

PHÂN LOAI REVIEW BÌNH LUÂN TÍCH CỰC VÀ TIÊU CỰC TRÊN FOODY



Các thành viên và nhiệm vụ



Dương Mạnh Quân

- Viết module thu thập dữ liệu
 - Thu thập dữ liệu
 - Làm slide báo cáo



Nguyễn Ngọc Hải

- Tiền xử lý dữ liệu
- Trực quan hóa dữ liệu
- Huấn luyện mô hình
Naive Bayes

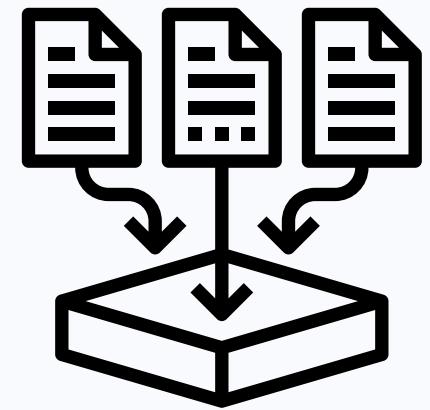


Trương Hoàng Nhật Huy

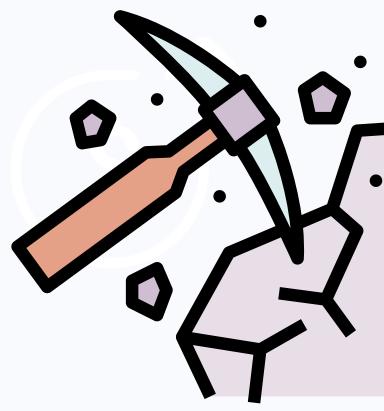
- Huấn luyện mô hình SVM
- Viết quyển báo cáo



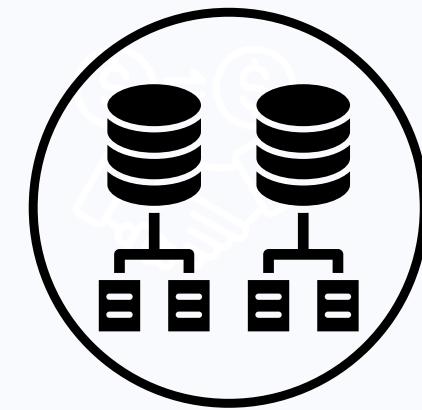
Tổng quan tiểu luận



Thu thập và mô
tả dữ liệu



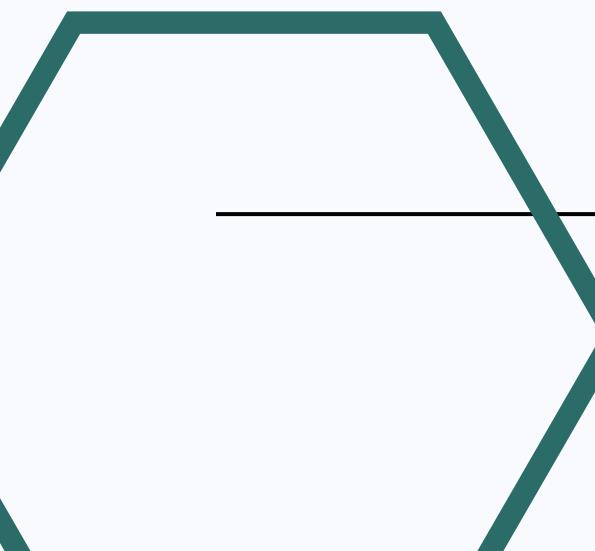
Trích xuất đặc trưng



Mô hình hóa dữ
liệu



Kết luận





1.Giới thiệu

1.1 Mục tiêu

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.

1.2 Giải quyết

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 . Từ đó giúp cho nắm rõ các đặc điểm, nhu cầu của khách hàng



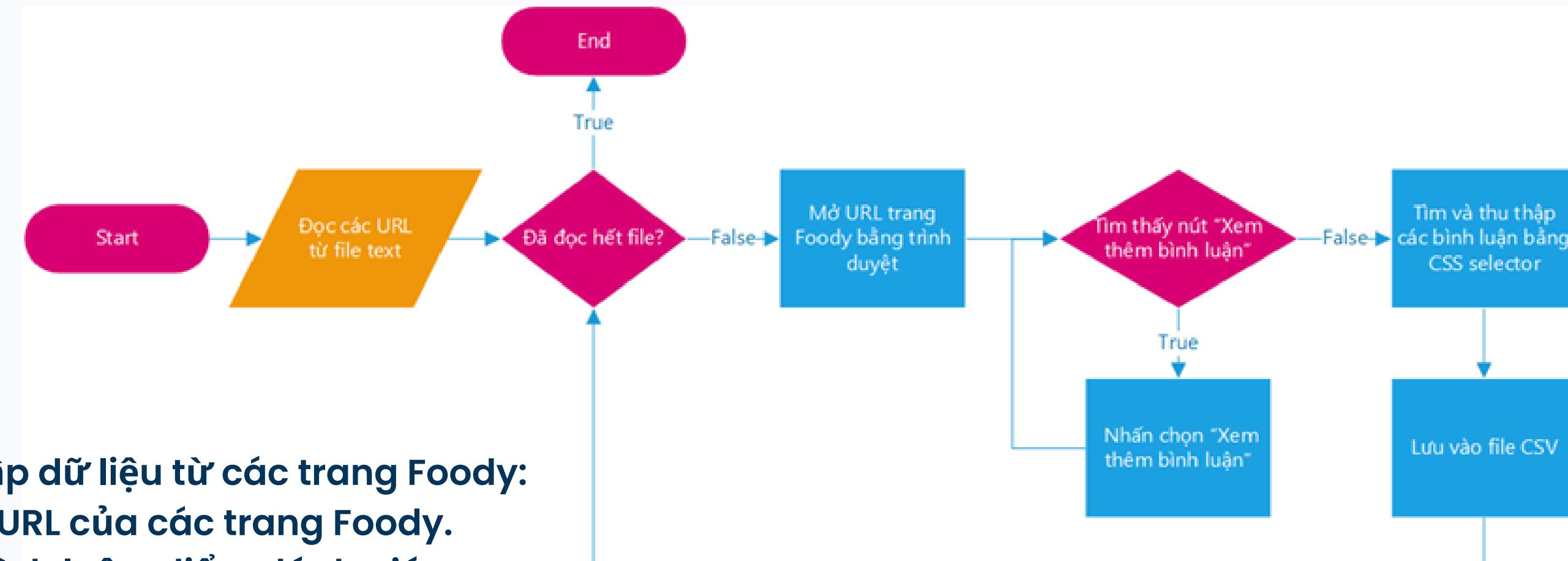


2.Thu thập dữ liệu

2.1 Thu thập dữ liệu

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

2.2 Script thu thập dữ liệu



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.



