**NHẬN DIỆN BÌNH LUẬN SẢN PHẨM SPAM TRÊN TIẾNG VIỆT**

Đỗ Phạm Phúc Tính  
*Trường Đại học Công nghệ Thông tin*

ĐHQG Thành phố Hồ Chí Minh, Việt Nam  
[20522020@gm.uit.edu.vn](mailto:20522020@gm.uit.edu.vn)

Cao Đình Duy Ngọc  
*Trường Đại học Công nghệ Thông tin*  
ĐHQG Thành phố Hồ Chí Minh, Việt Nam  
[20521661@gm.uit.edu.vn](mailto:20521661@gm.uit.edu.vn)

Nguyễn Trần Gia Thế  
*Trường Đại học Công nghệ Thông tin*  
ĐHQG Thành phố Hồ Chí Minh, Việt Nam  
[2051940@gm.uit.edu.vn](mailto:2051940@gm.uit.edu.vn)

Ngô Huỳnh Trưởng  
*Trường Đại học Công nghệ Thông tin*  
ĐHQG Thành phố Hồ Chí Minh, Việt Nam  
20522085@gm.uit.edu.vn

***Abstract*.** Với sự phát triển nhanh chóng của công nghệ 4.0 dẫn đến hình thức kinh doanh trực tuyến cũng rất phát triển, bình luận đánh giá sản phẩm của khách hàng ngày càng nhiều với nội dung và mục đích khác nhau. Bài toán nhận diện bình luận spam đang được nhiều nhà nghiên cứu quan tâm và xây dựng các mô hình học máy các phương pháp tiền xử lý nhằm nâng cao hiệu quả để giúp cho khách hàng mua được sản phẩm chất lượng, trải nghiệm mua sắm tốt hơn. Ở bài toán này, chúng tôi sử dụng dữ liệu được kế thừa từ bộ ViSpam Dectection [1] và thực hiện trên các mô hình học máy Logistic [2], SVM [3], KNN [4], Naive Bayes [5] nhằm đưa ra mô hình và tham số có kết quả tốt nhất để ứng dụng vào thực tế. Kết quả thí nghiệm cho thấy, mô hình hồi quy Logistic [2] cho kết quả cao nhất, đạt 83% đối với độ đo accuracy và 77% đối với độ đo F1-score.

Keywords. ViSpam Dectection đánh giá spam, GridSeachCV, mô hình máy học, Logistic Regression [2].

# **i. GIỚI THIỆU ĐỀ TÀI**

**1. Tổng quan đề tài**

Khi khách hàng muốn mua một sản phẩm thường sẽ đọc những phản hồi mà những khách hàng trước để biết sản phẩm đó có chất lượng tốt hay không. Nhưng với sự phát triển nhanh chóng của công nghệ 4.0 dẫn đến hình thức kinh doanh

trực tuyến cũng rất phát triển, bình luận đánh giá sản phẩm của khách hàng ngày càng nhiều với nội

dung và mục đích khác nhau. Vấn đề đặt ra là làm sao khách hàng biết được đó có phải những đánh

giá thật hay chỉ là những đánh giá mang nội dung, ý nghĩa khác. Doanh nghiệp, cửa hàng cũng không thể biết được đó là những đánh giá thật sự của khách hàng hay là những đánh giá spam của đối thủ cạnh tranh với mục đích hạ thấp danh tiếng của sản phẩm. Để giải quyết bài toán này, chúng tôi quyết định thực hiện bài toán nhận diện đánh giá spam trên sàn thương mai điện tử bằng tiếng Việt và áp dụng các phương pháp học máy để phân loại dữ liệu, xem bình luận nào có hành vi spam hoặc no spam và dùng các phương pháp phân tích và dự đoán. Mục đích của bài toán nhằm giúp doanh nghiệp nắm bắt thông tin một các dễ dàng và nhanh chóng, từ đó việc phát triển kinh doanh được cải thiện và nâng cao, cũng như khách hàng có thể đưa ra sự lựa chọn sản phẩm hợp lý.

