởng trực tiếp đến trải nghiệm của khách du lịch.

Trong xu thế hiện nay, các nền tảng về tìm kiếm và đánh giá dịch vụ du lịch, bao gồm cả khách sạn, luôn là ưu tiên hàng đầu của khách hàng khi muốn tìm hiểu về nơi muốn đến, một trong số đó là TripAdvisor. Được thành lập từ năm 2000, TripAdvisor hiện có hơn 1 tỷ lượt đánh giá và ý kiến về hàng triệu khách sạn, nhà hàng và điểm đến trên toàn cầu, bao gồm cả Việt Nam. Những đánh giá này cung cấp dữ liệu phong phú giúp phân tích và cải thiện chất lượng dịch vụ khách sạn, tạo niềm tin cho du khách và gia tăng năng lực cạnh tranh của ngành du lịch.

1.2. Mục Tiêu Nghiên Cứu

Ở đề tài "*Phân tích quan điểm đánh giá chất lượng dịch vụ khách sạn trên TripAdvisor*" này, nhóm nghiên cứu hướng đến nhận diện được nhóm đánh giá tích cực (positive), tiêu cực (negative) và trung tính (neutral) dựa trên kỹ thuật xử lí ngôn ngữ tự nhiên, các mô hình học máy và học sâu. Từ đó, đánh giá hiệu quả của các mô hình nhằm tìm ra mô hình tối ưu cho đề tài.

1.3. Phương Pháp Nghiên Cứu

- *Tiền xử lý dữ liệu:* Sử dụng các thư viện bao gồm re, Pandas, NumPy, PyVi và NLTK để làm sạch và chuẩn hóa dữ liệu.
- *Phân tích dữ liệu:* Sử dụng mô hình máy học gồm Multinomial Logistic Regression, Multinomial Naive Bayes, SVM và mô hình học sâu FNN để xác định thái độ của các đánh giá theo nhóm tích cực, tiêu cực và trung tính.

➢ *Đánh giá mô hình:* Nhóm đánh giá hiệu quả mô hình dựa trên Ma trận nhầm lẫn và các chỉ số Accuracy, Precision, Recall, F1-Score, Support, Macro Average và Weighted Average.

1.4. Tài Nguyên Sử Dụng

➢ Bộ dữ liệu *"TripAdvisor Vietnam Hotel Reviews"* được đăng tải trên NIAID Data Ecosystem vào ngày 23/05/2023, bởi nhóm tác giả An Dinh Van, Trinh Tran Thi Kieu, Hieu Tran Nguyen Ngoc, Anh Nguyen Thi Linh và Thao Huynh Nhi Thanh.

➢ *Ngôn ngữ và thư viện lập trình:*

- Ngôn ngữ lập trình Python: Dùng để xử lý dữ liệu, phân tích và biểu diễn trực quan
- Các thư viện hỗ trợ:
  - pandas
  - numpy
  - matplotlib
  - seaborn
  - scikit-learn
  - tensorflow
  - nltk

CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT

2.1. Các Phương Pháp Tiền Xử Lý Dữ Liệu

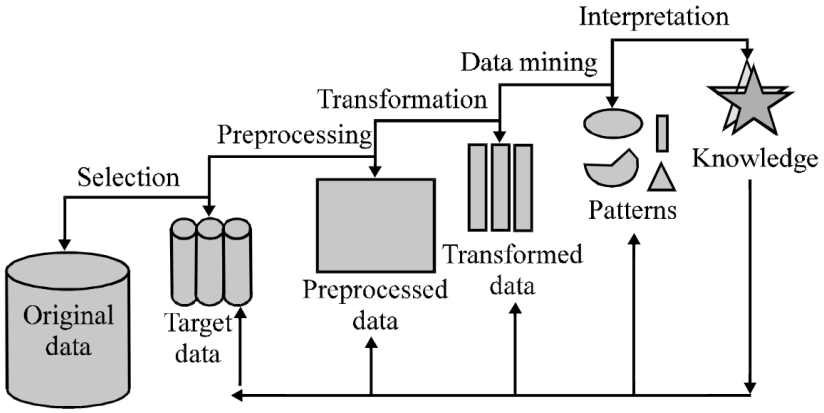
2.1.1. Tiền Xử Lý Dữ Liệu (Data Preprocessing)

Tiền xử lý dữ liệu *(Data Preprocessing)* là một trong những kỹ thuật quan trọng trong khai phá dữ liệu, bao gồm các bước chuẩn bị *(Data Preparation)* và chuyển đổi dữ liệu *(Data Transformation)* nhằm đảm bảo dữ liệu đầu vào đạt chất lượng, phù hợp cho quá trình phân tích và huấn luyện cho các mô hình học máy.

Mục tiêu của quy trình tiền xử lý dữ liệu nhằm giảm kích thước dữ liệu, phân tích mối tương quan giữa các thuộc tính, chuẩn hóa dữ liệu, loại bỏ các giá trị ngoại lai *(outliers)* và xử lý các giá trị bị thiếu, từ đó giúp tối ưu hóa hiệu suất hoạt động và độ chính xác của các mô hình học máy. Theo Kotsiantis và cộng sự (2006), dữ liệu thô thường chứa nhiều thông tin không liên quan hoặc không nhất quá, gây khó khăn cho quá trình phân tích. Nếu không được tiền xử lý đúng cách, mô hình học máy có thể sẽ đưa ra các kết quả kém chính xác và không hiệu quả trong việc khai phá các thông tin giá trị từ dữ liệu.

Quy trình tiền xử lý dữ liệu bao gồm các bước như sau, được minh họa trong *Hình 2.1 (Tamilselvi et al., 2015)*.

- ➢ Làm sạch dữ liệu *(Data Cleaning):* Phát hiện và xử lý các giá trị bị thiếu, giá trị nhiễu hoặc dữ liệu bất thường.
- ➢ Tích hợp dữ liệu *(Integration):* Kết hợp dữ liệu từ nhiều nguồn khác nhau thành một tập dữ liệu thống nhất.
- ➢ Chuyển đổi dữ liệu *(Transformation):* Bao gồm chuẩn hóa, rời rạc hóa và trích xuất đặc trưng nhằm chuyển đổi dữ liệu về dạng phù hợp để huấn luyện cho các mô hình máy học.
- ➢ Giảm chiều dữ liệu *(Reduction):* Giảm số lượng thuộc tính hoặc các quan sát không cần thiết nhằm tối ưu hóa tốc độ xử lý dữ liệu.

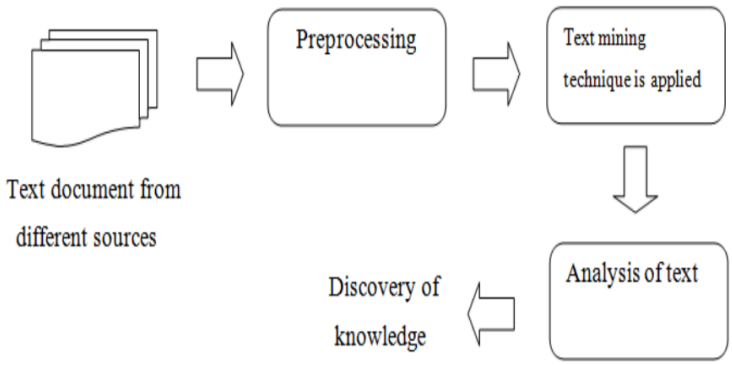


Hình 2.1. Các bước khai phá dữ liệu (Tamilselvi et al., 2015)

2.1.2. Text Mining

Text Mining, hay còn gọi là khai phá dữ liệu văn bản, là một nhánh của lĩnh vực khai phá dữ liệu *(data mining),* đóng vai trò quan trọng trong quá trình khai phá tri thức. Text Mining trích xuất thông tin tiềm ẩn từ dữ liệu văn bản không cấu trúc *(not-structured data)* đến dữ liệu bán cấu trúc *(semi-structured data)*.

Quy trình hoạt động của Text Mining bắt đầu bằng việc thu thập tài liệu từ các nguồn khác nhau. Tiếp đến, sẽ tiến hành truy xuất một tài liệu cụ thể và tiền xử lý dữ liệu bằng cách kiểm tra định dạng và các ký tự trong bộ dữ liệu. Sau đó, sẽ trải qua giai đoạn phân tích văn bản. Phân tích văn bản *(Text analysis)* là việc phân tích ngữ nghĩa của văn bản nhằm rút ra được các thông tin giá trị từ văn bản. Quá trình này sẽ được lặp lại cho đến khi trích xuất được lượng thông tin cần thiết. Thông tin thu được sẽ được đưa vào hệ thống thông tin quản lý nhằm cung cấp một lượng lớn tri thức cho người sử dụng hệ thống.



