**TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN**

**──────── \* ────────**



**BÁO CÁO**

**ĐỀ TÀI:**

**BÀI TOÁN PHÂN LOẠI VĂN BẢN**

Sinh viên thực hiện: **Thảo Hương**

Nội dung

[LỜI NÓI ĐẦU 3](#_Toc10817619)

[Chương 1. Tìm hiểu bài toán phân loại văn bản 4](#_Toc10817620)

[1.1. Giới thiệu 4](#_Toc10817621)

[1.2. Phát biểu bài toán tổng quát 4](#_Toc10817622)

[1.3. Bài toán phân loại comment đánh điện thoại và các thiết bị liên quan dựa trên tính tích cực tiêu cực 4](#_Toc10817623)

[Chương 2. Giải quyết bài toán phân loại văn bản 5](#_Toc10817624)

[2.1. Thu thập dữ liệu 6](#_Toc10817625)

[2.2. Chuẩn bị dữ liệu 6](#_Toc10817626)

[2.2.1. Làm sạch dữ liệu 6](#_Toc10817627)

[2.2.2. Biến đổi dữ liệu 7](#_Toc10817628)

[2.3. Xây dựng mô hình 8](#_Toc10817629)

[2.3.1. Thuật toán Naive Bayes 8](#_Toc10817630)

[2.4. Đánh giá mô hình 10](#_Toc10817631)

[TÀI LIỆU THAM KHẢO 12](#_Toc10817632)

[SOURCE CODE CHƯƠNG TRÌNH 12](#_Toc10817633)

# LỜI NÓI ĐẦU

Phân loại văn bản là một trong những vấn đề quan trọng trong lĩnh vực xử lý ngôn ngữ. Nhiệm vụ của loại bài toán này là gán các tài liệu văn bản vào các chủ đề cho trước. Trên thực tế, loại bài toán này khá phổ biến. Ví dụ như một nhà phân tích trong lĩnh vực nào đó cần phải tổng hợp rất nhiều tài liệu, bài viết về lĩnh vực đó để đọc và đưa ra phân tích của mình. Tuy nhiên, số lượng các tài liệu là rất lớn, đặc biệt với sự phát triển mạnh mẽ của internet như hiện nay, khiến cho việc đọc hết được tất cả các tài liệu là vô cùng khó khăn và mất nhiều thời gian đối với nhà phân tích. Vì vậy, cần có một hệ thống phân loại các tài liệu thành các chủ đề để nhà phân tích làm việc hiệu quả hơn. Một ví dụ khác trong thực tế là việc phân loại spam email. Nếu người dùng phải đọc hết mọi thư đến thì sẽ tốn rất nhiều thời gian vì spam email chiếm số lượng lớn. Vì vậy, cần có một hệ thống phân loại đâu là email spam và đâu là email không spam.

Có rất nhiều phương pháp được đưa ra để giải quyết loại bài toán này như: thuật toán Naive Bayes, KNN, Decision Tree, Neural Network và SVM,… Mỗi phương pháp đều có ưu nhược điểm riêng đối với từng bài toán cụ thể và đều cho ra những kết quả khả quan. Tuy nhiên trong khuôn khổ đề tài, em ưu tiên sử dụng phương pháp Naive Bayes để giải quyết một **“Bài toán phân loại văn bản”** cụ thể.

Để hoàn thành đề tài này, em xin gửi lời cảm ơn chân thành đến <who?> đã hết lòng giúp đỡ, hướng dẫn, chỉ dạy tận tình. Do đây là lần đầu tiếp xúc với Machine Learning để giải quyết một bài toán cụ thể có ý nghĩa thực tiễn, sai sót trong kiến thức, cách xử lý,… là điều không thể tránh khỏi. Em kính mong nhận được sự góp ý, nhận xét, chỉnh sửa của thầy để bài toán được hoàn thiện hơn.