Bài đánh giá spam là bài đánh giá hoàn toàn hoặc một phần sai sự thật về sản phẩm được bán online trên các trang web. Bài đánh giá spam là những bài đánh giá nhằm mục đích muốn việc kinh doanh trở nên tốt hơn hoặc làm ảnh hưởng xấu đến việc kinh doanh của người khác. Sau đây là các trường hợp được xem là spam:

* Những bài đánh giá cố tình đánh lừa người đọc bằng cách đưa ra những ý kiến tích cực để quảng bá một số sản phẩm hoặc đưa ra những ý kiến tiêu cực không đúng về một số sản phẩm nhằm tổn hại danh tiếng của họ.
* Những bài đánh giá mang tính chất văn mẫu do sàn thương mại gợi ý.
* Các bài đánh giá này tuy xuất hiện dưới dạng đánh giá nhưng không phải đánh giá về sản phẩm mà chỉ dùng để quảng cáo hoặc nhằm mục đích khác.
* Các bình luận với nội chưa dùng sản phẩm hoặc mua giúp người khác, hoặc các đánh giá hoặc các bài đánh giá bằng tiếng Anh.

**2. Định nghĩa bài toán**

Đầu vào: một bình luận đánh giá sản phẩm bằng tiếng Việt.

Đầu ra: 1 trong 2 nhãn:

- spam

- no spam

Ví dụ:

Đầu vào:

sản phẩm siu mềm và đẹp, canh sale nên mua với giá siu hời.. Váy mặc không bị lộ bên trong đâu nha nên m.n cứ yên tâm, mặc đep lắm ạ... Tuyệt cú mèo.... Chúc shop mua may bán đắt

Đầu ra: no spam

# **II. GIỚI THIỆU BỘ DỮ LIỆU**

Chúng tôi kế thừa dữ liệu từ bộ ViSpam Dectection [1], bộ này có 19870 dòng và 3 cột: comment, label và spam label. Tuy nhiên ở bài toán của chúng tôi, chúng tôi tiến hành loại bỏ cột spam label để phù hợp hơn với bài toán.

Trong đó thuộc tính label có 2 nhãn:

**0** tương ứng với đầu ra là **no spam**

**1** tương ứng với đầu ra là **spam**

Sau đây là một vài thống kê sơ bộ về bộ dữ liệu của chúng tôi sau khi đã loại bỏ cột spam label từ bộ dữ liệu gốc:

Bảng 1. Số lượng từng nhãn trong bộ dữ liệu của chúng tôi

|  |  |
| --- | --- |
| Nhãn | Số lượng |
| 0 | 14609 |
| 1 | 5261 |

Bảng 2. Số lượng từ trong 1 bình luận

|  |  |
| --- | --- |
| Khía cạnh | Số lượng |
| Số từ nhiều nhất trong 1 bình luận | 185 |
| Số từ ít nhất trong 1 bình luận | 0\* |
| Số từ trung bình của bình luận | 26 |

\*: Vì có những bình luận chỉ có những ký tự

Chúng tôi tiến hành chia tập dữ liệu sẵn sàng thành 3 **tập huấn luyện** (train set), phát triển (dev set) và **kiểm thử** (test set) để thí nghiệm các mô hình. 3 tập dữ liệu được chia ngẫu nhiên theo tỉ lệ train : dev : test là 7: 1.5 : 1.5. Cụ thể số lượng bình luận mỗi tập được thể hiện ở bảng sau:

Bảng 3. Số lượng dòng dữ liệu ứng với mỗi tập dữ liệu được chia.

|  |  |
| --- | --- |
| Tập dữ liệu | Số lượng |
| train | 13909 |
| dev | 2980 |
| test | 2981 |

Có thể thấy số lượng tập dev và tập test bằng nhau bởi vì chúng tôi muốn có sự khách quan, kết quả gần với thực tế nhất.

# **III. PHƯƠNG PHÁP THỰC HIỆN**

**1. Tổng quan phương pháp**

Trong học máy, phân lớp đề cập đến một vấn đề mô hình dự đoán trong đó nhãn lớp được dự đoán cho một ví dụ nhất định về dữ liệu đầu vào.

Trong phạm vi của báo cáo này, máy tính được cho trước một đoạn văn bản là bình luận của người dùng trên các trang bán hàng trực tuyến và được yêu cầu xác định đoạn văn bản đó có spam hay không.