2.Thu thập dữ liệu



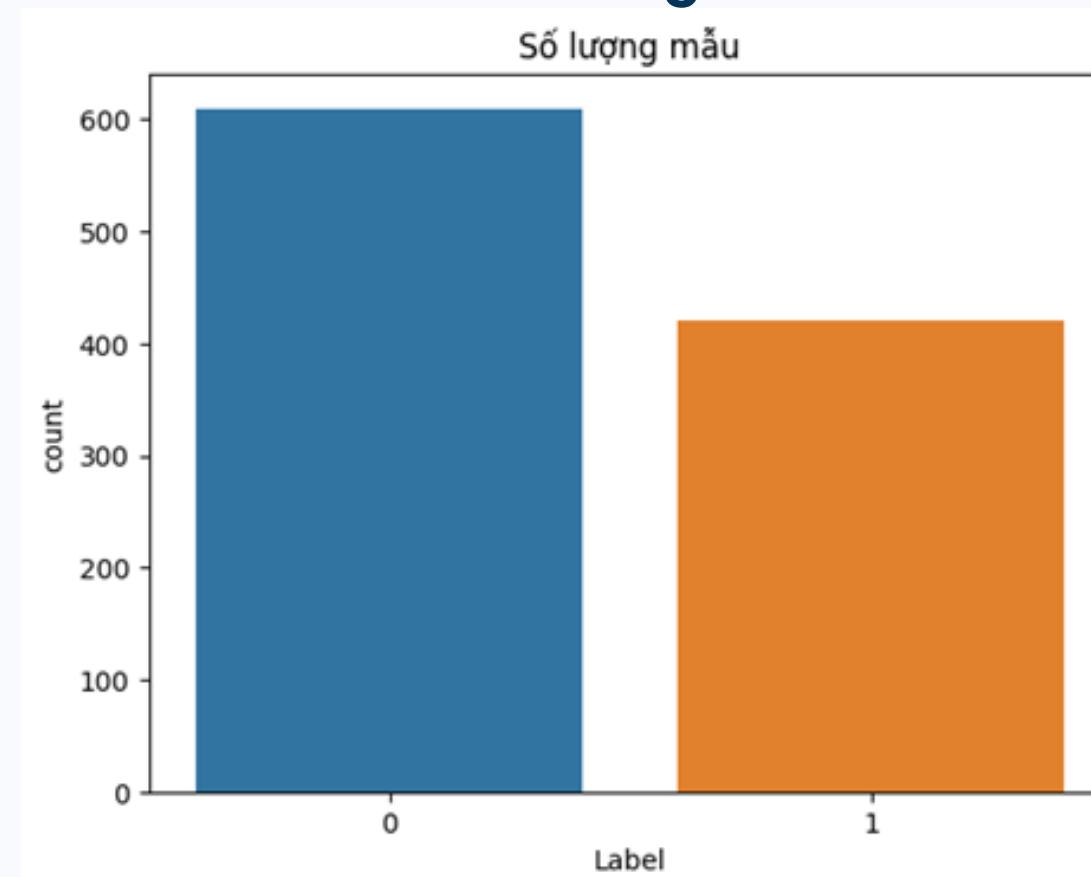
2.3 Mô tả dữ liệu

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:

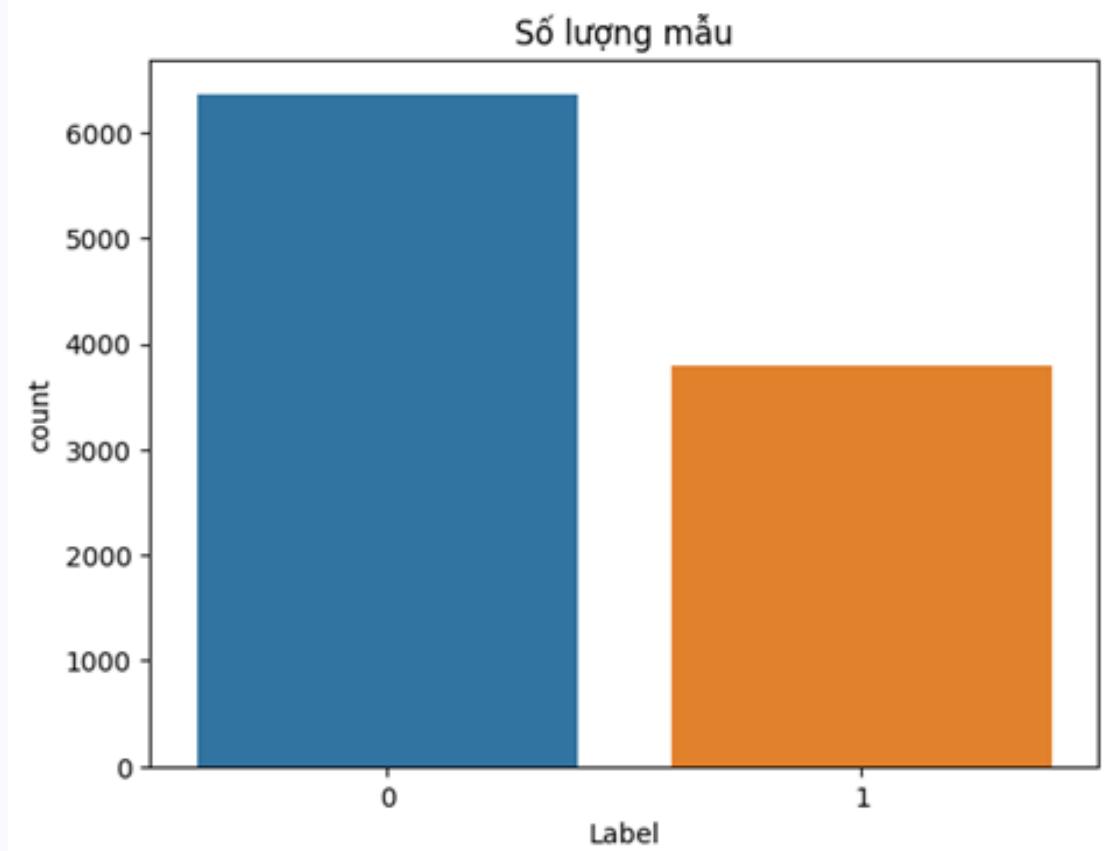
- Bình luận đánh giá là chuỗi kí tự tiếng Việt.
- Điểm đánh giá là số thực trong khoảng [0;10]