Hình 2.2. Quy trình khai phá văn bản (Text Mining Process)

Các kỹ thuật chính được sử dụng trong Text Mining bao gồm:

- ➢ Trích xuất thông tin (Information Extraction)
- ➢ Phân loại văn bản (Categorization)
- ➢ Phân cụm văn bản (Clustering)
- ➢ Tóm tắt văn bản (Summarization)
- ➢ Trực quan hóa dữ liệu văn bản (Visualization)

2.2. Các Mô Hình Học Máy

2.2.1. TF-IDF Vectorizer

TF-IDF *(Term Frequency - Inverse Document Frequency)* là một kỹ thuật vector hóa văn bản thường được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên để chuyển đổi văn bản thành dạng số, giúp các mô hình học máy có thể xử lý được. Khái niệm TF-IDF bao gồm hai thành phần quan trọng: *Tần suất xuất hiện từ (Term Frequency - TF)* và *Tần suất nghịch đảo văn bản (Inverse Document Frequency - IDF).*

TF-IDF đo lường tầm quan trọng của một từ trong một văn bản cụ thể so với toàn bộ tập dữ liệu. Giá trị TF-IDF của một từ càng tăng khi từ đó xuất hiện nhiều lần trong văn bản, nhưng đồng thời được điều chỉnh bởi tần suất xuất hiện của từ trong toàn bộ tập văn bản.

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

$$tf(t,d) = \frac{số\ lần\ xuất\ hiện\ của\ từ\ t\ trong\ văn\ bản\ d}{tổng\ số\ từ\ trong\ văn\ bản\ d}$$

- ➢ *Tần suất nghịch đảo văn bản (Inverse Document Frequency -IDF):* Đo lường mức độ phổ biến của một từ trong toàn bộ tập văn bản N. Từ càng xuất hiện nhiều trong các văn bản, trọng số của nó càng nhỏ.

$$idf(t) = log(\frac{N}{df(t)})$$

- ➢ *Điểm số TF-IDF của từ t trong văn bản d được tính như sau: (Arsyah et al., 2024)*

$$tfidf(t,d) = tf(t,d) \times idf(t)$$

2.2.2. Mô Hình Multinomial Logistic Regression

Trong giới hạn bài nghiên cứu "*Phân tích quan điểm đánh giá chất lượng dịch vụ khách sạn trên TripAdvisor*", nhóm thực hiện cần tiến hành phân loại dữ liệu thành ba lớp cảm xúc: '*positive', 'negative'* và '*neutral'*. Do đó, nhóm lựa chọn mô hình mở rộng hơn

của mô hình Logistic Regression là *'Mô hình Multinomial Logistic Regression'*. Khác với Logistic Regression nhị phân chỉ có thể giải quyết được 2 lớp, Multinomial Logistic Regression cho phép phân loại dữ liệu thành nhiều hơn hai lớp một cách chính xác và hiệu quả hơn, phù hợp với mục tiêu của bài toán phân tích cảm xúc.

Mô hình Multinomial Logistic Regression (hay Multi Class Logistic Regression), còn được gọi là Softmax Regression, là một dạng mở rộng của Logistic Regression để giải quyết các bài toán phân loại đa lớp, tức là với nhiều hơn hai kết quả có thể xảy ra (Vryniotis, V. (2013)).

Thay vì hàm sigmoid trong Logistic Regression nhị phân, Multinomial Logistic Regression đã mở rộng khái niệm thành hàm Softmax để có thể tính xác suất cho từng lớp trong bài toán phân loại đa lớp. Với một tập k lớp, xác suất $P(y = c|x)$ của một lớp c được tính bằng công thức:

$$P(y = c|x; \theta) = \frac{exp(\theta_c^T x)}{\sum_{j=1}^{k} exp(\theta_j^T x)}$$

Trong đó:

- x: Vector đặc trưng đầu vào
- θ_c: Vector trọng số tương ứng với lớp c
- k: Tổng số lớp trong bài toán
- exp: Hàm mũ tự nhiên e^z

Hàm Softmax đảm bảo: $\sum_{c=1}^{k} P(y = c|x) = 1$, đồng nghĩa tổng xác suất của tất cả các lớp sẽ bằng 1. Lớp xác suất lớn nhất sẽ được chọn làm đầu ra dự đoán.

2.2.3. Mô Hình Multinomial Naive Bayes

Thuật toán Multinomial Naive Bayes (MNB) là một biến thể phổ biến của Naive Bayes. Thuật toán này giả định rằng tần suất xuất hiện của các đặc trưng (features) trong mỗi danh mục là một phân phối đa thứ (phân phối multinomial) (McCallum & Nigam, 1998). Do đó, thuật toán MNB thích hợp với dữ liệu dưới dạng tần suất hoặc tần suất có trọng số trong dữ liệu. Nhờ vậy, thuật toán được áp dụng nhiều trong những bài toán phân

loại dữ liệu rời rạc, như bài toán phân tích văn bản và nhận diện ngôn ngữ (Jurafsky & Martin, 2020).

Vì là một biến thể của Naive Bayes, thuật toán MNB hoạt động theo nguyên lí Bayes. Cơ chế chính của định lý là cho phép chúng ta cập nhật kiến thức về xác suất của một sự kiện dựa trên thông tin bổ sung (Hastie et al., 2009). Định lý Bayes được phát biểu với công thức như sau:

$$P(A|B) = P(B|A)\ \ P(A)P(B)$$

Trong đó:

- *P(A|B) là xác suất của lớp A khi biết đặc trưng B.*
- *P(B|A) là xác suất của đặc trưng B khi biết lớp A.*
- *P(A) là xác suất của lớp A.*
- *P(B) là xác suất của đặc trưng B.*

Trong thực tế, thuật toán Multinomial Naive Bayes được ứng dụng trong

- **Phân loại email**: Xem xét email là spam hay hợp lệ (McCallum & Nigam, 1998).
- **Phân loại tin tức**: Xác định danh mục các bài báo (Bishop, 2006).
- **Phân tích cảm xúc**: Đánh giá thái độ tích cực hay tiêu cực trong bài đánh giá hay bình luận (Jurafsky & Martin, 2020).

2.2.4. Mô Hình Support Vector Machine

Support Vector Machine (SVM) là một thuật toán học máy mạnh mẽ chủ yếu được sử dụng cho các bài toán phân loại và hồi quy. Về cơ bản, SVM tìm kiếm một siêu phẳng (hyperplane) tối ưu trong không gian đặc trưng để phân chia các điểm dữ liệu thuộc các lớp khác nhau sao cho biên độ phân tách giữa các lớp là rộng nhất có thể. Mục tiêu của SVM là tối đa hóa khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất, hay còn gọi là các vector hỗ trợ (support vectors). Quá trình này giúp mô hình có khả năng phân loại chính xác các điểm dữ liệu chưa được thấy trong tương lai (Cortes & Vapnik, 1995).

SVM có thể giải quyết cả các bài toán phân loại tuyến tính và không tuyến tính. Đối với các bài toán phân loại tuyến tính, SVM tìm một siêu phẳng phân chia các lớp trong không gian đầu vào. Tuy nhiên, khi dữ liệu không thể phân chia bằng một siêu phẳng, SVM sử dụng một kỹ thuật gọi là kernel trick để ánh xạ các dữ liệu vào không gian đặc trưng có chiều cao hơn, nơi dữ liệu có thể phân tách tuyến tính. Các hàm kernel phổ biến bao gồm:

- ➢ Kernel tuyến tính (Linear Kernel) dùng cho các bài toán phân loại tuyến tính.
- ➢ Kernel Gaussian hay Radial Basis Function (RBF) dùng cho các bài toán phân loại không tuyến tính (Schölkopf et al., 1997).
- ➢ Kernel đa thức (Polynomial Kernel) dùng cho các bài toán phức tạp hơn.