# Chương 1. Tìm hiểu bài toán phân loại văn bản

## 1.1. Giới thiệu

Hiện nay, với sự xuất hiện và phát triển mạnh mẽ của Internet, sự bùng nổ thông tin diễn ra liên tục hàng ngày làm cho không gian dữ liệu gia tăng thường xuyên. Phương pháp sử dụng các tài liệu, văn bản giấy dần được số hóa do có nhiều tính năng vượt trội như: có thể lưu trữ lâu dài, cập nhât, sửa đổi, tìm kiếm một cách dễ dàng và nhanh chóng. Do đó số lượng văn bản số ngày nay đang tăng dần theo cấp số nhân. Cùng với sự gia tăng của số lượng văn bản, nhu cầu tìm kiếm văn bản cũng tăng theo. Do đó, vấn đề phân loại văn bản tự động được đặt ra như một bài toán quan trọng để giảm thiểu công sức, giúp người dùng có thể tìm kiếm nhanh chóng, thuận tiện.

## 1.2. Phát biểu bài toán tổng quát

Bài toán phân loại văn bản có thể được phát biểu như sau: Cho trước một tập văn bản D = {d1, d2,… dn} và tập các chủ đề (nhãn) cho trước C = {c1, c2,… cm}. Mục tiêu của bài toán là gán văn bản di vào nhãn cj cho trước.

## 1.3. Bài toán phân loại comment đánh điện thoại và các thiết bị liên quan dựa trên tính tích cực tiêu cực

Việc đọc hiểu được tính tích cực/tiêu cực của các bình luận đó có ý nghĩa quan trọng đối với các nhà phân tích và nhà phát hành sản phẩm, để họ hiểu được xu hướng, thị hiếu của người mua. để rồi cải thiện trong các sản phẩm sau này. Tuy nhiên việc đọc từng comment, review trong một không gian rất lớn kia tốn rất nhiều thời gian và công sức. Vì vậy, ở đây ta nghiên cứu bài toán phân loại comment đánh điện thoại và các thiết bị liên quan dựa trên tính tích cực tiêu cực để giải quyết vấn đề trên, cũng như để hiểu một bài toán phân loại văn bản nói chung (Các bước ta giải quyết bài toán cụ thể này có thể được tổng quát hóa để giải quyết các bài toán phân loại văn bản khác).

Phát biểu bài toán này như sau: Cho trước một tập review phim R = {r1, r2,… rn} và tập 2 nhãn C = {“tích cực”, “tiêu cực”}. Mục tiêu của bài toán là gán review ri vào nhãn “tích cực” hoặc “tiêu cực”.

# Chương 2. Giải quyết bài toán phân loại văn bản

Chương này sẽ nêu ra quy trình và các kĩ thuật để xử lý một bài toán phân loại văn bản. Ở đây ta sẽ giải quyết bài toán cụ thể: phân loại review phim dựa trên tính tích cực tiêu cực đã nêu ở Chương 1, qua đó phân tích, nêu ra các bước cần xử lý trong một bài toán phân loại văn bản tổng quát và một số kĩ thuật có thể áp dụng ở từng bước đó.

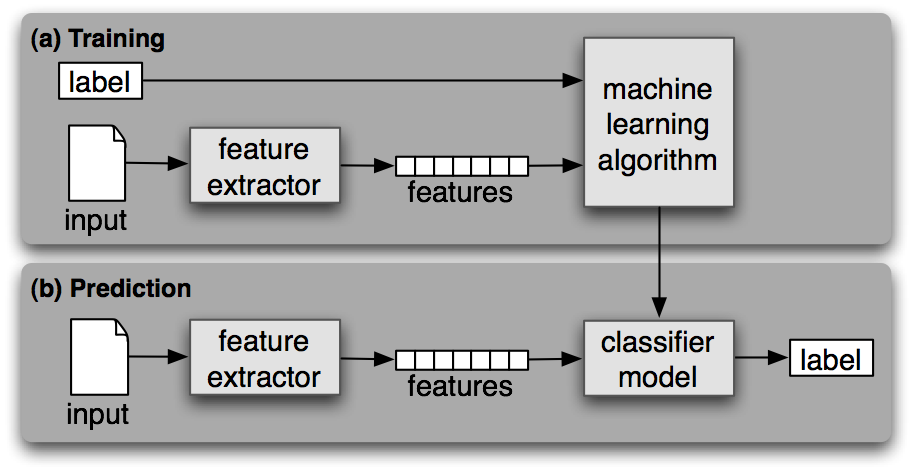


Figure 1: Supervised Classification (Nguồn: <https://www.nltk.org/book/ch06.html>)

(a) Trong quá trình training, một trình trích xuất đặc trưng được sử dụng để chuyển đổi từng input thành một bộ đặc trưng. Các bộ đặc trưng này, nắm bắt thông tin cơ bản về từng đầu vào nên được sử dụng để phân loại, sẽ được thảo luận trong phần tiếp theo. Các cặp bộ đặc trưng và nhãn được đưa vào thuật toán học máy để tạo mô hình.