**2. Các mô hình phân lớp**

**2.1 Mô hình Logistic Regression** [2]

Hồi quy logistic [2] là một thuật toán machine learning có giám sát mạnh mẽ được sử dụng cho các bài toán phân loại nhị phân (khi mục tiêu là phân loại). Sự khác biệt cơ bản giữa hồi quy tuyến tính và hồi quy logistic [2] là phạm vi của hồi quy logistic bị giới hạn từ 0 đến 1. Logistic sử dụng hàm Sigmoid:

**2.2 Mô hình Support Vector Machine** [3]

Support Vector Machines (SVM) [3] là một kỹ thuật phân lớp dữ liệu, là phương pháp học sử dụng không gian giả thuyết các hàm tuyến tính trên không gian đặc trưng nhiều chiều, dựa trên lý thuyết tối ưu và lý thuyết thống kê.

Trong kỹ thuật SVM [3] không gian dữ liệu nhập ban đầu sẽ được ánh xạ vào không gian đặc trưng và trong không gian đặc trưng này mặt siêu phẳng phân chia tối ưu sẽ được xác định. Margin là khoảng cách giữa ѕiêu phẳng (trong trường hợp không gian 2 chiều là đường thẳng) đến 2 điểm dữ liệu gần nhất tương ứng ᴠới 2 phân lớp. SVM [3] cố gắng tối ưu thuật toán bằng các tìm maхimiᴢe giá trị margin nàу, từ đó tìm ra ѕiêu phẳng đẹp nhất để phân 2 lớp dữ liệu.

**2.3 Mô hình Naive Bayes** [5]

Naive Bayes Classification là một thuật toán phân loại dựa trên tính toán xác suất áp dụng định lý Bayes. Thuật toán này thuộc nhóm Supervised Learning (Học có giám sát).

Theo định lý Bayes, ta có công thức tính xác suất ngẫu nhiên của sự kiện y khi biết x như sau:

Giả sử ta phân chia 1 sự kiện x thành n thành phần khác nhau x1, x2,…, xn.



Naïve Bayes theo đúng như tên gọi dựa vào một giả thuyết ngây thơ rằng x1, x2,…, xn là các thành phần độc lập với nhau. Từ đó có thể tính được:

Do đó ta có:

Trên thực tế thì ít khi tìm được dữ liệu mà các thành phần là hoàn toàn độc lập với nhau. Tuy nhiên giả thiết này giúp cách tính toán trở nên đơn giản, training data nhanh, đem lại hiệu quả bất ngờ với các lớp bài toán nhất định.

**2.4 Mô hình K-Nearest Neighbor** [4]

K-Nearest Neighbor (KNN) [4] là một trong những phương pháp phân loại cơ bản và đơn giản nhất, là một trong những lựa chọn đầu tiên cho nghiên cứu phân loại khi có rất ít hoặc không có kiến thức trước về sự phân bố của dữ liệu.

Thuật toán KNN [4] cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm cũng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp.

**3. Độ đo đánh giá mô hình**

Theo như bài toán phân với , chúng tôi sử dụng 2 độ đo là F1 – macro và Accuracy để đánh giá mô hình.

**F\_score**: là một trung bình điều hòa(harmonic mean) của các tiêu chí Preciѕion ᴠà Recall. Nó có хu hướng lấу giá trị gần ᴠới giá trị nào nhỏ hơn giữa 2 giá trị Preciѕion ᴠà Recall ᴠà đồng thời nó có giá trị lớn nếu cả 2 giá trị Preciѕion ᴠà Recall đều lớn.

**Accuracy**: là tỉ lệ giữa số điểm được phân loại đúng và tổng số điểm.

# **IV. Kết quả thực nghiệm**

**1. Kết quả trên tập phát triển**

Sau khi thực hiện tinh chỉnh các tham số trên tạp phát triển, chúng tôi có được kết quả của các mô hình như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Score Encoder** | | **Tham số model** | **Accuracy** | **F1 - macro** |
|
| Logistic Regression | CountVertorizer | ngram\_range(1, 1) | Tham số mặc định | 82.45 | 75.33 |
| GridSearchCV | 82.68 | 75.98 |
| ngram\_range(1, 2) | Tham số mặc định | **84.43** | **78.80** |
| GridSearchCV | **84.43** | **78.80** |
| Tf-idfVectorizer | ngram\_range(1, 1) | Tham số mặc định | 84.03 | 76.08 |
| GridSearchCV | 83.15 | 76.09 |
| ngram\_range(1, 2) | Tham số mặc định | 83.96 | 74.52 |
| GridSearchCV | 85.13 | 78.10 |

Bảng 4. kết quả của các bộ tham số trên mô hình Logistic Regression trên tập phát triển

Chúng tôi tìm ra bộ tham số mô hình tốt nhất là

**penalty: L2, C: 1.0, solver: newton-cg** và tham số mặc định khi dùng chiến lược GridSearchCV và thkết hợp với phương pháp mã hóa CountVectorizer ngram\_range (1,2)

và cho kết quả tốt nhất với kết quả 84.43% ở độ đo accuracy và 78.8% ở độ đo F1\_Score.

