TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

Phân loại review tích cực và tiêu cực của khách hàng về các quán ăn trên nền tảng Foody

Nhóm	6
Họ Và Tên Sinh Viên	Lớp Học Phần
Nguyễn Ngọc Hải	2011A
Trương Hoàng Nhật Huy	
Dương Mạnh Quân	

ĐÀ NẴNG, 06/2023

TÓM TẮT

Trong tiểu luận này, nhóm sẽ giải quyết các vấn đề liên quan đến xử lý ngôn ngữ tự nhiên. Cụ thể, đề tài là phân loại các bình luận tiêu cực tích cực và tiêu cực trên nền tảng review quán án Foody.

Vấn đề cần giải quyết là làm sao để có thể phân loại hàng nghìn bình luận một cách hiệu quả với độ chính xác cao.

Phương pháp giải quyết: thu thập các bình luận trên nền tảng Foody rồi gán nhãn dữ liệu dựa trên điểm số của bình luận. Sau đó tiến hành các bước để làm sạch dữ liệu. Cuối cùng dùng các công cụ để vector hóa dữ liệu dưới dạng số để làm input cho các mô hình học máy như: Naive Bayes và Support Vector Machine.

Cuối cùng, nhóm đã đạt được kết quả khả quan trong việc huấn luyện mô hình với độ chính xác cao và chi phí thấp.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	**Các nhiệm vụ**	**Tự đánh giá**
Nguyễn Ngọc Hải	- Tiền xử lý dữ liệu - Trực quan hóa dữ liệu - Huấn luyện mô hình Naive Bayes	- Đã hoàn thành - Đã hoàn thành - Đã hoàn thành
Trương Hoàng Nhật Huy	- Huấn luyện mô hình SVM - Viết quyển báo cáo	- Đã hoàn thành - Đã hoàn thành
Dương Mạnh Quân	- Viết module thu thập dữ liệu - Thu thập dữ liệu - Làm slide báo cáo	- Đã hoàn thành - Đã hoàn thành - Đã hoàn thành

Mục lục

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG BIỂU

1. Giới thiệu

1.1. Lý do chọn đề tài

Foody là một nền tảng chia sẻ các địa điểm ăn uống. Được xây dựng từ năm 2012, Foody đã trở thành một cộng đồng tin cậy, chứa hàng trăm ngàn địa điểm ăn uống và hàng trăm ngàn bình luận, hình ảnh. Đối với một daonh nghiệp trong mảng thực phẩm, Foody là một nguồn dữ liệu quý giá để nắm rõ các đặc điểm, nhu cầu của khách hàng.

Do đó, nhóm chúng em đã chọn đề tài "Phân loại review tích cực và tiêu cực của khách hàng của các quán ăn trên nền tảng Foody".

1.2. Giới thiệu bài toán

Xây dựng mô hình phân loại đánh giá tích cực, tiêu cực của khách hàng trên nền tảng Foody.

- Đầu vào là các bình luận đánh giá của khách hàng trên một trang Foody.
- Đầu ra là nhãn tích cực hay tiêu cực của các đánh giá đó.

Để giải quyết bài toán này, nhóm sẽ áp dụng các phương pháp xử lý văn bản, thử nghiệm các mô hình học máy để huấn luyện được một mô hình tối ưu nhất.

2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

Nhóm đã thu thập các bình luận, đánh giá trên nhiều trang Foody, sử dụng công cụ Selenium.

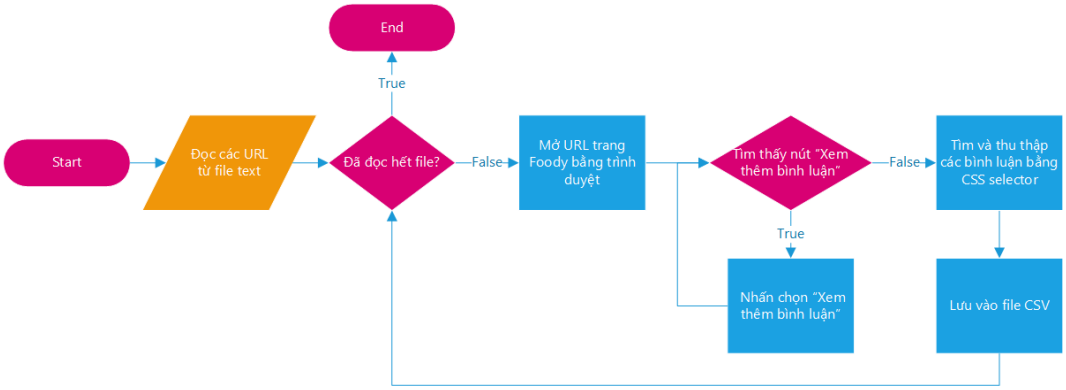
Selenium là một công cụ kiểm thử trang web cho các lập trình viên. Trong Python, Selenium cung cấp bộ API đơn giản, cho phép tương tác với trang web, qua đó dễ dàng thu thập dữ liệu văn bản.