(b) Trong quá trình dự đoán, trình trích xuất đặc trưng tương tự được sử dụng để chuyển đổi các input không nhìn thấy (tức là chưa có nhãn) thành các bộ đặc trưng. Các bộ đặc trưng này sau đó được đưa vào mô hình, tạo ra các nhãn dự đoán.

Bài toán ta đang xét có trước tập train và nhãn cho tập train đó, nên ta sử dụng mô hình học có giám sát để giải quyết bài toán. Ta sẽ nói kĩ hơn trong các phần tiếp theo.

Một số bước chính để giải quyết bài toán (đã lược qua một số bước), đầu ra của bước trước đó chính là đầu vào của bước sau tiếp theo:

* Thu thập dữ liệu (Data collection)
* Chuẩn bị (Tiền xử lý) dữ liệu (Data preparation)
* Xây dựng mô hình (Modeling)
* Đánh giá mô hình (Evaluation)

## 2.1. Thu thập dữ liệu

Trong giai đoạn thu thập dữ liệu ban đầu, ta cần xác định và thu thập những nguồn dữ liệu có sẵn - có cấu trúc, không cấu trúc hoặc bán cấu trúc - liên quan đến lĩnh vực của vấn đề.

Có thể phải đầu tư thêm để có được những yếu tố dữ liệu không thông dụng cho đến khi biết được nhiều hơn về dữ liệu và mô hình. Nếu có lỗ hổng trong việc thu thập dữ liệu, ta có thể phải xem lại yêu cầu dữ liệu phù hợp và thu thập thêm dữ liệu mới.

Việc thu thập nguồn dữ liệu có sẵn có thể thông qua các công cụ tìm kiếm thông dụng như Google,… Việc mua dữ liệu có thể được thực hiện thông qua việc giao dịch với các bên cung cấp dữ liệu

Ở bài toán này, em đã crawl các comment về điện thoại cũng như các sản phẩm liên quan như: tai nghe, loa, ốp,… từ trang web thegioididong.com làm dữ liệu phục vụ cho việc train mô hình cũng như đánh giá mô hình.

**Mô tả dữ liệu:**

Tập dữ liệu này bao gồm tổng cộng hơn 35000 bình luận (kèm với đó là số sao đánh giá), em chia tập dữ liệu này thành 3 lớp: “pos” (tích cực) chứa các bình luận có đánh giá lớn hơn 3 star, “neu” (trung bình) chứa các bình luận có đánh giá bằng 3 star, “neg” (tiêu cực) chứa các bình luận có đánh giá nhỏ hơn 3 star.

Như vậy có 29725 bình luận tích cực, 3030 bình luận tiêu cực và 2455 bình luận tiêu cực. Em chia dữ liệu train:test theo tỷ lệ 2:1.

## 2.2. Chuẩn bị dữ liệu

Bước này bao gồm mọi hành động để xây dựng tập dữ liệu mà sẽ được dùng trong bước mô hình hóa tiếp theo. Các hoạt động chuẩn bị dữ liệu bao gồm làm sạch dữ liệu (data cleaning) (xử lý giá trị thiếu hoặc không hợp lệ, loại bỏ trùng lặp, định dạng đúng cách), kết hợp dữ liệu từ nhiều nguồn (files, tables, platforms) và biến đổi dữ liệu thành dạng hữu ích hơn.