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



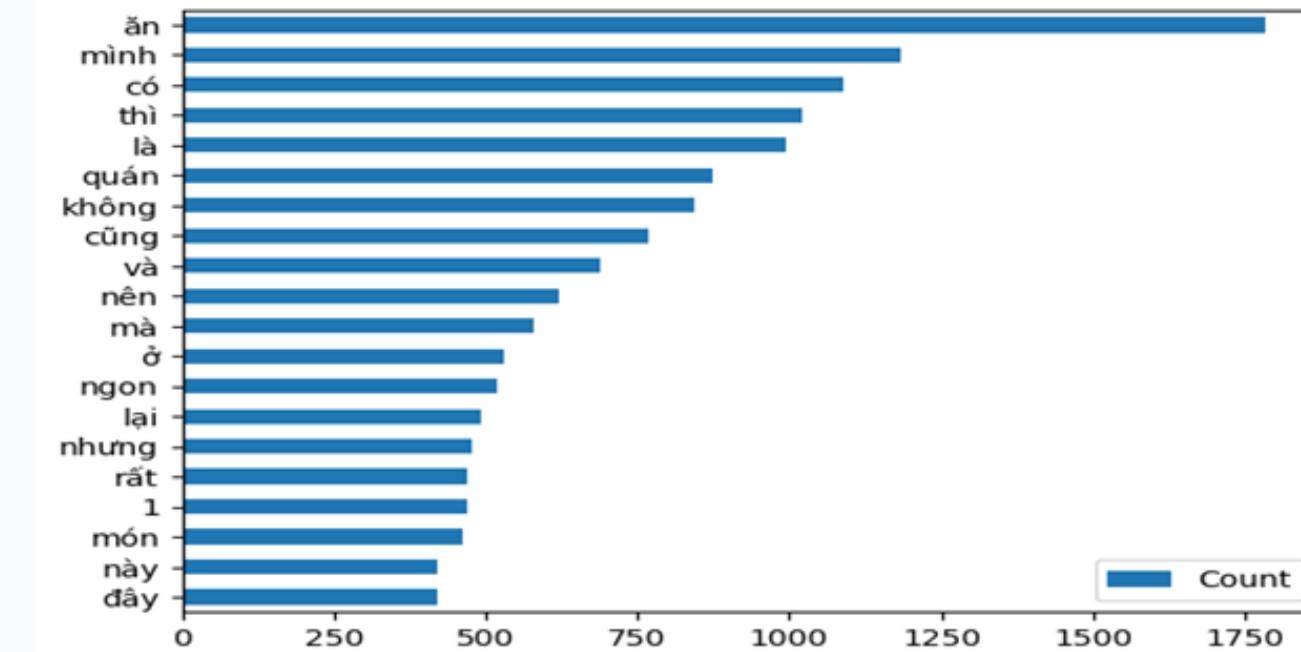
Trực quan hóa dữ liệu



Trước khi tiền xử lý



Word cloud của Small Data
trước khi tiền xử lý



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



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



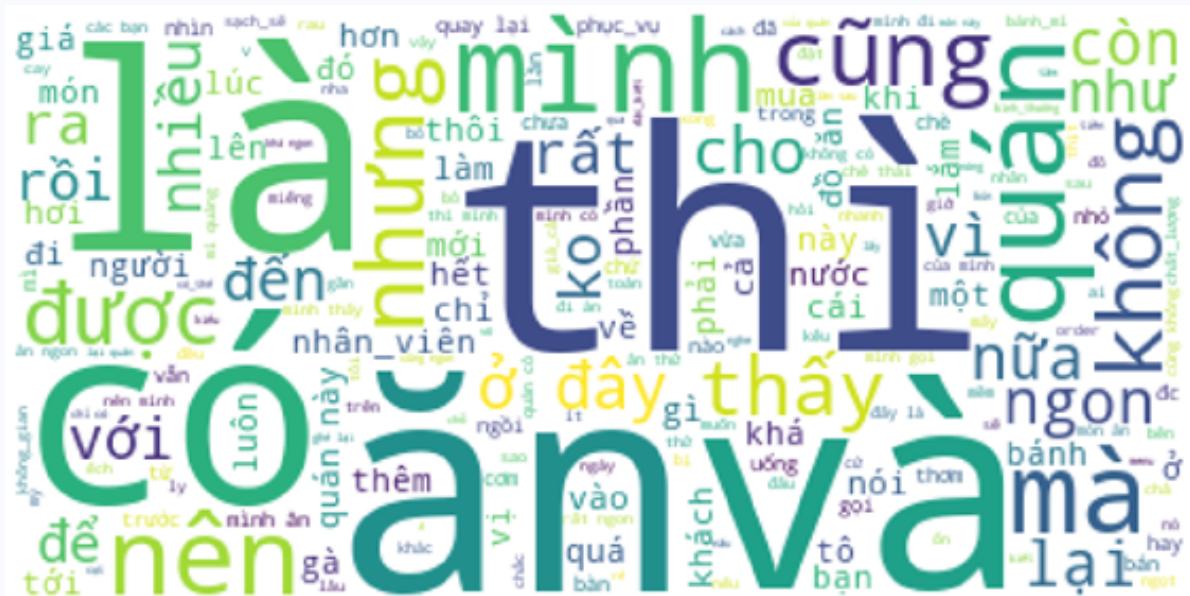
Word cloud của các bình luận tích cực
trong Small Data trước khi tiền xử lý