Nhóm đã viết script giúp thu thập dữ liệu từ các trang Foody:

- Đầu vào là file text chứa các URL của các trang Foody.
- Đầu ra là file CSV chứa các bình luận, điểm đánh giá.

Hàm thực thi cần các tham số là:

- browser_option: chọn trình duyệt để mở URL
- url_file: file text chứa các URL của các trang Foody.
- file_name_to_save: tên file CSV chứa các bình luận, điểm đánh giá.
- n_sample: số lượng mẫu dữ liệu cần thu thập.



Hình 1. Sơ đồ script thu thập dữ liệu

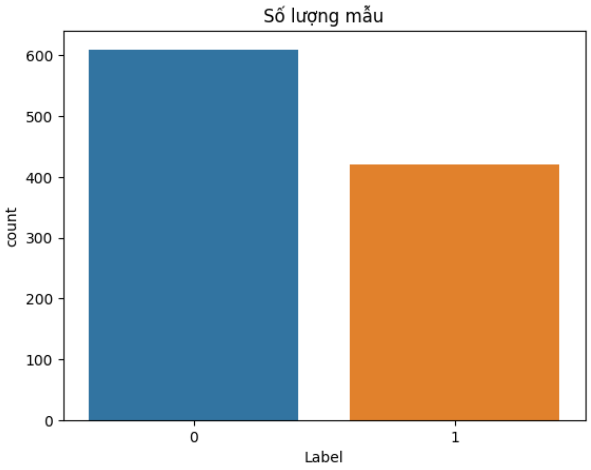
2.2. Mô tả dữ liệu

Bộ dữ liệu tự thu thập bao gồm Big Data chứa 10000 mẫu và Small Data chứa 1000 mẫu. Mỗi mẫu dữ liệu bao gồm:

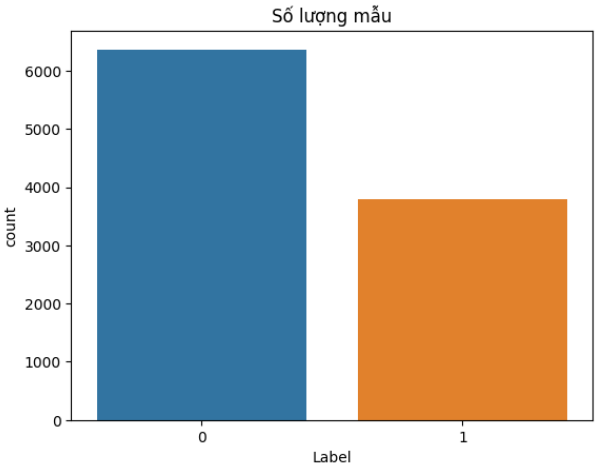
- Bình luận đánh giá là chuỗi kí tự tiếng Việt.
- Điểm đánh giá là số thực trong khoảng [0;10]

Sau khi thu thập được dữ liệu thô, sử dụng các kĩ thuật tiền xử lý dữ liệu văn bản tiếng Việt:

- Chuyển ký tự hoa sang ký tự thường.
- Xóa ký tự đặc biệt & emoji.
- Xác định từ ghép.
- Xóa các từ dừng.



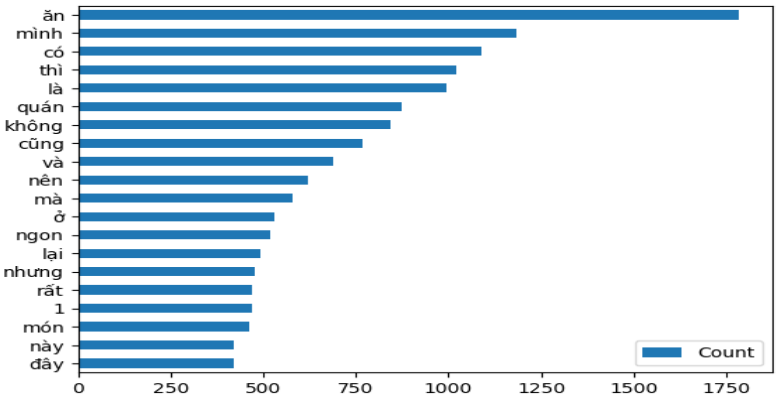
Hình 2. Sơ đồ số lượng mẫu dữ liệu tích cực & tiêu cực trên Small Data