### 2.2.1. Làm sạch dữ liệu

Ví dụ về một văn bản trong tập dataset:

Thiết kế đẹp, có lúc tay bị ướt nhưng vẫn sử dụng được phím cảm ứng màn hình. Nhân viên lịch sự, nhiệt tình

Ở đây dữ liệu ta có được là khá “sạch”. Thông thường văn bản được liệt kê không có cấu trúc, có thể lẫn các HTML tag, code JS, các số, các kí tự không có ý nghĩa (!@#$%^&\*)… đó chính là các noise (nhiễu), chúng ta cần phải làm sạch các nhiễu đó. Ngoài ra, các từ viết hoa hay viết thường, một hay một vài kí tự đều nên được chuyển về một dạng (cùng viết hoa hoặc viết thường).

Ta có thể sử dụng thư viện gensim hay một số thư viện khác để làm sạch dữ liệu.

Kết quả thu được của đoạn văn bản trên sau khi được làm sạch:

thiết kế đẹp có lúc tay bị ướt nhưng vẫn sử dụng được phím cảm ứng màn hình nhân viên lịch sự nhiệt tình

### 2.2.2. Biến đổi dữ liệu

Dù dữ liệu đã được làm sạch nhưng máy tính vẫn không thể hiểu được từng từ có ý nghĩa như nào, hơn nữa máy tính xử lý số học nhanh hơn nhiều so với xữ lý chuỗi kí tự, vì vậy ta cần chuyển đổi các văn bản thành dạng mà máy tính có thể hiểu được và xử lý tốt.

Word Embedding là tên gọi chung của các mô hình ngôn ngữ và các phương pháp học theo đặc trưng trong Xử lý ngôn ngữ tự nhiên, ở đó các từ hoặc cụm từ được ánh xạ sang các vector số (thường là số thực).

Word Embedding bao gồm nhiều phương pháp như word2vec, Co-occurrence Matrix , GloVe (Global Vector),… Trong bài toán này, ta sử dụng phương pháp BoW (Bag of Words). Ta lấy từ làm đơn vị nhỏ nhất để phân tách văn bản, như vậy mỗi văn bản là một vector nhiều chiều, mỗi chiều biểu thị trọng số của một từ. Số chiều của vector chính là số các từ có trong tập data, tập hợp số các từ này đóng vai trò là một túi từ (bag of words).

Một số kĩ thuật thường thấy để xác định trọng số của từ là:

* **Hot-one**:

Nếu từ đó xuất hiện trong văn bản thì ta đánh dấu bit 1, ngược lại trọng số của từ đó là 0. Với cách làm này, một từ xuất hiện nhiều với một từ xuất hiện rất ít trong một văn bản là tương đương nhau về mặt trọng số hay giá trị phân loại, điều này là không hợp lý và có thể làm giảm độ chính xác, ví vậy trong bài toán này ta nghiên cứu hai kĩ thuật xác định trọng số khác như sau:

* **TF (Term Frequency):**

Trọng số của một từ trong văn bản có giá trị bằng số lần xuất hiện của từ đó trong văn bản đang xét. Cách làm này về lý thuyết sẽ giúp làm tăng độ chính xác do giá trị phân loại của các từ nay đã được xác định một cách chính xác hơn. Tuy nhiên, kĩ thuật này vẫn còn một nhược điểm, đó chính là, có những từ xuất hiện ở nhiều ở hầu hết các văn bản, điều này làm cho giá trị phân loại của những từ này thấp hơn các từ khác, trong khi trọng số của chúng vẫn lớn.

* **TF-IDF (Term Frequency - Inverse Document Frequency):**

Trọng số của một từ trong văn bản được xác định bởi công thức:

tfidf(t,d,D) = tf(t,d) x idf(t,D)

trong đó:

* + tf(t,d) = Số lần xuất hiện từ t / Tổng số từ trong văn bản d
  + idf(t,d) = log (Tổng số văn bản / Số văn bản chứa từ t)

Cách làm này về lý thuyết, cho ta đánh giá đủ đúng về trọng số của một từ, tức giá trị phân loại của từ đó.

Ở đây ta đánh giá trọng số của từ đó (đặc trưng) theo phương pháp TF-IDF.