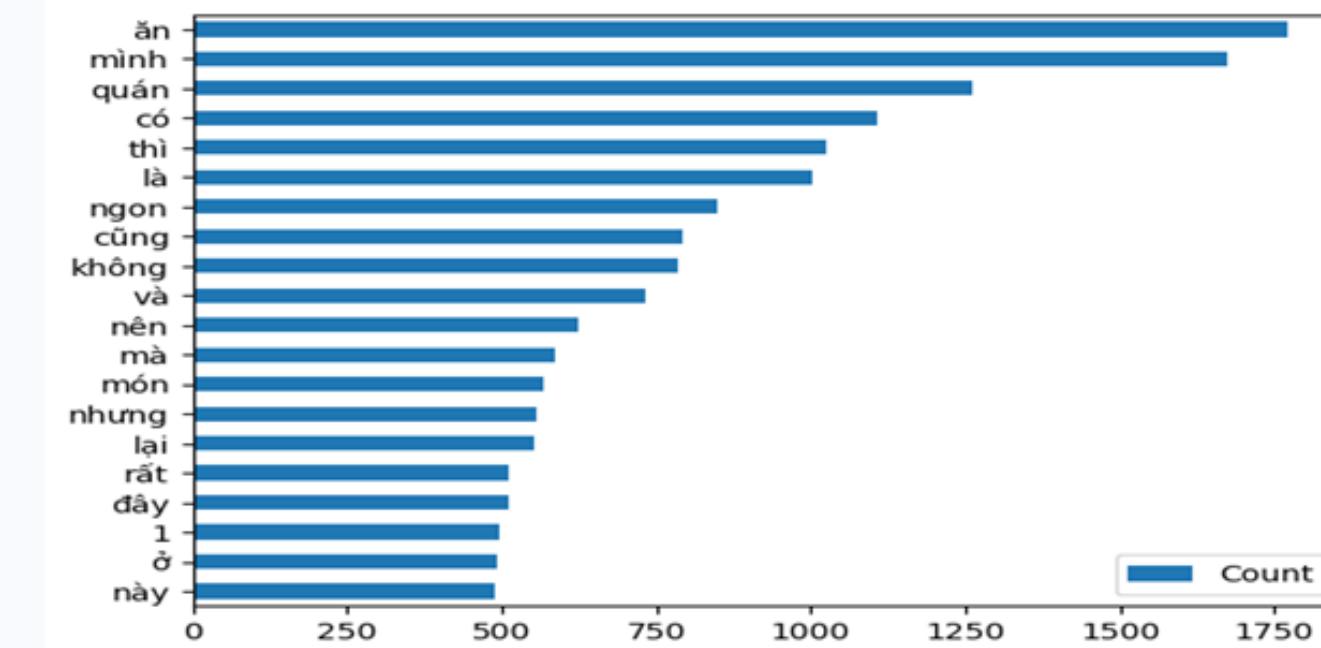
Trực quan hóa dữ liệu



Sau khi tiền xử lý



Word cloud của Small Data sau khi tiền xử lý



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



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



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

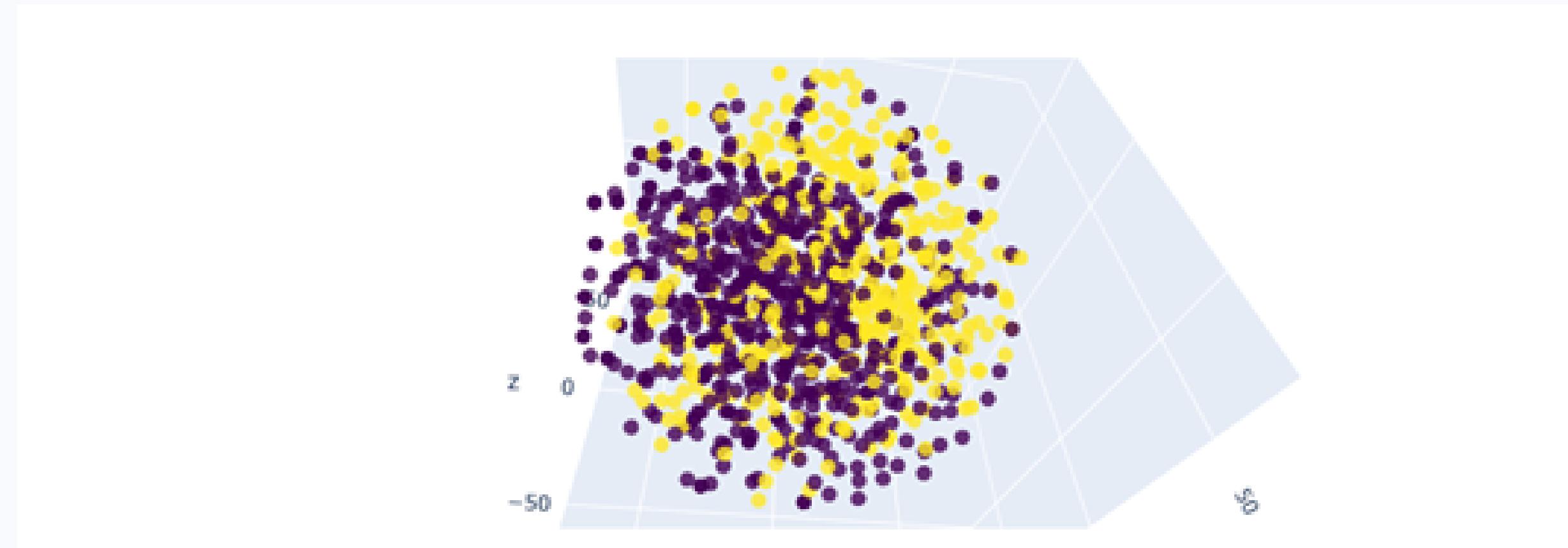


3.Trích xuất đặc trưng



3.1. Kĩ thuật 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



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

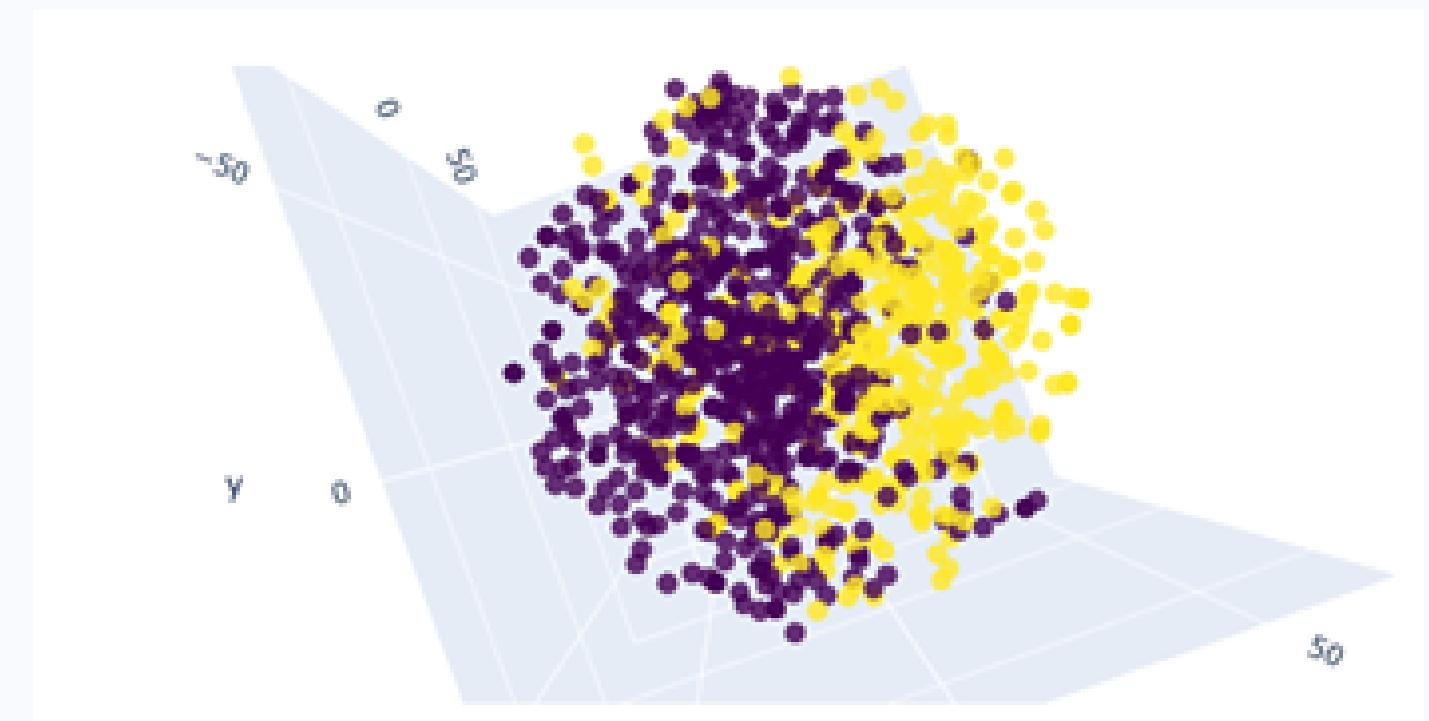
3.Trích xuất đặc trưng

3.2. Kĩ thuật 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:

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

- 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.



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



4. Mô hình hóa dữ liệu



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.

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

(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. Mô hình hóa dữ liệu

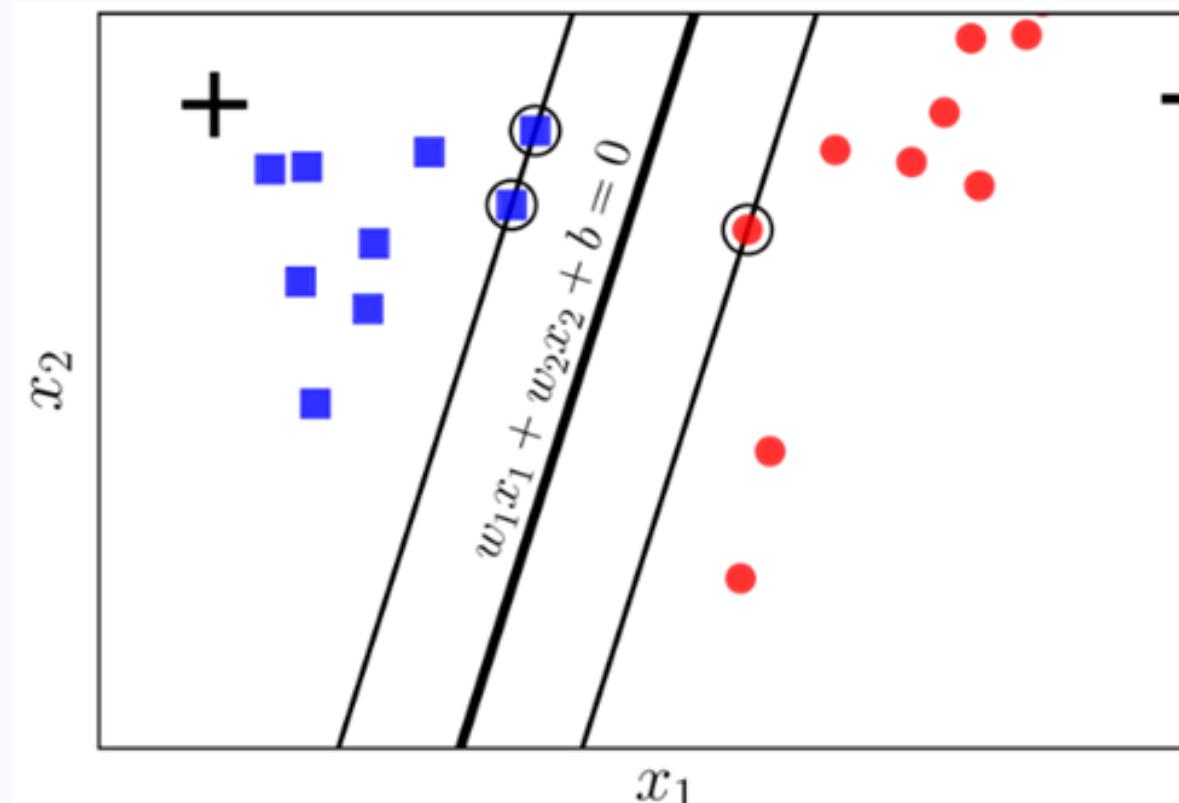


4.1 Lý thuyết

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



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”.