Hình 3. Sơ đồ số lượng mẫu dữ liệu tích cực & tiêu cực trên Big Data

2.2.1. Trước khi tiền xử lý:

Hình 4. Word cloud của Small Data trước khi tiền xử lý

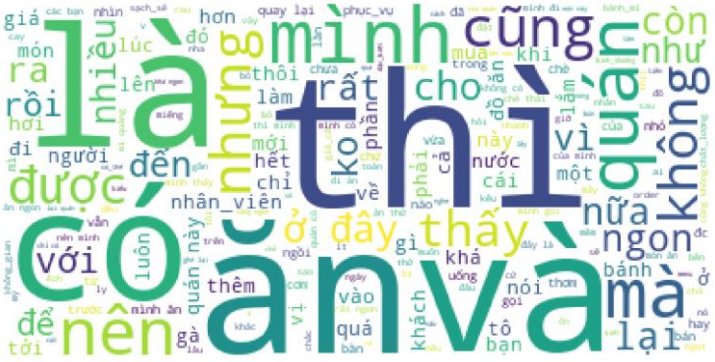


Hình 5. Histogram các từ xuất hiện nhiều nhất trong Small Data trước khi tiền xử lý

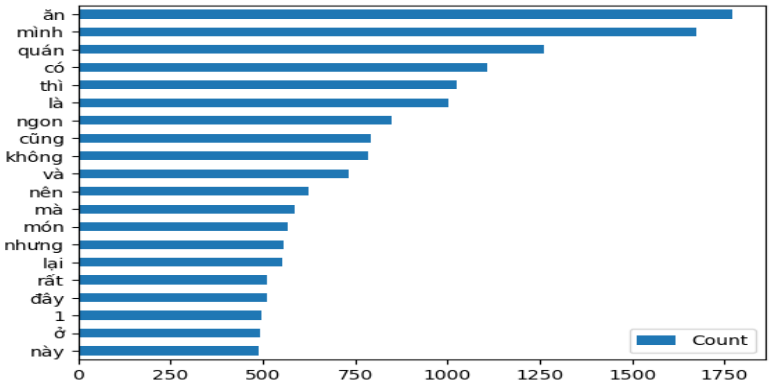
Hình 6. Word cloud của các bình luận tiêu cực trong Small Data trước khi tiền xử lý

Hình 7. Word cloud của các bình luận tích cực trong Small Data trước khi tiền xử lý

2.2.2. Sau khi tiền xử lý:



Hình 8. Word cloud của Small Data sau khi tiền xử lý



Hình 9. Histogram các từ xuất hiện nhiều nhất trong Small Data trước khi tiền xử lý

Hình 10. Word cloud của các bình luận tiêu cực trong Small Data sau khi tiền xử lý

Hình 11. Word cloud của các bình luận tích cực trong Small Data sau khi tiền xử lý

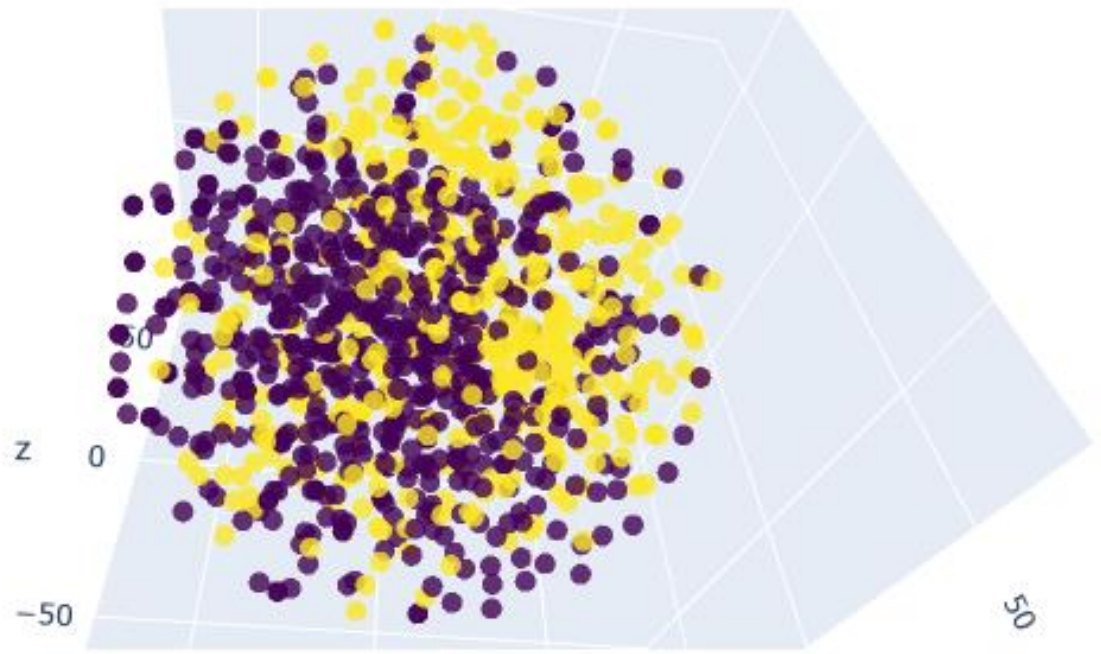
3. Trích xuất đặc trưng

Để trích xuất đặc trưng từ dữ liệu, nhóm đã áp dụng các kĩ thuật mã hóa dữ liệu văn bản thành vector: Bag of Words và Term Frequency–Inverse Document Frequency.