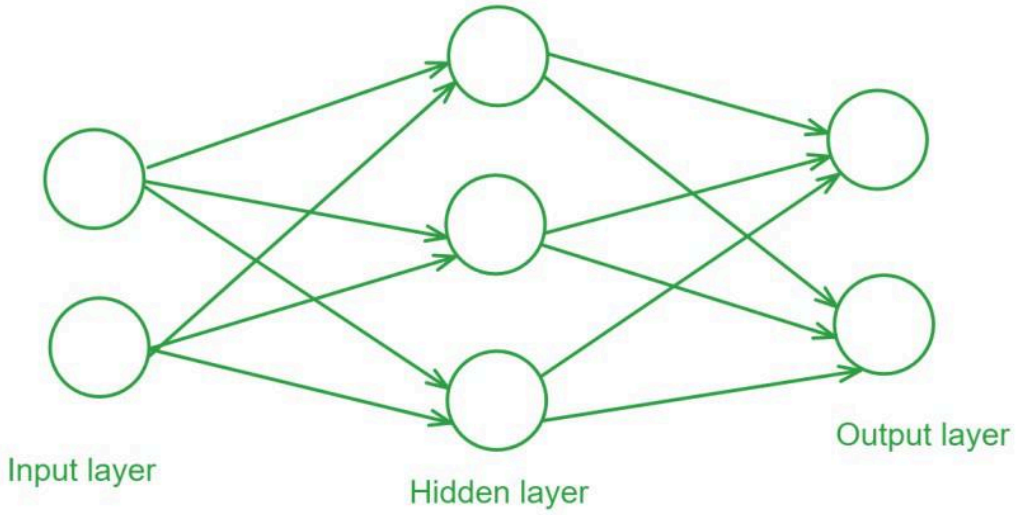
SVM đã được áp dụng thành công trong nhiều bài toán phân loại văn bản, đặc biệt trong các lĩnh vực như:

- ➢ Phân loại email thành spam hoặc không spam dựa trên các đặc trưng như từ khóa, tần suất từ, và các đặc trưng văn bản khác (Joachims, 1998).
- ➢ Phân loại tài liệu thành các nhóm khác nhau, ví dụ như phân loại báo cáo tài chính, bài viết khoa học, hay các bài luận văn (Cortes & Vapnik, 1995).
- ➢ Phân loại cảm xúc trong các bình luận, đánh giá, hoặc bài viết trên mạng xã hội. Ví dụ, các bài viết có thể được phân loại thành các nhóm cảm xúc tích cực, tiêu cực hoặc trung tính (Pang et al., 2002).

2.3. Mô Hình Feedforward Neural Network

Mạng nơ ron nhân tạo là một mô hình tính toán được lấy cảm hứng từ cách mà não bộ con người xử lý thông tin. Mô hình này bao gồm các lớp neuron (hay còn gọi là tế bào thần kinh nhân tạo) được kết nối với nhau để thực hiện việc xử lý dữ liệu và giải quyết các bài toán phức tạp. Trong đó, **Mạng Nơ-ron Truyền Thẳng (Feedforward Neural Network - FNN)** là một trong những mô hình cơ bản và được sử dụng rộng rãi nhất.

Mạng Nơ-ron Truyền Thẳng (Feedforward Neural Network - FNN) là một loại mạng nơ-ron nhân tạo, trong đó các kết nối giữa các nút (neuron) không tạo thành chu trình (cycle).



Hình . Mạng nơ ron truyền thẳng

(Nguồn: GeeksforGeeks, 2024)

2.3.1. Cấu Trúc Của Fnn

Bao gồm 3 thành phần chính:

Lớp đầu vào (Input layer): là các nơron nhận các biến đầu vào X_i với i=1,2,...l, l là số biến đầu vào. Các nơron lớp đầu vào nhận dữ liệu từ các biến đầu vào rồi chuyển cho các nơron ở lớp ẩn.

Lớp ẩn (Hidden layer): Gồm một hoặc nhiều lớp ẩn nằm giữa lớp đầu vào và lớp đầu ra. Các lớp này học các mẫu phức tạp trong dữ liệu, với mỗi nơron trong lớp ẩn tính toán tổng trọng số của các đầu vào, sau đó áp dụng một hàm kích hoạt phi tuyến.

Lớp đầu ra (Output layer): Cung cấp kết quả cuối cùng của mạng. Số lượng nơron trong lớp này phụ thuộc vào số lượng nhãn (classes) trong bài toán phân loại và số lượng đầu ra trong bài toán hồi quy.

Ngoài ra, các kết nối của các nơron đều có trọng số riêng, các trọng số này được điều chỉnh trong quá trình huấn luyện để giảm thiểu sai số dự đoán.

2.3.2. Quy Trình Huấn Luyện

Lan truyền xuôi (Forward Propagation): Dữ liệu đầu vào đi qua từng lớp từ lớp đầu vào, các lớp ẩn, đến lớp đầu ra. Mỗi neuron tính toán tổng trọng số của đầu vào, áp dụng hàm kích hoạt, và truyền tín hiệu.

Tính toán lỗi: Sai số giữa đầu ra dự đoán và đầu ra thực tế được tính bằng hàm mất mát, ví dụ như **Mean Squared Error** hoặc **Cross-Entropy Loss**.

Lan truyền ngược (Backpropagation): Sai số được truyền ngược từ đầu ra qua các lớp để tính toán độ dốc của hàm mất mát theo trọng số.

Cập nhật trọng số: Sử dụng thuật toán **Gradient Descent** hoặc các biến thể (như Adam Optimizer) để điều chỉnh trọng số, giảm sai số.

CHƯƠNG 3 - CÁC KẾT QUẢ THỰC NGHIỆM

3.1. Tổng Quan Bộ Dữ Liệu

3.1.1. Sơ Lược Về Bộ Dữ Liệu

Bộ dữ liệu được sử dụng trong đồ án là bộ dữ liệu *'TripAdvisor Vietnam Hotel Reviews'* được đăng tải trên NIAID Data Ecosystem vào ngày 23/05/2023, bởi nhóm tác giả An Dinh Van, Trinh Tran Thi Kieu, Hieu Tran Nguyen Ngoc, Anh Nguyen Thi Linh và Thao Huynh Nhi Thanh. Bộ dữ liệu thu thập các đánh giá công khai của người dùng trên nền tảng du lịch trực tuyến TripAdvisor, cung cấp trải nghiệm, ý kiến và mức độ hài lòng của khách hàng khi lưu trú tại nhiều khách sạn khác nhau trên khắp Việt Nam.

3.1.2. Mô Tả Thuộc Tính

Dữ liệu bao gồm 62,887 dòng, với 14 thuộc tính được mô tả chi tiết như sau:

- ***language*** *(object):* Ngôn ngữ của đánh giá
- ***rating*** *(int64):* Điểm đánh giá do người đánh giá cung cấp, nằm trong khoảng từ 1 đến 5
- ***additionalRatings*** *(object):* Các xếp hạng bổ sung khác mà người dùng có thể cung cấp
- ***createdDate*** *(object):* Ngày đăng tải đánh giá
- ***helpfulVotes*** *(int64):* Số lượng phiếu bầu hữu ích mà đánh giá nhận được từ người khác
- ***username*** *(object):* Tên người dùng đã đăng đánh giá
- ***userId*** *(object):* ID của người dùng
- ***title*** *(object):* Tiêu đề của đánh giá
- ***text*** *(object):* Nội dung đánh giá chi tiết do người dùng cung cấp
- ***locationId*** *(int64):* ID của khách sạn trong hệ thống TripAdvisor
- ***parentGeoId*** *(int64):* ID khu vực cha, khu vực địa lý mà khách sạn tọa lạc
- ***hotelName*** *(object):* Tên khách sạn mà người dùng đã lưu trú và đánh giá
- ***stayDate*** *(object):* Ngày lưu trú của người dùng tại khách sạn
- ***tripType*** *(object):* Loại chuyến đi (ví dụ: du lịch cá nhân, công tác, kỳ nghỉ gia đình…)

Kết quả kiểm tra thông tin cơ bản:

- ➢ *Tổng số dòng:* 62,887 (số lượng quan sát)
- ➢ *Tổng số cột:* 14 (biến thuộc tính)
- ➢ Chứa giá trị null ở các thuộc tính: *username, userId, title*

```
<class 'pandas.core.frame.DataFrame'>
Index: 62887 entries, 654797953 to 788466559
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   language           62887 non-null  object
 1   rating             62887 non-null  int64
 2   additionalRatings  62887 non-null  object
 3   createdDate        62887 non-null  object
 4   helpfulVotes       62887 non-null  int64
 5   username           62424 non-null  object
 6   userId             62432 non-null  object
 7   title              62885 non-null  object
 8   text               62887 non-null  object
 9   locationId         62887 non-null  int64
 10  parentGeoId        62887 non-null  int64
 11  hotelName          62887 non-null  object
 12  stayDate           62887 non-null  object
 13  tripType           62887 non-null  object
dtypes: int64(4), object(10)
memory usage: 7.2+ MB
```

Hình 3.1. Mô tả thông tin cơ bản của bộ dữ liệu nguyên bản

3.2. Tiền Xử Lý Dữ Liệu

3.2.1. Tổng Quan Bộ Dữ Liệu Thô

Đầu tiên, trước khi bước vào giai đoạn tiền xử lý, nhóm tiến hành quan sát tổng quan bộ dữ liệu thô. Bộ dữ liệu được đọc từ tệp dữ liệu *'Reviews.csv'*, và sử dụng thư viện *Pandas* để tải lên.