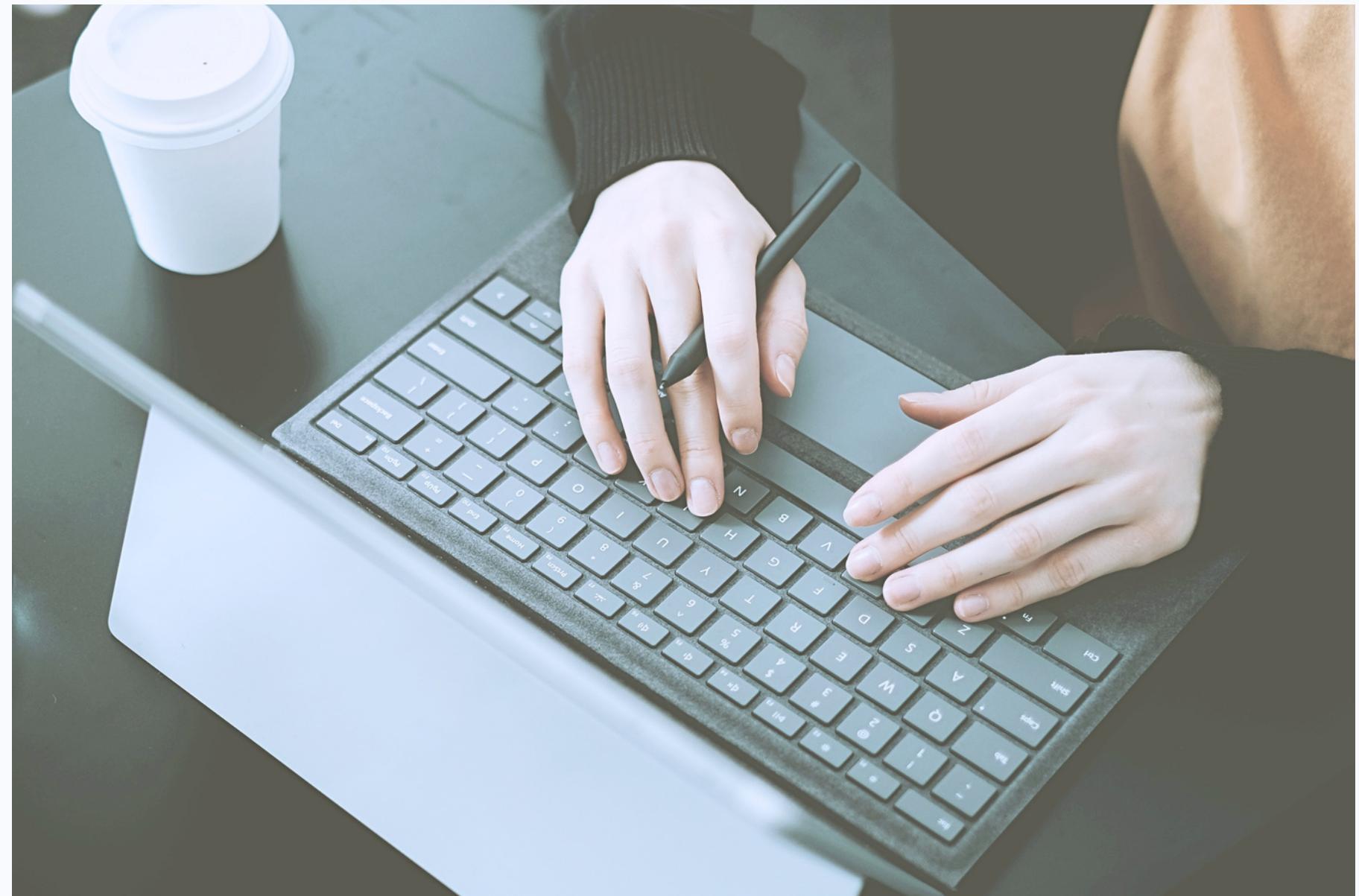


4. Mô hình hóa dữ liệu



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. Mô hình hóa dữ liệu

4.3 Kết quả:

4.3.1 Huấn luyện trên Small Data :

Naive Bayes

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

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

Naive Bayes + Bag of Words

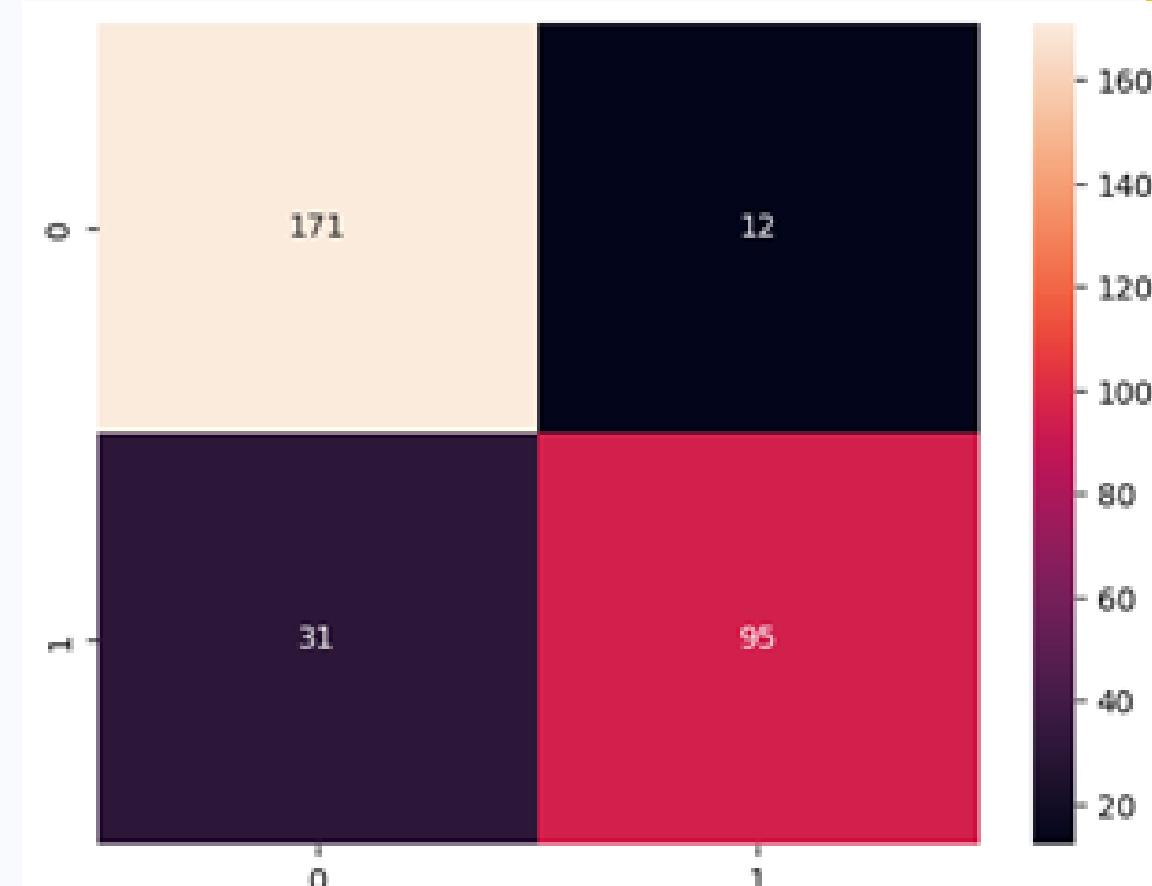
	Metric Precision	Metric Recall	Metric f1-score	Accuracy
Đánh giá tích cực	85%	93%	89%	86%
Đánh giá tiêu cực	89%	75%	82%	