3.1. Bag of Words

Bag of Words là kĩ thuật vector hóa dữ liệu văn bản bằng cách sử dụng "túi đựng từ" gồm tập hợp tất cả các từ cần quan tâm, và đếm tần số xuất hiện của mỗi từ.

Kĩ thuật này không quan tâm đến ngữ pháp cũng như trật tự từ, và vector trích xuất được là đồng đều với mọi văn bản.



Hình 12. Trực quan hóa dữ liệu sau khi vector hóa bằng kĩ thuật Bag of Words và giảm chiều dữ liệu

3.2. Term Frequency–Inverse Document Frequency

Term Frequency–Inverse Document Frequency, hay TF-IDF, là kĩ thuật vector hóa dữ liệu. Sử dụng cùng cách vector hóa như Bag of Words, nhưng mỗi từ sẽ có trọng số đánh giá mức độ quan trọng của mỗi từ.

Mức độ quan trọng của một từ trong văn bản sẽ tỉ lệ thuận với tần suất xuất hiện trong văn bản, và tỉ lệ nghịch với tần suất các văn bản chứa từ đó.

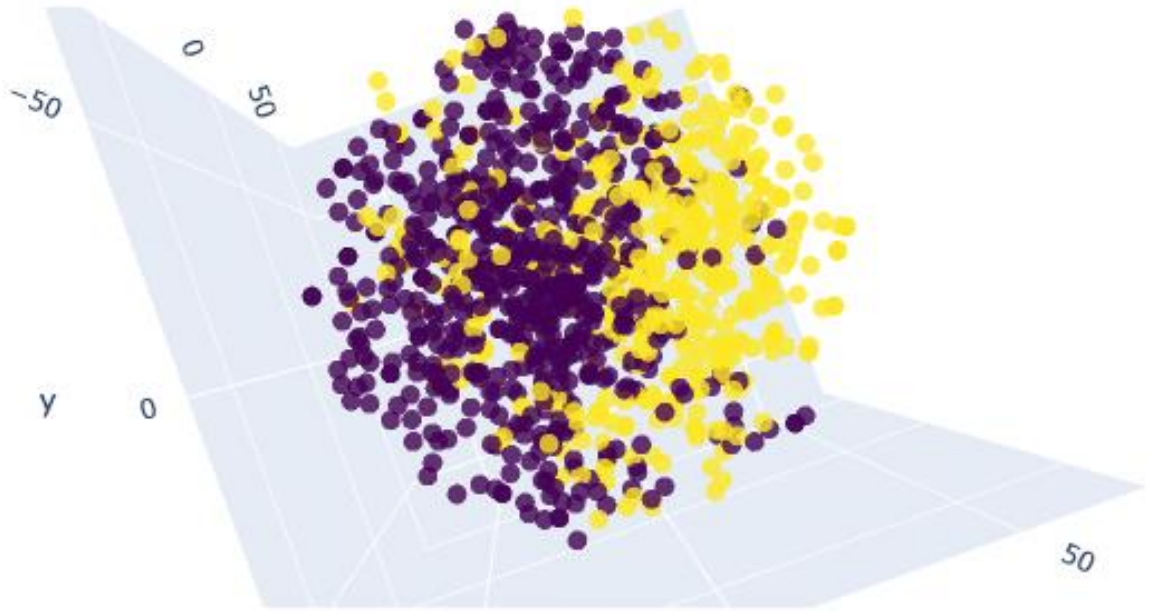
Trọng số của mỗi từ được tính theo công thức:

$$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i})$$

Trong đó:

- $tf_{i,j}$ là số lần từ i xuất hiện trong văn bản j.
- df_i là số lượng văn bản chứa từ i.
- N là số lượng văn bản trong tập dữ liệu.

Nhờ vào trọng số của từ mà các từ ít xuất hiện nhưng quan trọng với văn bản sẽ ảnh hưởng tốt hơn đến kết quả phân loại cuối cùng.



Hình 13. Trực quan hóa dữ liệu sau khi vector hóa bằng kĩ thuật Bag of Words và giảm chiều dữ liệu

4. Mô hình hóa dữ liệu

Sau khi xử lý, mã hóa văn bản, nhóm sẽ xây dựng, kiểm thử 2 mô hình là Naive Bayes và Support Vector Machine.