## 2.3. Xây dựng mô hình

Để xây dựng mô hình dự đoán, ta sử dụng một tập huấn luyện (training set) và một hoặc nhiều thuật toán học máy. Quá trình mô hình hóa có thể phải được lặp lại liên tiếp để đạt được những hiểu biết trung gian, dẫn đến sàng lọc trong chuẩn bị dữ liệu. Có thể thử nhiều thuật toán với thông số khác nhau để tìm ra mô hình tốt nhất.

Trong khuôn khổ của môn học, ta nghiên cứu hai thuật toán sau:

### 2.3.1. Thuật toán Naive Bayes

#### 2.3.1.1. Cơ sở lý thuyết:

Xét bài toán phân lớp với tập các lớp là C = {c1, c2,… cn}. Giả sử có một điểm dữ liệu (biểu diễn dưới dạng vector d chiều) . Cụ thể ta cần tính:

p(y = c|x), viết tắt thành p(c|x). Tức tính xác suất đầu ra là lớp c biết đầu vào là vector x.

Việc tính được biểu thức trên sẽ giúp ta xác định được xác suất điểu dữ liệu trên rơi vào từng class, từ dó ta có thể phân lớp (gán nhãn) cho dữ liệu đó vào lớp có xác suất lớn nhất:

c = argmax p(c|x) với C {c1, c2,… cn}

Biểu thức trên thường khó tính được trực tiếp. Đây là lúc định lý Naïve Bayes được áp dụng:

c = argmax p(c|x)

= argmax p(x|c)p(c)/p(x)

= argmax p(x|c)p(c) (Do p(x) không phụ thuộc vào c)

p(c) có thể tính được theo tỉ lệ điểm dữ liệu trong tập training rơi vào lớp c chia cho tổng số lượng dữ liệu trong tập training.

Ta có: p(x|c) = p(x1,x2,…,xd|c) = p(x2,…,xd|cx1) \* p(x1) = ….

Như ta thấy nếu khai triển toàn bộ công thức trên, p(x|c) thường rất khó tính toán do x là một biến ngẫu nhiên nhiều chiều, cần rất nhiều dữ liệu training để có thể xây dựng phân phối đó. Để giúp bài toán đơn giản, người ta thường giả sử các thành phần của biến ngẫu nhiên x là độc lập với nhau nếu biết c, tức là:

p(x|c) = p(x1,x2,…,xd|c) =

Giả thiết các chiều của dữ liệu độc lập với nhau nếu biết c là quá chặt gần như không có dữ liệu nào mà các thành phần hoàn toàn độc lập với nhau. Tuy nhiên trong thực tế, giả thiết *naive* này lại mang đến kết quả tốt.

Naïve Bayes Classification đơn giản, có tốc độ train và test nhanh, mang lại hiệu quả trong các bài toán có dữ liệu lớn.

Ở bước train, p(c) và p(xi|c), i = 1,2,…d sẽ được xác định dựa vào tập training data.

Ở bước test, với một điểm dữ liệu mới x, lớp của nó được xác định bởi:

c = argmax p(c)  với C {c1, c2,… cn}

Khi d lớn, các xác suất nhỏ, giá trị của vế phải trong biểu thức trên sẽ là một số rất nhỏ, khi tính toán có thể gặp sai số. Do vậy, ta lấy log vế phải

c = argmax log(p(c)) +

Việc này không ảnh hưởng tới kết quả vì log là một hàm đồng biến trên tập các số dương.

**Các phân phối thường dùng cho p(xi|c):**

* Gaussian Naive Bayes
* Bernoulli Naive Bayes
* Multinomial Naive Bayes

dựa theo luật phân phối của p(xi|c) mà ta sử dụng các phân phối cho hợp lý.

Ở trong bài toán này, ta sử dụng Multinomial Naive Bayes. Lúc này, mỗi văn bản được biểu diễn bởi một vector có độ dài d chính là số từ trong từ điển. Giá trị của thành phần thứ i trong mỗi vector chính là số lần từ thứ i xuất hiện trong văn bản đó.