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

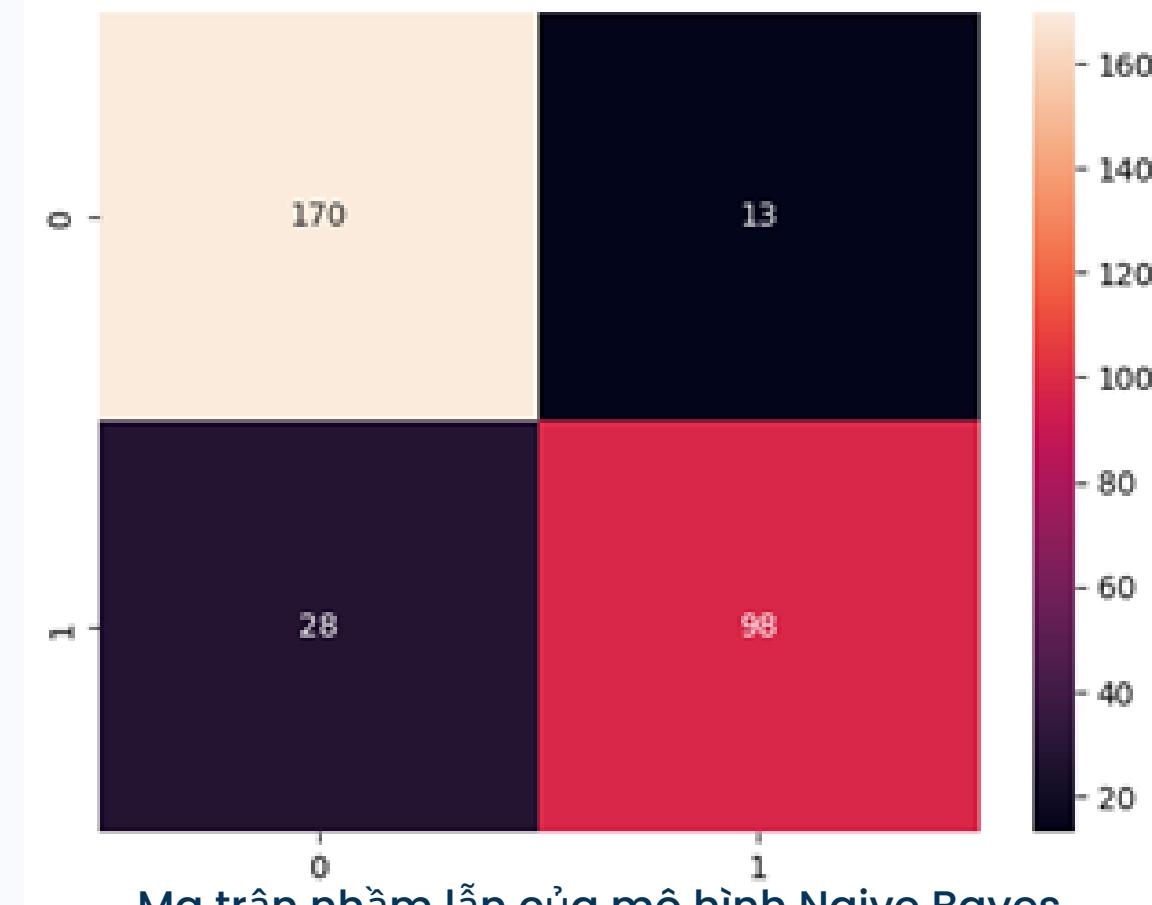
Naive Bayes + TF-IDF

	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



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



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



4. Mô hình hóa dữ liệu

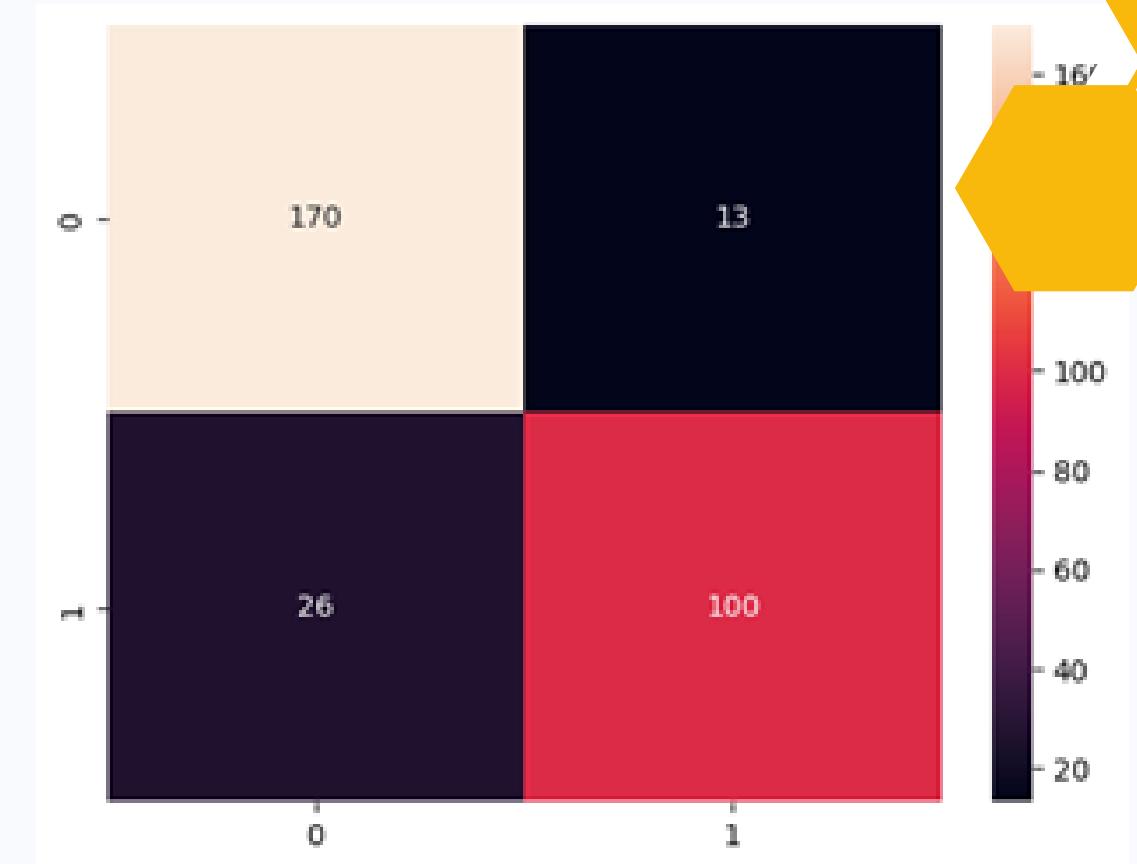
4.3 Kết quả:

4.3.1 Huấn luyện trên Small Data :

Support Vector Machine

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

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

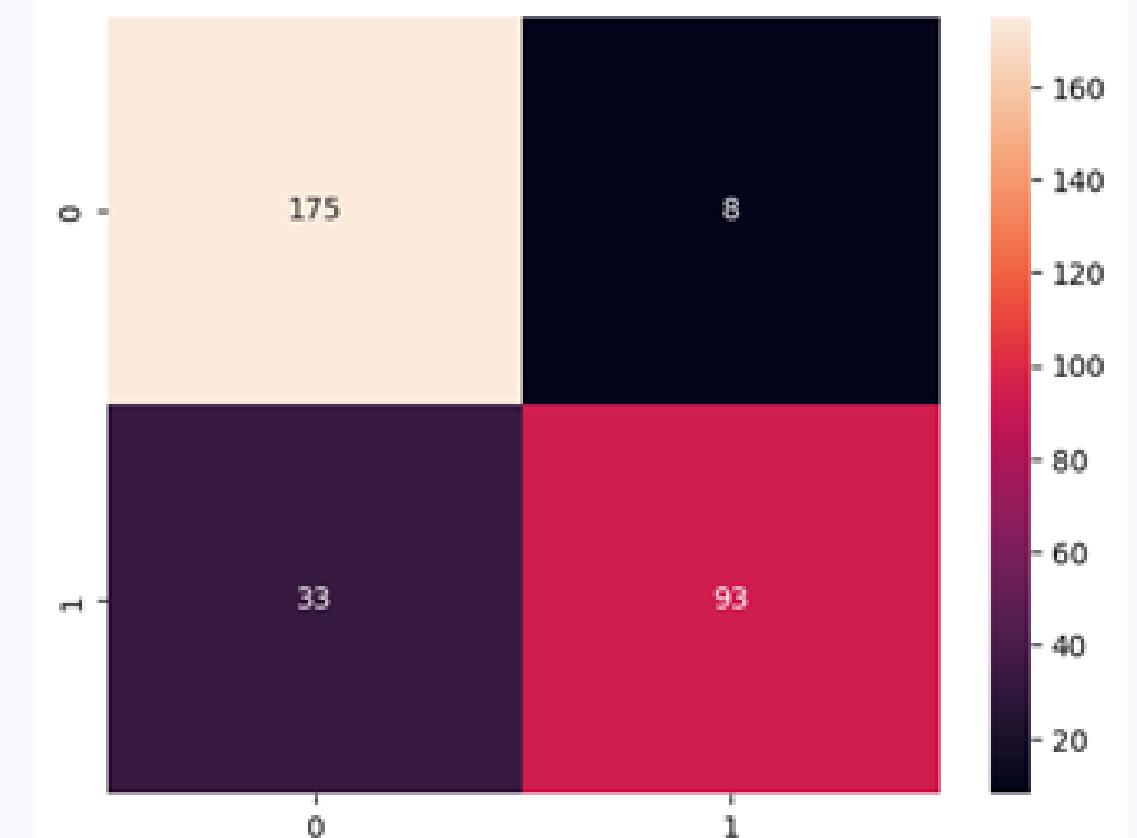


Ma trận nhầm lẫn của mô hình SVM + Bag of Words kiểm thử trên Small Data

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



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

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



4. Mô hình hóa dữ liệu

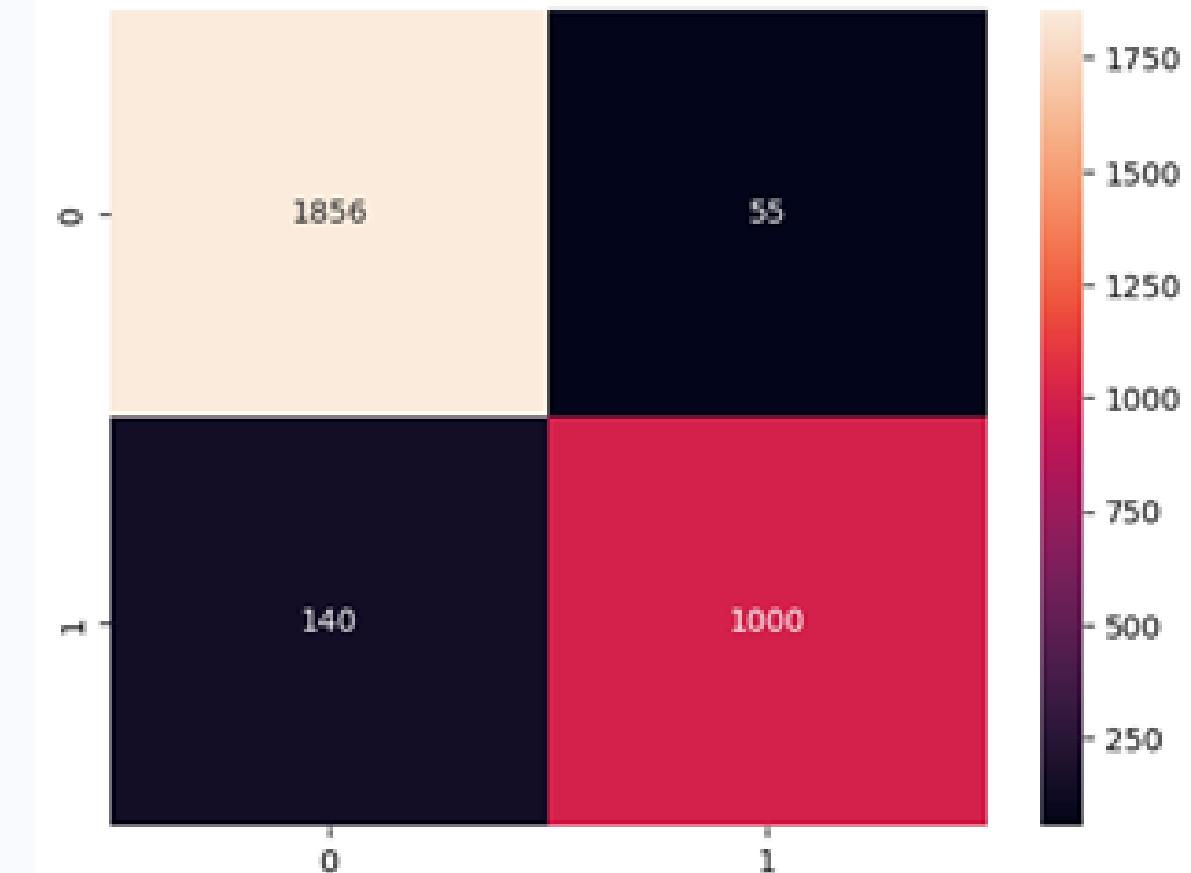
4.3 Kết quả :