4.1. Lý thuyết

4.1.1. Naive Bayes

Trong học máy, Naive Bayes là một thuật toán phân lớp dữ liệu, dựa theo định lý Bayes trong Xác suất thống kê.

$$P(H_i|A) = \frac{P(H_i).P(A|H_i)}{P(A)}$$

Hình 14. Công thức Bayes

Thuật toán sẽ lần lượt tính xác suất của một dữ liệu thuộc về một trong các lớp, rồi ra quyết định trên cơ sở xác suất của dữ liệu thuộc lớp nào là lớn nhất.

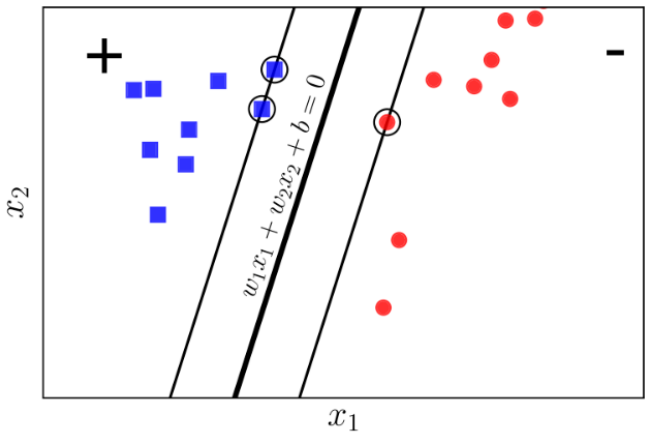
Nhờ vào tốc độ training và test rất nhanh, nên Naive Bayes phù hợp với các bài toán phân loại văn bản số lượng lớn.

Thư viện Scikit-learn cung cấp nhiều mô hình khác nhau của Naive Bayes. Nhóm sử dụng MultinomialNB, cần 2 siêu tham số là “alpha” và “fit prior”.

4.1.2. Support Vector Machine

Support Vector Machine, hay SVM, là một bộ thuật toán học máy có giám sát, được sử dụng cho các bài toán phân lớp, phân tích hồi quy hay phát hiện ngoại lệ.

Trong bài toán phân lớp, giả sử các dữ liệu là các điểm trong không gian, thì SVM sẽ đi tìm mặt phẳng tối ưu để phân chia các điểm đó thành các lớp dữ liệu; sao cho khoảng cách của điểm gần nhất trong các lớp khác nhau, là bằng nhau và lớn nhất có thể.



Hình 15. SVM trong bài toán phân lớp

Thư viện Scikit-learn cung cấp nhiều mô hình khác nhau của SVM. Nhóm sử dụng LinearSVC, cần 2 siêu tham số là "penalty" và "C".

4.2. Triển khai

Trước hết, bộ dữ liệu đã được xử lý và trích xuất đặc trưng sẽ được chia thành tập Train và tập Test với tỉ lệ 7:3.

Tiếp theo, tập Train sẽ được đem đi huấn luyện với 2 mô hình trên, sử dụng bộ siêu tham số mặc định. Các mô hình đã huấn luyện sẽ được kiểm thử trên tập Test.

Sau cùng, nhóm sẽ đi tìm siêu tham số tối ưu trên 2 mô hình, huấn luyện bằng tập Train và kiểm thử bằng tập Test.

4.3. Kết quả

4.3.1. Huấn luyện trên Small Data

1) Naive Bayes

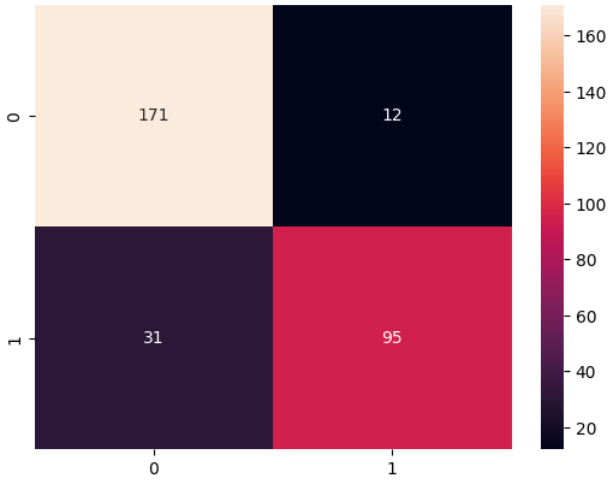
	Bag of Words	TF-IDF
Bộ tham số mặc định	88.07%	75.17%
Bộ siêu tham số	88.63% (alpha=0.5; fit_prior=False)	88.63% (alpha=0.1; fit_prior=False)

Bảng 1. Kết quả huấn luyện mô hình Naive Bayes trên Small Data

2) Naive Bayes + Bag of Words

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	85%	93%	89%	86%
Đánh giá tiêu cực	89%	75%	82%	