Bên cạnh đó, nhóm cũng tạo một dataframe *'properties_check'* giúp kiểm tra các thông tin về các thuộc tính trong bộ dữ liệu, bao gồm:

- ➢ Số quan sát trong mỗi cột dữ liệu
- ➢ Số giá trị unique: Số lượng giá trị khác nhau trong từng cột
- ➢ Kiểu dữ liệu của từng cột

	Số quan sát	Số giá trị unique	Kiểu dữ liệu
id	62887	62887	int64
language	62887	14	object
rating	62887	5	int64
additionalRatings	62887	2546	object
createdDate	62887	4659	object
helpfulVotes	62887	52	int64
username	62424	54130	object
userId	62432	56836	object
title	62885	51910	object
text	62887	62863	object
locationId	62887	1497	int64
parentGeoId	62887	185	int64
hotelName	62887	1470	object
stayDate	62887	202	object
tripType	62887	6	object

Hình 3.2. Dataframe 'properties_check'

3.2.2. Chỉnh Dạng Dữ Liệu

Nhằm phù hợp với mục tiêu của bài toán là phân tích quan điểm đánh giá chất lượng dịch vụ khách sạn dựa trên nền tảng TripAdvisor, nhóm tiến hành chỉnh dạng dữ liệu nhằm loại bỏ các yếu tố không cần thiết và đảm bảo tính nhất quán của dữ liệu.

3.2.2.1. Lọc Các Đánh Giá Tiếng Việt

Đầu tiên, để đảm bảo dữ liệu đầu vào chỉ bao gồm các đánh giá bằng tiếng Việt, phù hợp với mục tiêu và phạm vi của bài toán. Nhóm sẽ tiến hành lọc văn bản dựa trên cột *'language'*. Cụ thể, chỉ những quan sát có giá trị *'language'* là *'vi'* (ký hiệu cho tiếng Việt) mới được giữ lại để phục vụ cho quá trình phân tích.

3.2.2.2. Xóa Các Cột Không Cần Thiết

Tiếp theo, nhóm tiến hành loại bỏ các cột không cung cấp thông tin quan trọng, chỉ giữ lại những cột có liên quan trực tiếp đến việc phân tích quan điểm người dùng, bao gồm các cột *'rating' (đánh giá), 'title' (tiêu đề của đánh giá)* và *'text' (nội dung đánh giá)*.

3.2.2.3. Gán Nhãn 'sentiment'

Nhóm thực hiện gán nhãn *'sentiment'* cho các đánh giá, dựa trên cột *'rating' (điểm đánh giá)*. Quy tắc gắn nhãn được thực hiện như sau:

- *'negative':* Nếu điểm *'rating'* nhỏ hơn 3 (Đánh giá tiêu cực)
- *'positive':* Nếu điểm *'rating'* lớn hơn 3 (Đánh giá tích cực)

➢ *'neutral':* Nếu điểm *'rating'* bằng 3 (Đánh giá trung lập)

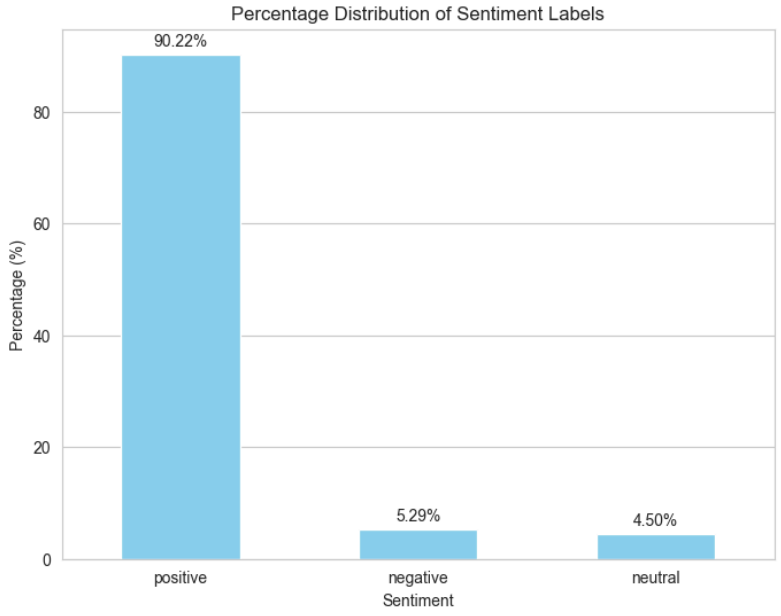
	rating	title	text	sentiment
0	4	Một đêm khó ngủ	- Nằm ở số 4 Tôn Đức Thắng (trong khu villa ca...	positive
4	4	Một công việc đang tiến triển - theo nghĩa đen.	Khách sạn đẹp, nhân viên và các cơ sở nhưng cô...	positive
7	1	không có một chút nào là thoải mái	tham quan để giới thiệu cho khu này thì bắt bu...	negative
8	5	Tuyệt vời và Nghỉ dưỡng số 1 Quảng Ngãi	Tôi có dịp công tác tại Quảng Ngãi và được giớ...	positive
9	5	Một nơi nghỉ dưỡng tuyệt vời.	Resort mới khai trương nên còn vắng vẻ, yên tĩ...	positive
...	...	...	...	...
62873	5	khách sạn mới chuẩn 5 sao, gần biển view đẹp	điều ấn tượng đầu tiên là khách sạn sát ngay b...	positive
62881	4	Khách sạn tốt duy nhất ở Tam Dương	Các khách sạn Putaleng là thương hiệu mới vì v...	positive
62882	5	Khách sạn tốt	Sau chuyến đi Sapa. . chúng tôi đã đi đến bình...	positive
62885	5	Rất tuyệt vời cho chuyến công tác tại Đức Hòa ...	Tôi có chuyến công tác tại khu công nghiệp Hải...	positive
62886	5	KHÁCH SẠN ĐẸP NHẤT, TIỆN NGHI SANG TRỌNG NHẤT ...	Tôi có chuyến công tác tại KCN Tân Đô, những l...	positive

15362 rows × 4 columns

Hình 3.3. Cột dữ liệu 'sentiment'

3.2.2.4. Giảm Kích Thước Dữ Liệu

Thực hiện kiểm tra tỷ lệ giữa các nhãn *'sentiment'*, nhóm nhận thấy bộ dữ liệu ban đầu có sự mất cân đối lớn giữa các nhãn, cụ thể *'positive'* chiếm hơn 90% tổng bộ dữ liệu, trong khi *'negative'* và *'neutral'* chỉ chiếm lần lượt khoảng 5% và 4%. *(Biểu đồ 3.1)*



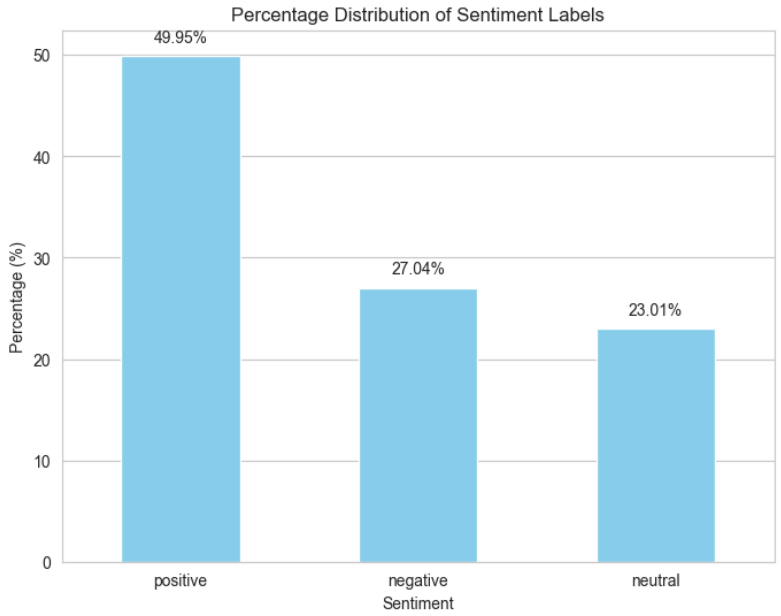
Biểu đồ 3.1. Biểu đồ tỷ lệ % của các nhãn trong cột 'sentiment' trước khi giảm kích thước