/\* Có thể bỏ qua đoạn này nếu cậu thấy ko cần thiết--------

Khi đó, p(xi|c) tỉ lệ với tần suất từ thứ i xuất hiện trong các văn bản của lớp c, giá trị này có thể tính bằng cách:

λci=p(xi|c)=Nci/Nc    
Trong đó :

* Nci là tổng số lần từ thứ i xuất hiện trong các văn bản của lớp c.
* Nc là tổng số từ (kể cả lặp) xuất hiện trong lớp c. Nói cách khác, nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào lớp c. Có thể suy ra rằng: Nc=∑i=1,..d Nci,từ đó ∑i=1,..d λci = 1

Cách tính này có một hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong lớp c thì biểu thức trên sẽ bằng 0, dẫn đến kết quả không chính xác. Để giải quyết vẫn đề này, ta sử dụng kĩ thuật *Laplace smoothing*:

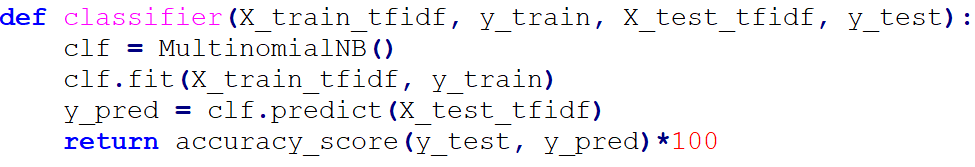
λ’ci= (Nci +α)/(Nc + dα)

với α là một số dương, thường lấy = 1 để tránh trường hợp tử số bằng 0. Mẫu số được cộng với dα để đảm bảo ∑i=1,..d λ’ci = 1.

-----------------------------------------------------------------------end\*/

#### 2.3.1.2. Áp dụng lý thuyết để xây dựng mô hình:

Sau khi đã ánh xạ toàn bộ văn bản sang các vector theo phương pháp biểu diễn đã nói trước đó, ta xây dựng mô hình Multinomial Naïve Bayes dựa trên cơ sở lý thuyết trên như sau:



Sau khi đã huấn luyện mô hình với dữ liệu, ta thu được classifier. Với mỗi văn bản mới, ta có thể tiền xử lý dữ liệu, ánh xạ văn bản sang dạng các vector rồi đưa qua classifier, ta sẽ nhận được output là nhãn dự đoán cho văn bản đó.

## 2.4. Đánh giá mô hình

Trước khi triển khai, mô hình cần được đánh giá để biết chất lượng và đảm bảo nó đúng, đầy đủ đối với bài toán đang xét. Đánh giá mô hình đòi hỏi tính toán các phương pháp chẩn đoán khác nhau và các đầu ra khác nhau như bảng, đồ thị, cho phép giải thích chất lượng của dữ liệu và hiệu quả của nó trong việc giải quyết vấn đề. Đối với một mô hình dự đoán, một tập kiểm tra (testing set) độc lập với tập huấn luyện nhưng tuân theo phân phối xác suất tương tự và có một kết quả đã biết được sử dụng để đánh giá mô hình. Đôi khi, mô hình cuối cùng còn được ứng dụng để xác nhận đánh giá cuối cùng.

Trong bài toán ta đang xét, ta sẽ sử dụng tập test data được cung cấp sẵn trong tập data set để đánh giá từng mô hình phân loại. Kết quả về độ chính xác mà từng mô hình dự đoán được được biểu diễn ở bảng sau:

Ta có nhận xét: Đối với bài toán phân loại comment về điện thoại và các sản phẩm liên quan dựa trên tính tích cực tiêu cực và với tập dữ liệu này, mô hình MultinomialNaiveBayes phương pháp tính trọng số TF-IDF giúp ta đạt được độ chính xác lớn nhất là 71.60%

# TÀI LIỆU THAM KHẢO

1.<https://www.ibmbigdatahub.com/blog/why-we-need-methodology-data-science>

2.<https://machinelearningcoban.com/>

3.<https://codetudau.com/bag-of-words-tf-idf-xu-ly-ngon-ngu-tu-nhien/index.html>

4.<https://viblo.asia/p/phan-loai-van-ban-tu-dong-bang-machine-learning-nhu-the-nao-4P856Pa1ZY3>

5.<https://viblo.asia/p/so-luoc-word-embedding-gDVK2RAeKLj>

# SOURCE CODE CHƯƠNG TRÌNH

<https://github.com/thaohuongg/Vietnamese-Comments-Sentiment-Classification>