Do đó, chúng tôi quyết định sử dụng bộ tham số này để đánh giá trên tập test.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Score Encoder** | | **Tham số model** | **Accuracy** | **F1 - macro** |
|
| KNN | CountVertorizer | ngram\_range(1, 1) | Tham số mặc định | 67.55 | 64.06 |
| GridSearchCV | **74.6** | **68.17** |
| ngram\_range(1, 2) | Tham số mặc định | 67.55 | 64.06 |
| GridSearchCV) | **74.6** | **68.17** |
| Tf-idfVectorizer | ngram\_range(1, 1) | Tham số mặc định | 35.03 | 34.29 |
| GridSearchCV | 41.81 | 41.8 |
| ngram\_range(1, 2) | Tham số mặc định | 35.03 | 34.29 |
| GridSearchCV | 41.81 | 41.8 |

Bảng 5. kết quả của các bộ tham số trên mô hình KNN trên tập phát triển

Với mô KNN khi sử dụng phương pháp mã hóa CountVector với bộ tham số ngram\_range=(1,1) và bộ tham số ngram\_range=(1,2) kết hợp với chiến lược tinh chỉnh tham số mô hình GridSearchCV với bộ tham số **n\_neighbors= 2,   
  
  
p=2 Weights = ‘uniform’**cho kết quả bằng nhau và tốt nhất với kết quả 74.6% ở độ đo accuracy và 68.17% ở độ đo F1\_Score.

Do đó, chúng tôi quyết định sử dụng bộ tham số này để đánh giá trên tập test.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Score Encoder** | | **Tham số model** | **Accuracy** | **F1 - macro** |
|
| SVM | CountVertorizer | ngram\_range(1, 1) | Tham số mặc định | 84.93 | 78.09 |
| GridSearchCV | 84.36 | **78.18** |
| ngram\_range(1, 2) | Tham số mặc định | 85.07 | 77.95 |
| GridSearchCV | 85.13 | 77.24 |
| Tf-idfVectorizer | ngram\_range(1, 1) | Tham số mặc định | **85.20** | 77.64 |
| GridSearchCV | 83.62 | 76.07 |
| ngram\_range(1, 2) | Tham số mặc định | 84.36 | 75.32 |
| GridSearchCV | 84.80 | 77.75 |

Bảng 6. kết quả của các bộ tham số trên mô hình SVM trên tập phát triển

Ở bảng trên ta thấy, với mô hình SVM, đối với độ đo Accuracy có kết quả cao nhất là 85.2% với tham số mặc định và phương pháp mã hóa TF-IDF Vectorizer.

Trong khi đó, ở độ đo F1\_score kết quả cao nhất đạt được thuộc về chiến lược tinh chỉnh tham số GridSearchCV với bộ tham số **C: 100, gamma: 0.001, kernel: rbf** và phương pháp mã hóa CountVectorizer cùng bộ tham số ngram\_range(1,1) cho kết quả 78.18%.

Vì bộ dữ liệu của chúng tôi mất cân bằng giữa số lượng của 2 nhãn do đó, chúng tôi quyết định mô hình này sẽ sử dụng bộ tham số cho kết quả F1\_score cao nhất để đánh giá kết quả trên tập test.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Score Encoder** | | **Tham số model** | **Accuracy** | **F1 - macro** |
|
| Naïve Bayes | CountVertorizer | ngram\_range(1, 1) | Tham số mặc định model | **83.22** | **75.35** |
| GridSearchCV(cv = 10) | 83.05 | 74.78 |
| ngram\_range(1, 2) | Tham số mặc định model | 76.61 | 53.41 |
| GridSearchCV(cv = 10) | 78.49 | 60.3 |
| Tf-idfVectorizer | ngram\_range(1, 1) | Tham số mặc định model | **83.22** | **75.35** |
| GridSearchCV(cv = 10) | 83.05 | 74.78 |
| ngram\_range(1, 2) | Tham số mặc định model | 76.61 | 53.41 |
| GridSearchCV(cv = 10) | 78.49 | 60.3 |

Bảng 7. kết quả của các bộ tham số trên mô hình Naïve Bayes trên tập phát triển

Với mô hình KNN khi sử dụng tham số mặc định của mô hình kết hợp với từng phương pháp trong 2 phương pháp mã hóa cho kết quả bằng nhau và tốt nhất với kết quả 83.22% ở độ đo accuracy và 75.35% ở độ đo F1\_Score.