Bảng 2. Kết quả kiểm thử mô hình Naive Bayes + Bag of Words trên Small Data

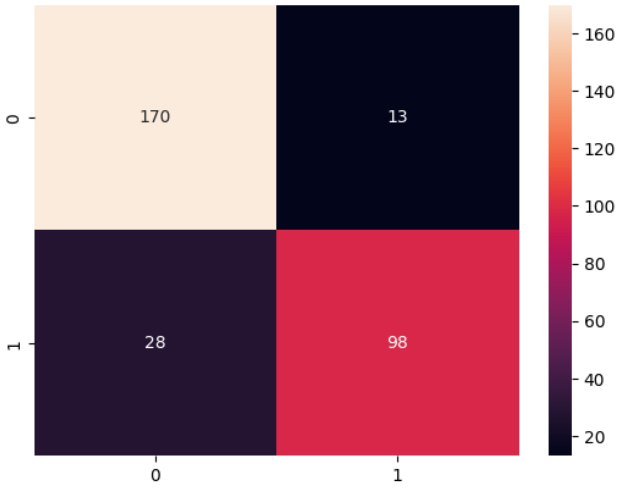


Hình 16. Ma trận nhầm lẫn của mô hình Naive Bayes + Bag of Words kiểm thử trên Small Data

3) Naive Bayes + TF-IDF

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	86%	93%	89%	87%
Đánh giá tiêu cực	88%	78%	83%	

Bảng 3. Kết quả kiểm thử mô hình Naive Bayes + TF-IDF trên Small Data



Hình 17. Ma trận nhầm lẫn của mô hình Naive Bayes + TF-IDF kiểm thử trên Small Data

4) Support Vector Machine

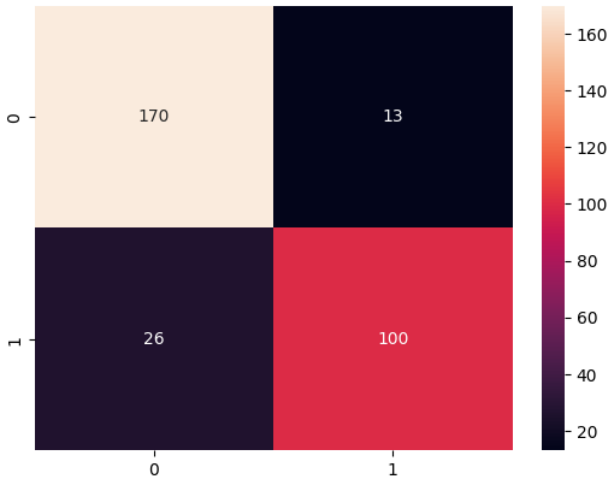
	Bag of Words	TF-IDF
Bộ tham số mặc định	81.55%	87.66%
Bộ siêu tham số	86.69% (C=0.01; penalty=l2)	88.07% (C=0.1; penalty=l2)

Bảng 4. Kết quả huấn luyện mô hình SVM trên Small Data

5) Support Vector Machine + Bag of Words

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	87%	93%	90%	87%
Đánh giá tiêu cực	88%	79%	84%	

Bảng 5. Kết quả kiểm thử mô hình SVM + Bag of Words trên Small Data

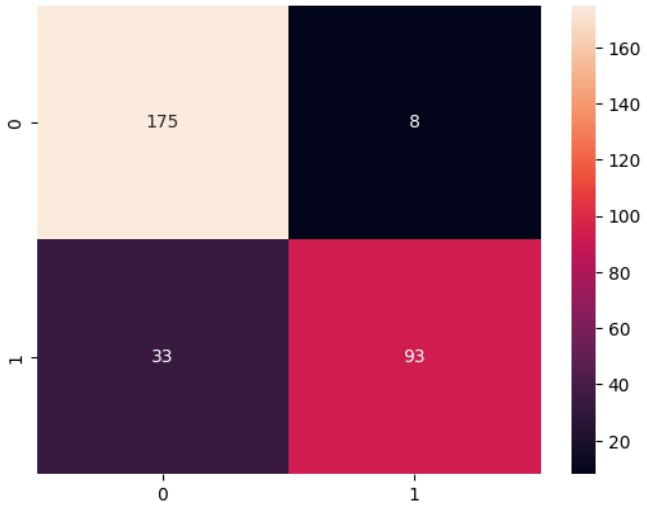


Hình 18. Ma trận nhầm lẫn của mô hình SVM + Bag of Words kiểm thử trên Small Data

6) Support Vector Machine + TF-IDF

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	84%	96%	90%	87%
Đánh giá tiêu cực	92%	74%	82%	