Nếu bộ dữ liệu không được cân bằng thì có thể gây ra hiện tượng mô hình thiên lệch khi huấn luyện, làm cho mô hình dự đoán bỏ qua các lớp còn lại, dự đoán chủ yếu là *'positive'*. Để giải quyết vấn đề này, nhóm quyết định tiến hành giảm số lượng nhãn

'positive' sao cho tỷ lệ giữa các nhãn *'positive', 'negative'* và *'neutral'* trở nên đồng đều hơn.

```
## Giảm kích thước số lượng nhãn 'positive'
# Lọc dữ liệu có nhãn 'positive' (lấy 1500 dòng)
positive_data = eda_data[eda_data['sentiment'] == 'positive'].head(1500)

# Lọc dữ liệu có nhãn 'negative' và 'neutral' (lấy tất cả)
negative_neutral_data = eda_data[eda_data['sentiment'].isin(['negative', 'neutral'])]

# Kết hợp dữ liệu lại (1500 dòng positive và tất cả dòng negative/neutral)
eda_data = pd.concat([positive_data, negative_neutral_data])
```

Sau khi thực hiện giảm kích thước, kiểm tra lại tỷ lệ phân bổ của các nhãn đã trở nên cân bằng hơn *(Biểu đồ 3.2)*, giúp cải thiện độ chính xác của các mô hình học máy khi phân loại các đánh giá, giảm thiểu hiện tượng bias.



Biểu đồ 3.2. Biểu đồ tỷ lệ % của các nhãn trong cột 'sentiment' sau khi giảm kích thước

```
<class 'pandas.core.frame.DataFrame'>
Index: 3003 entries, 0 to 62856
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   rating     3003 non-null   int64
 1   title      3003 non-null   object
 2   text       3003 non-null   object
 3   sentiment  3003 non-null   object
dtypes: int64(1), object(3)
memory usage: 117.3+ KB
```

Hình 3.4. Dữ liệu sau khi giảm kích thước

3.3. Text Mining

3.3.1. Gộp Văn Bản

Nhằm tối ưu hóa các thông tin văn bản trong dữ liệu, nhóm tiến hành gộp hai cột *'title'* và *'text'* thành một cột duy nhất, được đặt tên là *'review'*. Trong đó:

- ***title*** *(object):* Tiêu đề của đánh giá
- ***text*** *(object):* Nội dung đánh giá chi tiết do người dùng cung cấp
- ***review*** *(object):* Đánh giá của người dùng

Việc kết hợp thông tin từ tiêu đề và nội dung đánh giá chi tiết nhằm giúp văn bản có đầy đủ ngữ cảnh hơn, cung cấp chính xác dữ liệu hơn cho các mô hình học máy. Đồng thời, sau khi đã gộp dữ liệu, nhóm tiến hành loại bỏ hai cột cũ (*'title'* và *'text'*) để tránh dư thừa dữ liệu.

3.3.2. Xử Lý Văn Bản Thô

Sau khi gộp văn bản, nhóm tiếp tục sử dụng hàm *'clean_text'* để làm sạch dữ liệu trên cột *'review'* nhằm chuẩn hóa văn bản và loại bỏ các yếu tố gây nhiễu, bao gồm:

- Xóa dấu câu và ký tự đặc biệt như "!", "@", "#", ",", ".", "?", không mang giá trị ý nghĩa trong phân tích quan điểm đánh giá của khách hàng nên sẽ được loại bỏ.
- Xóa khoảng trắng thừa để đảm bảo văn bản không có khoảng trắng dư thừa giữa các từ.
- Chuyển văn bản thành dạng chữ thường *(lowercase)* để đồng nhất dữ liệu và giảm thiểu sự sai lệch khi phân tích từ.

3.3.3. Xử Lý Từ Không Có Trong Tiếng Việt

Sau khi văn bản được làm sạch, nhằm đảm bảo tính chính xác và phù hợp với ngữ cảnh tiếng Việt, nhóm tiến hành lọc các từ không nằm trong từ điển tiếng Việt bằng hàm *'filter_words'*. Trong đó, nhóm sử dụng hàm *'word_tokenize'* từ thư viện *nltk* để tách văn bản thành các từ riêng lẻ. Nhóm sử dụng thêm mô hình *Punkt Sentence Tokenizer* để đảm bảo hàm *'word_tokenize'* hoạt động chính xác.

```
import nltk
nltk.download('punkt_tab')
```

Mô hình *Punkt* là công cụ đã được huấn luyện sẵn từ thư viện *nltk*, giúp chia nhỏ văn bản thành các từ hoặc câu dựa trên dấu câu và khoảng trắng. Việc sử dụng mô hình này giúp quy trình tokenize văn bản chính xác và hiệu quả hơn. Sau khi đã tách văn bản thành công, nhóm tiến hành so sánh với danh sách từ điển chuẩn trong file *TuDon.txt*

3.3.4. Sửa Lỗi Chính Tả, Tách Từ Và Loại Bỏ Stopwords

Tiếp theo, nhóm tiến hành sử dụng hàm *'process_text'* nhằm sửa lỗi chính tả, tokenize văn bản và loại bỏ stopwords trên cột văn bản *'cleaned review'*.

- ➢ Sử dụng *ViTokenizer* từ thư viện *pyvi* để thực hiện và tokenize văn bản.
- ➢ Tích hợp sử dụng thêm danh sách stopwords tiếng Việt từ file *vietnamese-stopwords.txt* để loại bỏ các từ không cần thiết (như "và", "là", "của", "nhưng"....), không mang nhiều thông tin quan trọng, khỏi văn bản đã được tokenized.

```
# Kết hợp cột 'title' và 'text' thành một chuỗi duy nhất, sau đó làm sạch
văn bản bằng cách sử dụng hàm clean_text
eda_data['review'] = (eda_data['title'] + ' ' +
eda_data['text']).apply(clean_text)

# Áp dụng hàm filter_words để lọc các từ trong cột 'review' chỉ giữ lại
các từ có trong danh sách cho phép
eda_data['filtered_review'] = eda_data['review'].apply(filter_words)

# Áp dụng hàm process_text để token hóa và loại bỏ các stopwords từ cột
'filtered_review'
eda_data['processed_review'] =
eda_data['filtered_review'].apply(process_text)
```

	sentiment	review
0	positive	đêm ngủ nằm tôn đức thắng khu cao_cấp đối_diện...
4	positive	công_việc tiến_triển nghĩa_đen khách_sạn đẹp n...
8	positive	tuyệt_vời nghỉ_dưỡng quảng_ngãi dịp công_tác q...
9	positive	nghỉ_dưỡng tuyệt_vời khai_trương vắng_vẻ yên_t...
10	positive	địa_điểm lý_tưởng quảng_ngãi vô_tình ghé khách...
...	...	...
61927	neutral	ồn_ào gia_đình phòng thời_gian chúng_tôi tổ_ch...
62502	neutral	chỗ ok chỗ sạch_sẽ hòa không_khí tuy_nhiên chỗ...
62516	negative	kiểm_tra đi chúng_tôi du_lịch đồng_bằng sông c...
62611	negative	tiền vì_vậy giới_thiệu kiểm_tra đối_tác sống h...
62856	neutral	thức_ăn phòng đào_tạo kém dịch_vụ khách_hàng t...