Do đó, chúng tôi quyết định sử dụng bộ tham số này để đánh giá trên tập test

.

**2.Kết quả trên tập kiểm thử (tập test)**

|  |  |  |
| --- | --- | --- |
| Mô hình | Độ chính xác (Accuracy | Độ đo F1 |
| Logistic Regression(tham số mặc định) | 85.47 | 80.19 |
| Logistic Regression (penalty: L2, C: 1.0, solver: newton-cg) | **85.51** | **80.22** |
| SVM C: 100, gamma: 0.001, kernel: rbf | 85.44 | 80.19 |
| KNN n\_neighbors= 2, p=2 Weights = ‘uniform’ | 74.34 | 67.64 |
| Naïve Bayes(tham số mặc định) | 83.33 | 75.46 |

Bảng 5. Kết quả trên tập test của các mô hình cùng với bộ tham số tốt trên tập phát triể

Thông qua bảng trên, cho thấy ở bài toán này và với bộ dữ liệu của chúng tôi, mô hình Logisti

Regression cho kết quả tốt nhất với 85.51% ở độ đo accuracy và 80.22% với độ độ F1\_score và kém hiệu quả nhất là mô hình KNN.

# **V. PHÂN TÍCH LỖI VÀ HƯỚNG PHÁT TRIỂN**

1. Phân tích lỗi

Sau khi kiểm tra lại kết quả, chúng tôi thấy các mô hình máy học còn có những dự đoán sai, do những nguyên nhân sau: mô hình học máy chưa hiểu được ngữ nghĩa của bình luận, các bước tiền xử lý chưa được hiệu quả; bộ dữ liệu mất cân bằng.  
2. Hướng phát triển

Ở bài toán này, chúng tôi còn gặp một số khó khăn do đó chúng tôi có những định hướng trong tương lai như sau: Mở rộng bộ dataset, thử nghiệm các phương pháp tiền xử lý mới, thử nghiệm trên các mô hình học sâu kết hợp tinh chỉnh các mô hình.

# **VI. KẾT LUẬN**

|  |  |
| --- | --- |
| Đỗ Phạm Phúc Tính | - Phân công công việc cho nhóm  - Viết báo cáo  -chạy mô hình  - thuyết trình |
| Cao Đình Duy Ngọc | - Xử lý dữ liệu  - Chạy mô hình  - Viết báo cáo  -- Thuyết trình |
| Ngô Huỳnh Trưởng | - chạy mô hình  - Làm slide  - Thuyết trình |
| Nguyễn Trần Gia Thế | - Chạy mô hình  - Làm slide  - Viết báo cáo  - Tổng hợp và chiếu slide |

Trong bài báo này chúng tôi đã kế thừa tập dữ liệu ViSpam Detection [1] với khoảng

19.870 đánh giá tiếng Việt. Chúng tôi đã đề xuất

các phương pháp học máy để nhận biết các đánh giá spam tiếng Việt. Qua đó với bài toán này và tập dữ liệu của chúng tôi kế thừa, mô hình Logistic Regression [2] cho kết quả đạt 85.51% đối với độ đo accuracy và 80.22% đối với độ đo F1-score.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | Co Van Dinh, Son T. Luu, Anh Gia-Tuan Nguyen, "Detecting Spam Reviews on Vietnamese E-commerce Websites". |
| [2] | Hoss Belyadi, Alireza Haghighat, in *Machine Learning Guide for Oil and Gas Using Python*, pp. 169-295. |
| [3] | "COMPUTER SCIENCE," [Online]. Available: https://neralnetwork.wordpress.com/2018/05/11/thuat-toan-support-vector-machine-svm/. |
| [4] | "Codelearn," [Online]. Available: https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn. |
| [5] | Gilberto Souza, Departamento de Engenharia Mecânica Escola Politécnica da USP, Arthur Melani, Miguel Michalski, Renan Silva, in *Reliability Analysis and Asset Management of Engineering Systems*, pp. 165-187. |