Bảng 6. Kết quả kiểm thử mô hình Naive Bayes + TF-IDF trên Small Data



Hình 19. Ma trận nhầm lẫn của mô hình SVM + TF-IDF kiểm thử trên Small Data

4.3.2. Huấn luyện trên Big Data

1) Naive Bayes

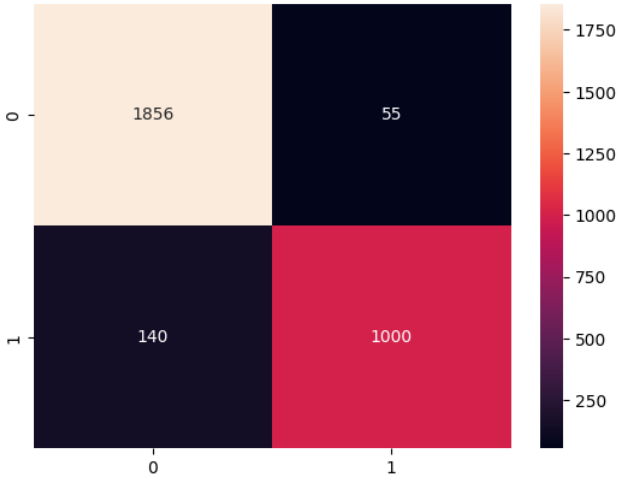
	Bag of Words	TF-IDF
Bộ tham số mặc định	91.05%	86.13%
Bộ siêu tham số	92.37% (alpha=0.1; fit_prior=False)	93.15% (alpha=0.1; fit_prior=False)

Bảng 7. Kết quả huấn luyện mô hình Naive Bayes trên Big Data

2) Naive Bayes + Bag of Words

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	93%	97%	95%	94%
Đánh giá tiêu cực	95%	88%	91%	

Bảng 8. Kết quả kiểm thử mô hình Naive Bayes + Bag of Words trên Big Data

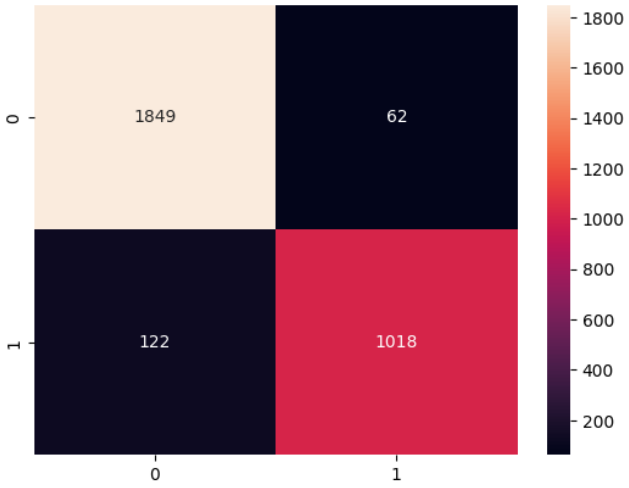


Hình 20. Ma trận nhầm lẫn của mô hình Naive Bayes + Bag of Words kiểm thử trên Big Data

3) Naive Bayes + TF-IDF

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	94%	97%	95%	94%
Đánh giá tiêu cực	94%	89%	92%	

Bảng 9. Kết quả kiểm thử mô hình Naive Bayes + TF-IDF trên Big Data



Hình 21. Ma trận nhầm lẫn của mô hình Naive Bayes + TF-IDF kiểm thử trên Big Data

4) Support Vector Machine

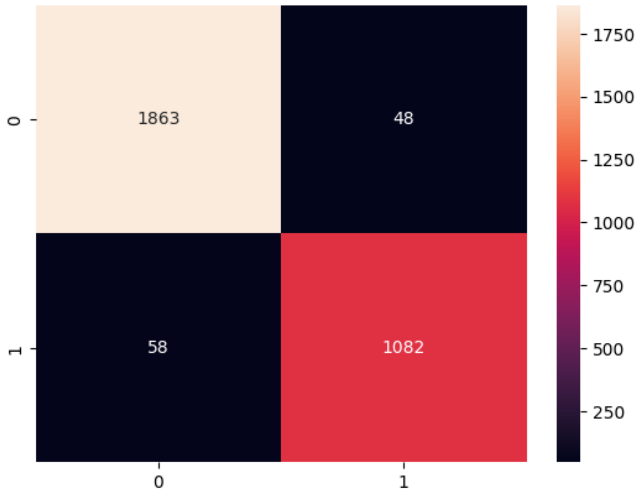
	Bag of Words	TF-IDF
Bộ tham số mặc định	95.14%	95.67%
Bộ siêu tham số	95.41% (C=0.1; penalty=l2)	95.67% (C=1.0; penalty=l2)

Bảng 10. Kết quả huấn luyện mô hình SVM trên Small Data

5) Support Vector Machine + Bag of Words

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	97%	97%	97%	97%
Đánh giá tiêu cực	96%	95%	95%	