4.3.2 Huấn luyện trên Big Data :

Naive Bayes

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

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

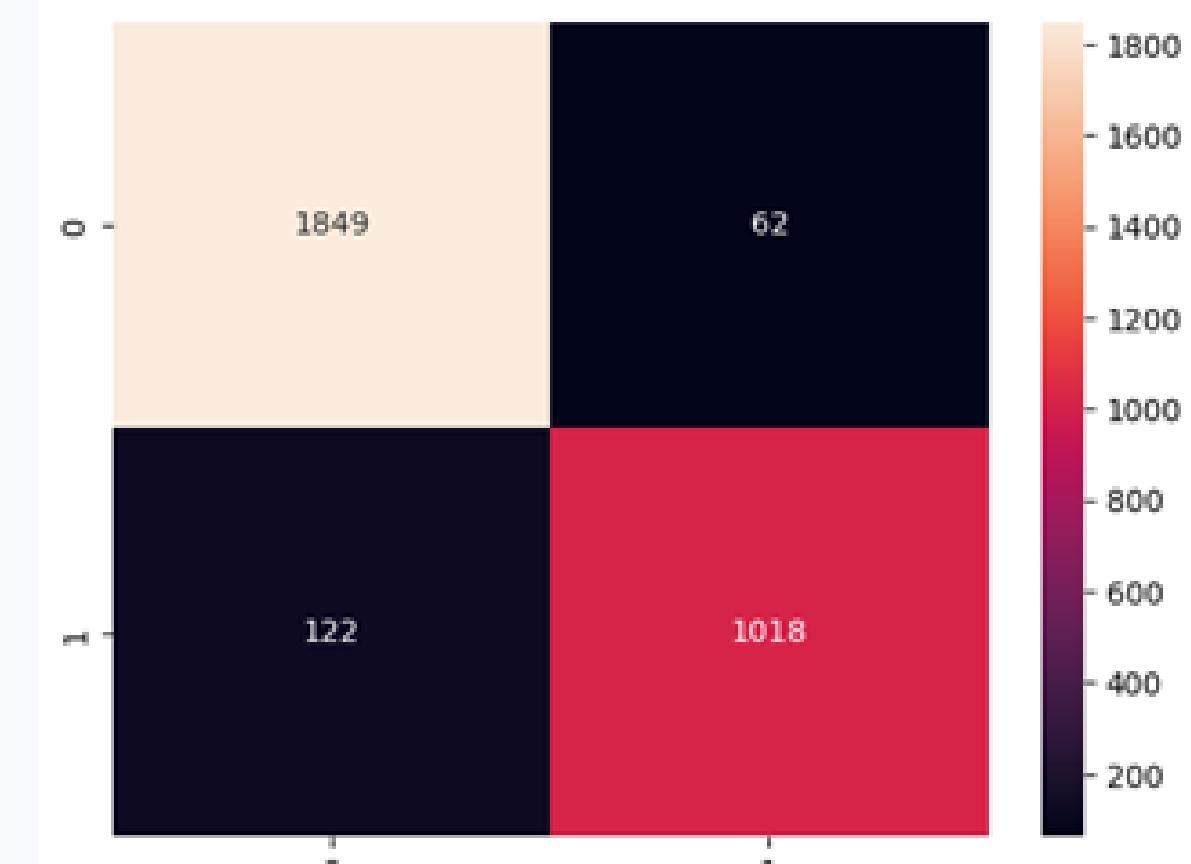


Ma trận nhầm lẫn của mô hình Naive Bayes
+ Bag of Words kiểm thử trên Big Data

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



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

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



4. Mô hình hóa dữ liệu

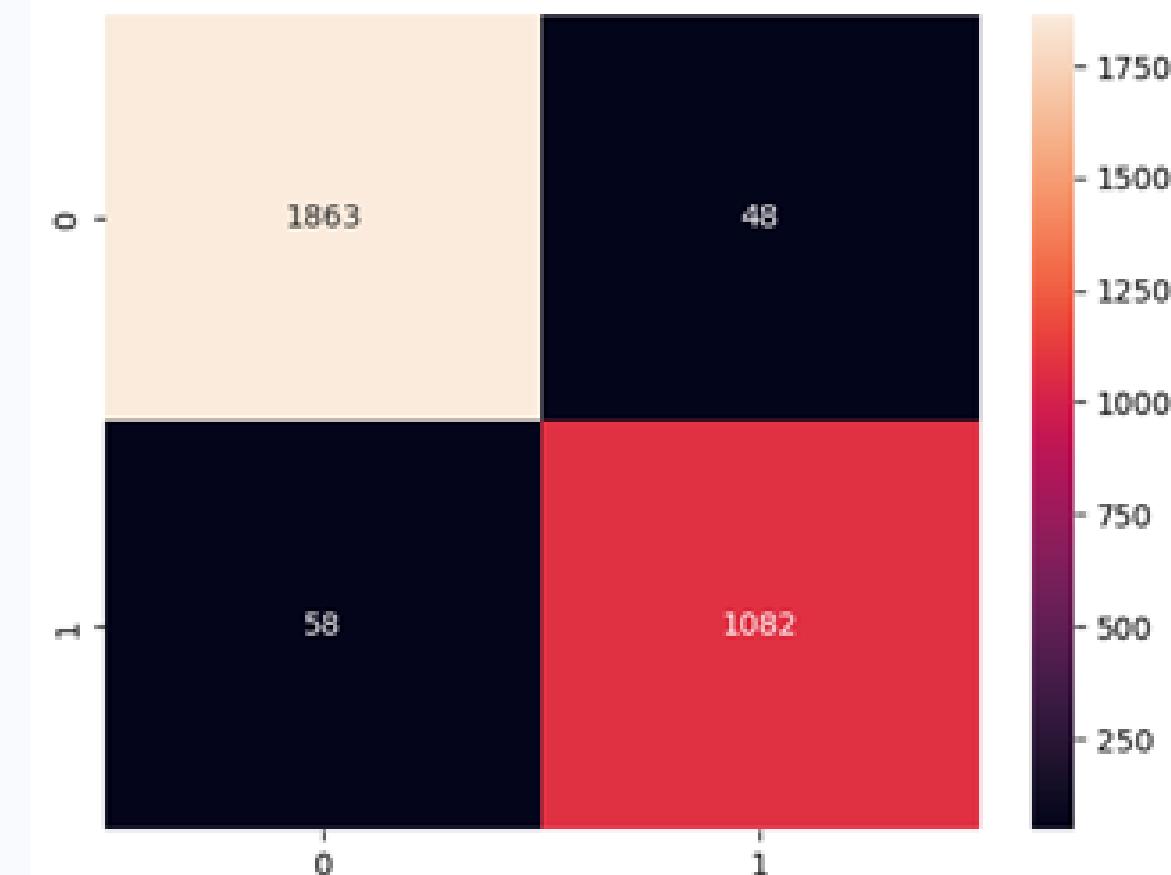
4.3 Kết quả:

4.3.2 Huấn luyện trên Big Data :

Support Vector Machine

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

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

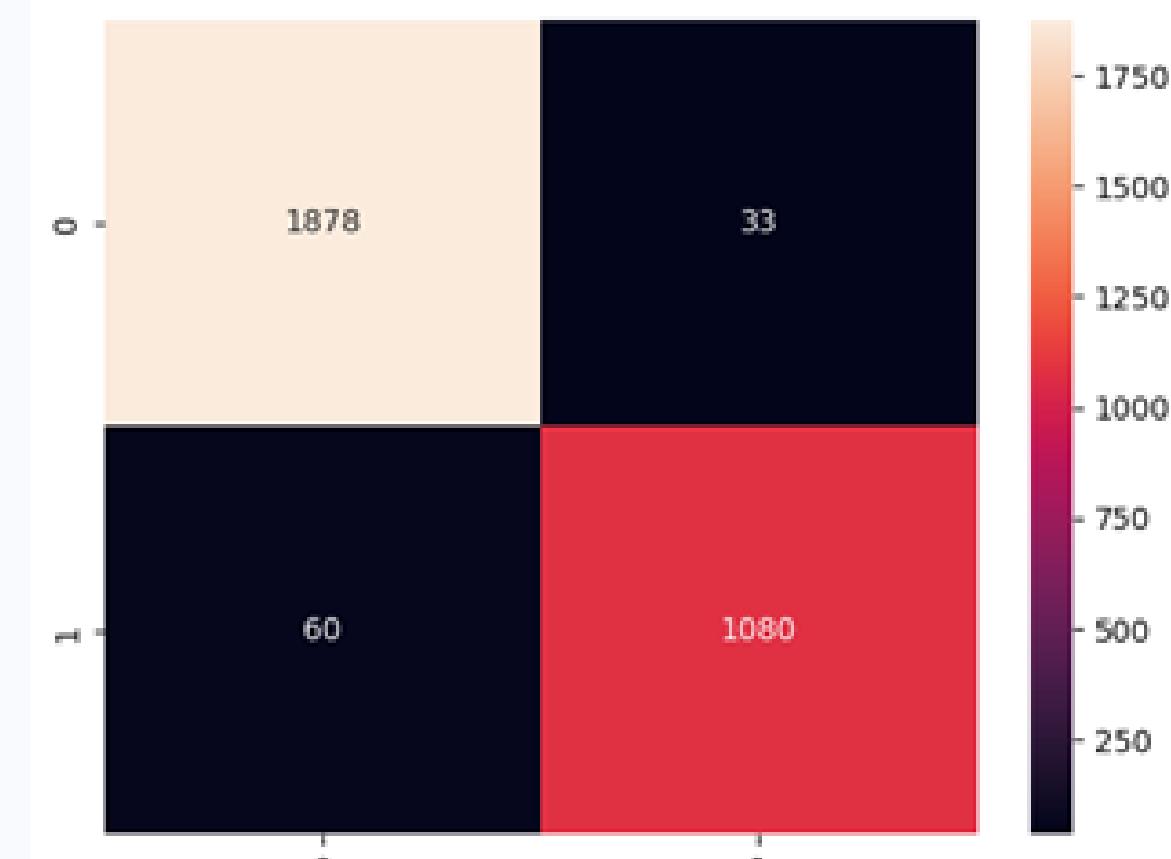


Ma trận nhầm lẫn của mô hình SVM + Bag of Words kiểm thử trên Big Data

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

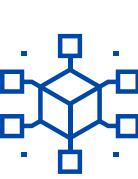


Ma trận nhầm lẫn của mô hình SVM + TF-IDF kiểm thử trên Big Data

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



5.Kết luận

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 13.03%
- Khi huấn luyện trên dataset nhỏ thì mô hình **Naive Bayes** cho độ chính xác cao hơn 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 thì **SupportVector Machine** cho kết quả tốt hơn (metric accuracy) so với **Naive Bayes** khi kiểm thử
- 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



THANK FOR
WATCHING