3003 rows × 2 columns

Hình 3.5. Dữ liệu 'eda_data' sau khi xử lý

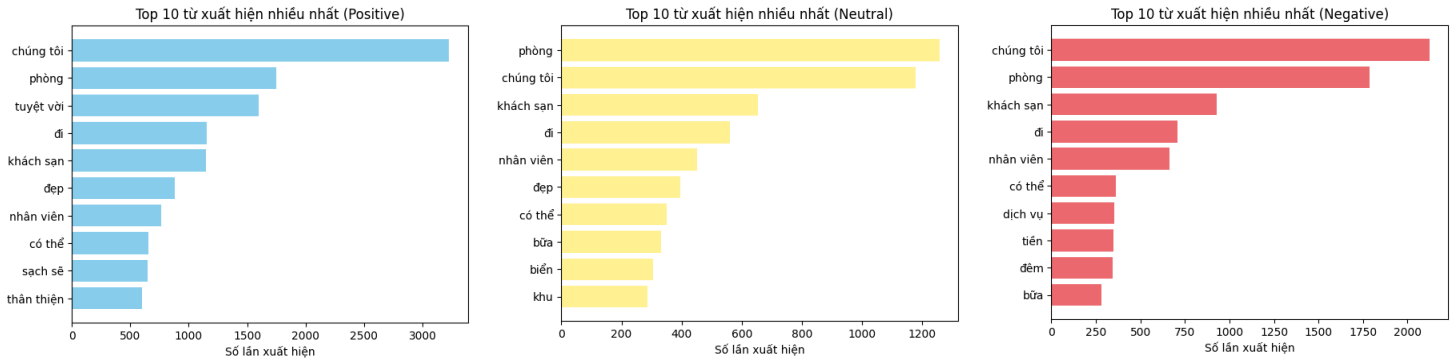
3.4. Trực Quan Hóa Dữ Liệu

Trước khi tiến hành huấn luyện mô hình, nhóm thực hiện trực quan hóa để quan sát các đặc trưng của bộ dữ liệu.

Đầu tiên, nhóm tiến hành sao chép dữ liệu từ *'eda_data'* sang *'visual_data'* để tạo ra một bản sao độc lập của *'eda_data'*, cho phép thực hiện các thao tác biến đổi và trực quan hóa mà không làm ảnh hưởng đến dữ liệu gốc.

```
visual_data = eda_data.copy()
```

Tiếp đến, nhóm tạo hàm *'get_top_words'* để đếm tần suất xuất hiện của các từ trong dữ liệu, phân loại theo các nhãn *'positive', 'negative'* và *'neutral'*. Hàm này được sử dụng để hỗ trợ tạo ra các biểu đồ trực quan, thể hiện top các từ phổ biến trong từng nhóm nhãn.



Biểu đồ 3.3. Biểu đồ thể hiện top 10 từ xuất hiện nhiều nhất theo các nhãn

Dựa trên *Biểu đồ 3.3,* có thể thấy các đánh giá tích cực tập trung vào các tính từ khen ngợi như *'tuyệt vời', 'đẹp'* và *'sạch sẽ'*. Trong khi đó, đánh giá tiêu cực thường liên quan đến vấn đề trải nghiệm dịch vụ của khách hàng như *'dịch vụ', 'tiền'* hoặc *'đêm'*. Nhóm đánh giá trung lập chủ yếu mô tả trải nghiệm chung, ít biểu lộ cảm xúc mạnh mẽ.

Tiếp theo, nhóm cũng tạo WordCloud để trực quan hóa tần suất xuất hiện của các từ trong toàn bộ dữ liệu đánh giá khách sạn của khách hàng. Với kích thước của mỗi từ trong WordCloud thể hiện mức độ phổ biến của từ đó trong tập dữ liệu.

Biểu đồ 3.4. WordCloud cho toàn bộ dữ liệu

3.5. Phân Chia Dữ Liệu

Quá trình phân chia tập dữ liệu được tiến hành như sau:

Đầu tiên, nhóm tiến hành sao chép dữ liệu từ *'eda_data'* sang *'model_data'* để tạo ra một bản sao độc lập của *'model_data'* , cho phép thực hiện các thao tác biến đổi để chuẩn bị cho các mô hình máy học mà không làm ảnh hưởng đến dữ liệu gốc.

Dữ liệu được lấy từ hai cột trong tập dữ liệu đã được tiền xử lý, trong đó cột *'cleaned review'* chứa các bình luận đã được làm sạch và cột *'sentiment'* chứa nhãn về quan điểm của các bình luận. Mục tiêu của nghiên cứu là phân loại các bình luận này thành các lớp quan điểm tương ứng.

Tiếp theo, tập dữ liệu được chia thành hai phần: một phần được sử dụng để huấn luyện mô hình và phần còn lại được sử dụng để kiểm tra độ chính xác của mô hình. Cụ thể, 80% dữ liệu được dùng để huấn luyện mô hình, trong khi 20% còn lại để đánh giá hiệu suất của mô hình sau khi huấn luyện.

Sau khi chia dữ liệu thành các tập huấn luyện và kiểm tra, các văn bản trong tập huấn luyện và tập kiểm tra được chuyển đổi thành các đặc trưng số bằng phương pháp **TF-IDF *(Term Frequency-Inverse Document Frequency)***. Phương pháp này giúp biến đổi các đánh giá văn bản thành các vector số mà mô hình có thể xử lý được. Trong đó, mỗi từ trong văn bản được gán một giá trị dựa trên tần suất xuất hiện của nó trong văn bản và độ quan trọng của từ đó trong toàn bộ tập dữ liệu. Đặc biệt, tham số ***ngram_range = (1, 2)*** được sử dụng để bao gồm cả các đơn từ (unigram) và các cặp từ (bigram), giúp mô hình nắm bắt được cả các từ đơn lẻ và các kết hợp từ có ý nghĩa đặc biệt trong các đánh giá.

3.6. Huấn Luyện Mô Hình

Sau khi thực hiện tiền xử lý dữ liệu đầu vào cho các mô hình học máy, nhóm tiến hành huấn luyện và đánh giá các mô hình. Nhóm sử dụng hai hàm *'plot_confusion_matrix'* và *'train_and_evaluate_model'* để hỗ trợ trong việc đánh giá hiệu suất, giúp lựa chọn được các tham số tối ưu cho các mô hình học máy.

- Hàm *'plot_confusion_matrix'* vẽ ma trận nhầm lẫn bằng biểu đồ heatmap, để thể hiện tỷ lệ phần trăm nhầm lẫn giữa các lớp dự đoán và lớp thực tế.

➢ Hàm *'train_and_evaluate_model'* thực hiện việc huấn luyện mô hình học máy với các tham số tối ưu thông qua *GridSearchCV,* giúp tự động xác định các tham số tối ưu cho mô hình trong phạm vi định nghĩa cho trước.

Trong bước tiếp theo, nhóm tiến hành huấn luyện và đánh giá ba mô hình học máy phổ biến bao gồm: *Multinomial Logistic Regression, Multinomial Naive Bayes, Support Vector Machine* và một mô hình học sâu *Feedforward Neural Network.*

3.6.1. Multinomial Logistic Regression

Với mô hình *Multinomial Logistic Regression,* nhóm thực hiện tìm kiếm các thông số tối ưu thông qua *GridSearchCV* với các tham số bao gồm:

➢ *c (hệ số điều chỉnh độ phức tạp của mô hình):* được thử nghiệm ở các mức 0.1, 1 và 10.

➢ *solver (giải pháp tối ưu hóa):* được xác định là *'lbfgs'*, là thuật toán tối ưu hóa sử dụng *Quasi-Newton* được sử dụng để xử lý các bài toán phân loại đa lớp trong mô hình hồi quy Logistic.

➢ *multi_class:['multinomial']* giúp mô hình xác định xử lý bài toán phân loại đa lớp.

➢ *max_iter:[100,200]* nhằm giới hạn số lần lặp tối đa trong quá trình huấn luyện mô hình.

```
print("Multinomial Logistic Regression:")

# Các tham số riêng biệt
logistic_params = [{'C': [0.1, 1, 10], 'solver': ['lbfgs'], 'multi_class':
['multinomial'], 'max_iter': [100, 200]}]

best_multi_logistic    =    train_and_evaluate_model(LogisticRegression(),
logistic_params, X_train_tfidf, y_train, X_test_tfidf, y_test)
```

3.6.2. Multinomial Naive Bayes

Tương tự với việc huấn luyện mô hình Multinomial Logistic Regression, nhóm sử dụng *nb_params* để tối ưu hóa mô hình thông qua *GridSearchCV* với các tham số, bao gồm:

➢ *alpha:[0.1, 0.5, 1.0]* là tham số điều chỉnh Laplace

➢ *fit_prior:[True, False]* dùng để xác định liệu mô hình có tính toán các xác suất ban đầu từ dữ liệu huấn luyện hay không. Các xác suất này dùng trong quá trình phân loại để xác định khả năng xuất hiện của mỗi lớp dữ liệu.

```
# Multinomial Naive Bayes
print("Multinomial Naive Bayes:")
nb_params = {'alpha': [0.1, 0.5, 1.0], 'fit_prior': [True, False]}
best_nb = train_and_evaluate_model(MultinomialNB(), nb_params,
X_train_tfidf, y_train, X_test_tfidf, y_test)
```

3.6.3. Support Vector Machine (SVM)

Với mô hình SVM, nhóm sử dụng *svm_params* để tối ưu hóa mô hình thông qua *GridSearchCV* với các tham số như sau:

➢ *c:[0.1, 1, 10]* là tham số giúp điều chỉnh độ phức tạp của mô hình SVM.