Bảng 11. Kết quả kiểm thử mô hình SVM + Bag of Words trên Big Data

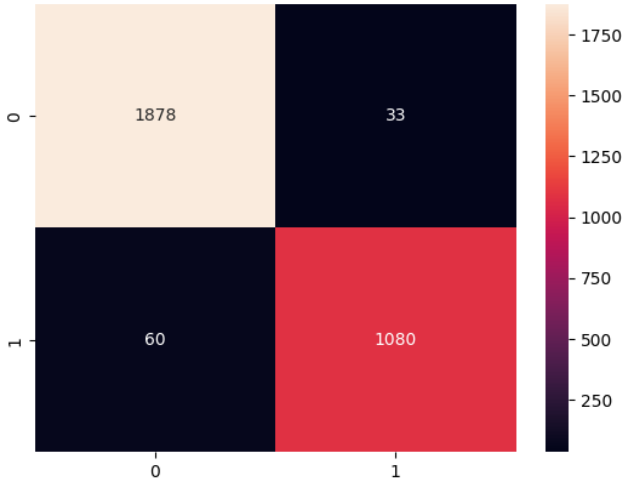


Hình 22. Ma trận nhầm lẫn của mô hình SVM + Bag of Words kiểm thử trên Big Data

6) Support Vector Machine + TF-IDF

	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	97%	98%	98%	97%
Đánh giá tiêu cực	97%	95%	96%	

Bảng 12. Kết quả kiểm thử mô hình Naive Bayes + TF-IDF trên Big Data



Hình 23. Ma trận nhầm lẫn của mô hình SVM + TF-IDF kiểm thử trên Big Data

5. Kết luận

Kết quả đạt được:

- Thu thập được nguồn dữ liệu trên nền tảng Foody.
- Áp dụng được nhiều phương pháp khác nhau để tiền xử lý ngôn ngữ tự nhiên.
- Mã hoá được văn bản thành ma trận và trực quan hoá được dữ liệu.
- Áp dụng linh hoạt các mô hình để cho kết quả có độ chính xác cao với chi phí tương đối thấp.

Trong quá trình huấn luyện mô hình:

- Việc sử dụng bộ siêu tham số không ảnh hưởng tới độ chính xác khi sử dụng CountVectorizer làm input. Tuy nhiên nó lại cải thiện đáng kể khi sử dụng TfidfVectorizer làm input. Đặc biệt với dataset nhỏ thì có thể cải thiện độ chính xác lên tới 13.46%.
- Khi huấn luyện trên dataset nhỏ thì mô hình Naive Bayes cho độ chính xác cao khi áp dụng với CountVectorizer. Trong khi đó thì mô hình Support Vector Machine đáp ứng tốt với TfidfVectorizer.
- Khi huấn luyện trên dataset lớn thì việc sử dụng bộ siêu tham số gần như không cải thiện độ chính xác của thuật toán.
- Huấn luyện trên dataset lớn luôn cho kết quả tốt hơn dataset nhỏ.

Trong quá trình kiểm thử:

- Khi kiểm thử với 2 mô hình có sử dụng bộ siêu tham số thì kết quả của tập test với 2 input khác nhau đều cho kết quả tương đương nhau. Giá trị của metric accuracy (độ chính xác) nằm trong khoảng 86-87% đối với dataset nhỏ, 94-97% đối với dataset lớn.
- Với mô hình được huấn luyện trên dataset lớn, mô hình Support Vector Machine cho kết quả nhỉn hơn (metric accuracy) so với Naive Bayes khi kiểm thử. (97% so với 94%).
- Khi kiểm thử với mô hình được huấn luyện trên dataset nhỏ, giá trị metric precision đối với đánh giá tích tiêu cực tốt hơn so với tích cực. Tuy nhiên, với metric recall và f1-score thì giá trị của đánh giá tích cực tốt hơn rất nhiều so với tiêu cực.
- Khi kiểm thử với mô hình được huấn luyện trên dataset lớn thì tất cả các metric có giá trị tương đương nhau.

Hướng phát triển:

- Thu thập thêm nhiều nguồn dữ liệu để tăng độ chính xác.
- Áp dụng thêm các phương pháp tiền xử lý dữ liệu khác, đặc biệt là đối với một ngôn ngữ khó như tiếng việc, giúp giảm chi phí trong việc huấn luyện mô hình và nâng cao độ chính xác.
- Thử nghiệm thêm nhiều loại mô hình học máy phức tạp khác để cải thiện độ chính xác với một mức chi phí có thể chấp nhận được.

6. Tài liệu tham khảo

- Công cụ thu thập dữ liệu: Selenium with Python — Selenium Python Bindings 2 documentation (selenium-python.readthedocs.io)
- Các kĩ thuật vector hóa văn bản: Vectorization Techniques in NLP [Guide] (neptune.ai)
- Naive Bayes: Machine Learning cơ bản (machinelearningcoban.com)
- SVM: Machine Learning cơ bản (machinelearningcoban.com)