➢ *kernel:['linear', 'rbf']* là tham số giúp xác định loại kernel function được sử dụng trong SVM:

- *'linear'* tương ứng với *linear kernel*, được sử dụng phù hợp với các bài toán phân loại tuyến tính.
- *'rbf'* tương ứng với *Radial Basis Function*, giúp mô hình xử lý các bài toán phân loại phi tuyến.

```
# SVM
print("Support Vector Machine:")
svm_params = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}
best_svm = train_and_evaluate_model(SVC(), svm_params, X_train_tfidf,
y_train, X_test_tfidf, y_test)
```

3.6.4. Feedforward Neural Network

Mô hình Feedforward Neural Network (FNN) được sử dụng dưa trên thư viện TensorFlow/Keras để xử lý bài toán phân loại. Đầu tiên, nhóm khởi tạo mô hình bao gồm ba lớp dày đặc (Dense), với các kích thước 128, 32, và số lượng lớp đầu ra tương ứng với nhãn phân loại là 3. Hàm kích hoạt relu được sử dụng cho các lớp ẩn và softmax cho lớp đầu ra.

```
# Tạo mô hình FNN (Feedforward Neural Network)
model_fnn = Sequential([
    Input(shape=(X_train_tfidf_dense.shape[1],)),  # Sử dụng Input
    Dense(128, activation='relu'),
    Dense(32, activation='relu'),
     Dense(len(label_encoder.classes_), activation='softmax')   # Số lượng
nhãn
])
```

Mô hình được huấn luyện qua 20 epoch với dữ liệu huấn luyện và kiểm tra, batch_size là 32.

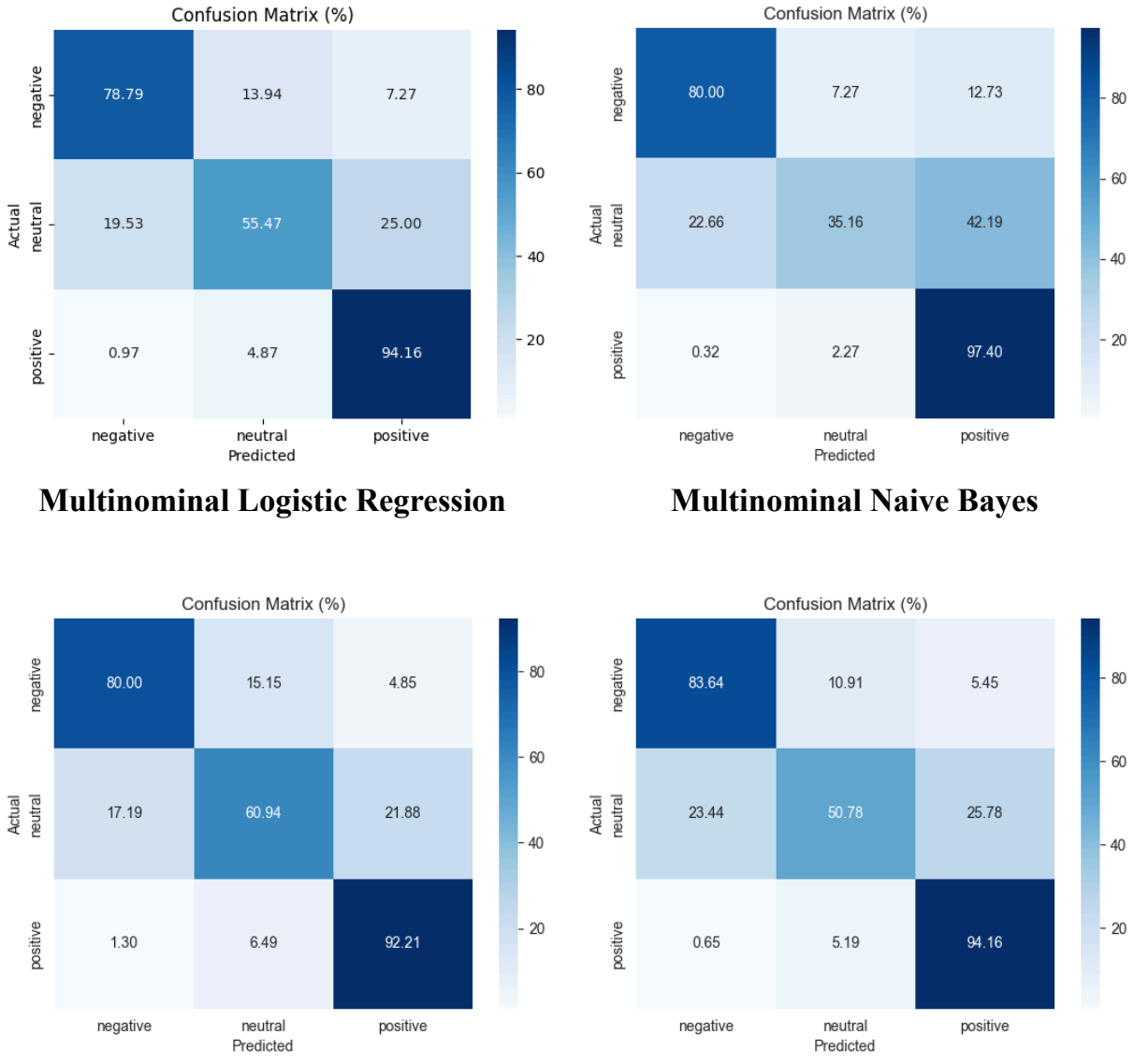
```
# Huấn luyện mô hình
history = model_fnn.fit(
    X_train_tfidf_dense,
    y_train_encoded,
    validation_data=(X_test_tfidf_dense, y_test_encoded),
    epochs=20,
    batch_size=32
)

# Dự đoán trên dữ liệu kiểm tra
y_pred = model_fnn.predict(X_test_tfidf_dense)
y_pred_labels = np.argmax(y_pred, axis=1)
```

CHƯƠNG 4 - KẾT LUẬN

4.1. Các Kết Quả Đạt Được

Khi quan sát các ma trận nhầm lẫn, có thể nhận thấy các mô hình cho kết quả phân loại khá tốt. Trong đó, **FNN và SVM** có hiệu suất tổng thể cao hơn, đặc biệt với lớp Negative và Positive. Ngược lại, Naive Bayes cho thấy kết quả phân loại kém nhất trong cả 4 mô hình.

Tuy nhiên, ở cả 4 mô hình, kết quả phân lớp của lớp Neutral đều là thách thức lớn vì kết quả phân loại khá thấp. Điều này có thể đến từ việc Neutral thường chứa thông tin khó phân biệt, không nghiêng rõ ràng về phía tích cực hoặc tiêu cực.



Biểu đồ 3.5. Ma trận nhầm lẫn thể hiện kết quả dự đoán của các mô hình

(Nguồn: Nhóm tác giả)

4.1.1. Mô Hình Multinominal Logistic Regression

Dựa trên kết quả đánh giá, mô hình đạt **độ chính xác 82%** với hiệu suất phân loại tốt ở lớp *Negative* và *Positive*. Cụ thể, lớp *Negative* có **F1-score 0.80**, trong khi lớp *Positive* nổi bật với **F1-score 0.90**, cho thấy khả năng dự đoán chính xác cao. Tuy nhiên, lớp *Neutral* gặp khó khăn lớn với **F1-score chỉ 0.60**, do precision và recall thấp lần lượt là 0.65 và 0.55. Độ chênh lệch này ảnh hưởng đến kết quả chung, cho thấy cần cải thiện việc phân loại lớp Neutral thông qua các phương pháp xử lý dữ liệu hoặc điều chỉnh trọng số.

	precision	recall	f1-score	support
negative	0.82	0.79	0.80	165
neutral	0.65	0.55	0.60	128
positive	0.87	0.94	0.90	308
accuracy			0.82	601
macro avg	0.78	0.76	0.77	601
weighted avg	0.81	0.82	0.81	601

4.1.2. Mô Hình Multinominal Naive Bayes

Dựa trên kết quả đánh giá, mô hình **Multinomial Naive Bayes** đạt **độ chính xác 79%**. Mô hình thể hiện tốt ở lớp *Negative* với **F1-score 0.81** và lớp *Positive* với **F1-score 0.88** cùng recall 0.97. Tuy nhiên, lớp *Neutral* có hiệu suất kém với **F1-score 0.47** và recall chỉ 0.35, cho thấy mô hình gặp khó khăn trong việc phân biệt dữ liệu trung tính. Điều này có thể do tính chất đơn giản của Naive Bayes và sự mất cân bằng dữ liệu, dẫn đến kết quả không cao cho lớp Neutral.

	precision	recall	f1-score	support
negative	0.81	0.80	0.81	165
neutral	0.70	0.35	0.47	128
positive	0.80	0.97	0.88	308
accuracy			0.79	601
macro avg	0.77	0.71	0.72	601

weighted avg	0.78	0.79	0.77	601

4.1.3. Mô Hình Support Vector Machine

Với tham số tối ưu C = 1 và kernel = linear, kết quả đánh giá của mô hình SVM đạt độ chính xác 82%. Lớp Negative đạt F1-score 0.82, cho thấy khả năng dự đoán tốt. Lớp Positive vượt trội với F1-score 0.90 và recall 0.92, phản ánh hiệu suất cao. Tuy nhiên, lớp Neutral chỉ đạt F1-score 0.62 với recall 0.61, cho thấy vẫn còn thách thức trong việc phân loại dữ liệu trung tính. Tổng thể, SVM hoạt động ổn định và cân bằng trên các lớp.

	precision	recall	f1-score	support
negative	0.84	0.80	0.82	165
neutral	0.63	0.61	0.62	128
positive	0.89	0.92	0.90	308
accuracy			0.82	601
macro avg	0.79	0.78	0.78	601
weighted avg	0.82	0.82	0.82	601

4.1.4. Mô Hình Feedforward Neural Network

	precision	recall	f1-score	support
negative	0.84	0.84	0.84	165
neutral	0.66	0.59	0.63	128
positive	0.89	0.93	0.91	308
accuracy			0.83	601
macro avg	0.80	0.79	0.79	601
weighted avg	0.83	0.83	0.83	601

Dựa trên kết quả đánh giá của mô hình **FNN (Feedforward Neural Network)**, độ chính xác đạt **83%**, cho thấy hiệu suất tổng thể khá tốt. Cụ thể:

- ➢ **Lớp Negative**: **F1-score 0.84**, phản ánh khả năng phân loại chính xác và cân bằng giữa precision và recall.
- ➢ **Lớp Positive**: **F1-score 0.91** với recall 0.93, thể hiện hiệu suất xuất sắc trên lớp tích cực.
- ➢ **Lớp Neutral**: **F1-score 0.63** và recall 0.59, cho thấy mô hình còn gặp khó khăn trong việc nhận diện dữ liệu trung tính.

Kết quả này cho thấy **FNN** hoạt động tốt nhưng vẫn cần cải thiện khả năng phân loại lớp *Neutral*, có thể thông qua cân bằng dữ liệu hoặc điều chỉnh trọng số lớp.

4.2. Kết luận

Trong 4 mô hình (Multinominal Logistic Regression, Multinominal Naive Bayes, SVM, FNN), **FNN (Feedforward Neural Network)** thể hiện là mô hình tốt nhất với độ chính xác cao nhất là 83% và F1-score cao nhất cho cả lớp *Positive* và *Negative*. Mô hình SVM cũng cho kết quả mạnh mẽ với độ chính xác 82%, nhưng FNN vượt trội hơn về khả năng cân bằng phân loại giữa các lớp, đặc biệt là trong việc xử lý lớp *Neutral*.

4.3. Những Hạn Chế Và Hướng Pháp Triển

4.3.1. Hạn Chế

Đề tài gặp khó khăn với số lượng dữ liệu hạn chế, đặc biệt là đối với lớp Neutral, dẫn đến hiệu suất phân loại không cao. Bên cạnh đó, sự mất cân bằng dữ liệu giữa các lớp cũng ảnh hưởng đến kết quả tổng thể.

4.3.2. Hướng Phát Triển

Để phát triển đề tài, cần tăng cường số lượng dữ liệu trung tính và áp dụng các phương pháp cân bằng dữ liệu như oversampling hoặc undersampling. Đồng thời, điều chỉnh các tham số của mô hình hoặc sử dụng các phương pháp học sâu có thể giúp cải thiện hiệu quả phân loại.

Bên cạnh đó, có thể bổ sung thông tin chi tiết như các thuộc tính khác và phân nhóm người dùng. Điều này sẽ giúp phân tích sâu hơn về từng nhóm khách hàng và các yếu tố ảnh hưởng đến đánh giá chất lượng dịch vụ, từ đó tạo ra một hệ thống đánh giá đa chiều để hiểu rõ hơn về nhu cầu và trải nghiệm của từng đối tượng khách hàng.

TÀI LIỆU THAM KHẢO

NIAID Data Discovery Portal. (2024). NIAID Data Discovery Portal. https://data.niaid.nih.gov/resources?id=zenodo_7967493

Tổng cục Du lịch Việt Nam. (2023). Báo cáo du lịch Việt Nam 2023

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences, 12(16), 4102-4107.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised learning. International journal of computer science, 1(2), 111-117.

Tamilselvi, R., Sivasakthi, B., & Kavitha, R. (2015). An efficient preprocessing and postprocessing techniques in data mining. Int. J. Res. Comput. Appl. Robot, 3(4), 80-85.

Gaikwad, S. V., Chaugule, A., & Patil, P. (2014). Text mining methods and techniques. International Journal of Computer Applications, 85(17).

Hearst, M. (2003). What is text mining. SIMS, UC Berkeley, 5.

Arsyah, U. I., Pratiwi, M., & Muhammad, A. (2024). Twitter Sentiment Analysis of Public Space Opinions using SVM and TF-IDF Methods. Indonesian Journal of Computer Science, 13(1).

Qutab, I., Malik, K. I., & Arooj, H. (2022). Sentiment classification using multinomial logistic regression on Roman Urdu text. Int. J. Innov. Sci. Technol, 4(2), 323-335.

Vryniotis, V. (2013). Machine Learning Tutorial: The Multinomial Logistic Regression (Softmax Regression).

McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In AAAI-98 workshop on learning for text categorization (pp. 41-48).

Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. Pearson.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (1997). Estimating the Support of a High-Dimensional Distribution. Neural Computation.

Joachims, T. (1998). Text classification with Support Vector Machines: Learning with many relevant features. Proceedings of the European Conference on Machine Learning (ECML).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing.

GeeksforGeeks. (2024, June 20). Feedforward neural network. Retrieved December 16, 2024, from https://www.geeksforgeeks.org/feedforward-neural-network/

PHỤ LỤC

1. Phụ Lục 1: Source Code

Mã nguồn của nhóm: https://github.com/phiyenng/NLP-HotelReviewsSentimentAnalysis

2. Phụ Lục 2: Phân Công Công Việc

Nhiệm vụ	Thành viên	Mức độ hoàn thành
- Tiền xử lí dữ liệu - Trực quan hóa dữ liệu - Mô hình học sâu FNN - Kết luận	Đỗ Thị Mỹ Khánh	100%
- Tổng quan đề tài - Mô hình học máy Multinomial Naive Bayes - Slide	Trần Nguyễn Thảo Nguyên	100%
- Lý thuyết TF-IDF Vectorizer - Phân chia dữ liệu - Huấn luyện mô hình - Mô hình học máy Support Vector Machine	Nguyễn Hữu Thanh	100%
- Lý thuyết Tiền xử lí dữ liệu - Tổng quan bộ dữ liệu - Tiền xử lí dữ liệu - Trực quan hóa dữ liệu - Mô hình học máy Multinomial Logistic Regression	Nguyễn Vân Phi Yến	100%