TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN

Khoa Công nghệ thông tin

# 

**Báo cáo bài tập lớn môn Khai Phá Dữ Liệu**

Giảng viên : Ts. Trần Mai Vũ

Nhóm 7

|  |  |
| --- | --- |
| Lê Đức Thắng | 18021160 |
| Tạ Quang Ngọc | 18020952 |
| Nguyễn Đức Thắng | 18021145 |
| Thái Phi Hoàng | 18020019 |

Mục lục

[**Tổng quát bài toán phân loại văn bản**](#_4oarg3e4ft11) **3**

[**Phân tích dữ liệu**](#_k8ipig9ww3vm) **4**

[Thống kê các nhãn](#_j21vozi499am) 4

[Nhận xét về dữ liệu](#_pozxhw61u0ar) 4

[**Phương pháp giải quyết**](#_w5me28wrbisj) **5**

[Dựa theo độ chính xác](#_ufsf5glz8yw7) 5

[Dựa vào gom nhóm](#_i3bs2qtbtxjl) 6

[Chuẩn hóa dữ liệu](#_tm1lz8b5n799) 7

[Xây dựng dữ liệu train/test](#_gam088weibxr) 8

[Solution fastText](#_no8d7dyezl4j) 8

[Solution Logistic Regression](#_mrp8rwiumvtj) 9

[Solution SVM](#_jstygk16t8qz) 10

[**Đánh giá các mô hình**](#_vwp81gc39ncp) **11**

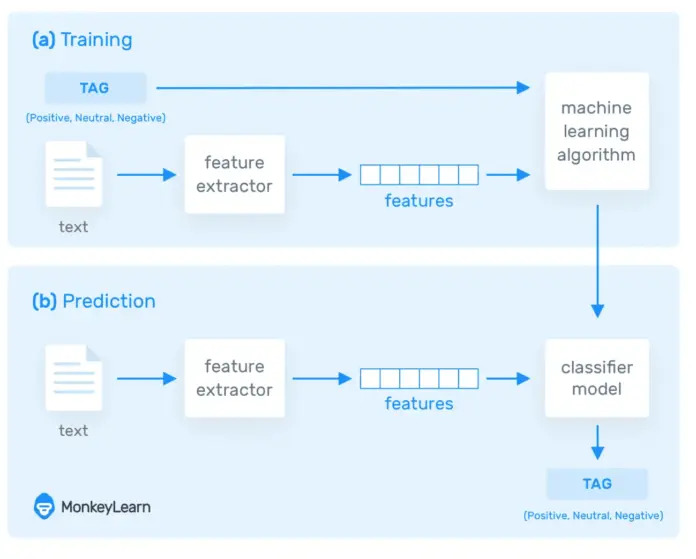
[So sánh kết quả của các mô hình](#_1k0msru58o98) 11

[Nhận xét và đánh giá](#_1k0msru58o98) 11

[**Tài liệu tham khảo**](#_569ew8ihofc1) **12**

# Tổng quát bài toán phân loại văn bản

Phân loại văn bản (Text classification) là một bài toán phổ biến trong xử lý ngôn ngữ tự nhiên (Natural language processing). Là bài toán thuộc nhóm học có giám sát (Supervised learning) trong học máy. Bài toán này yêu cầu dữ liệu cần có nhãn (label). Mô hình sẽ học từ dữ liệu có nhãn đó, sau đó được dùng để dự đoán nhãn cho các dữ liệu mới mà mô hình chưa gặp.

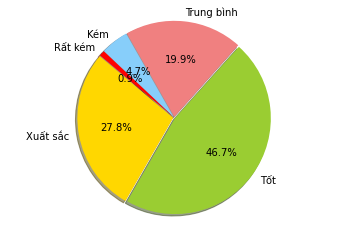
Hình ảnh dưới đây sẽ cho chúng ta cái nhìn tổng quát về cách hoạt động của một bài toán phân loại văn bản:

* Giai đoạn (a): Huấn luyện (training) là giai đoạn học tập của mô hình phân loại văn bản. Ở bước này, mô hình sẽ học từ dữ liệu có nhãn (trong ảnh trên nhãn là Positive, Negative, Neutral). Dữ liệu văn bản sẽ được số hóa thông qua bộ trích xuất đặc trưng (feature extractor) để mỗi mẫu dữ liệu trong tập huấn luyện trở thành 1 vector nhiều chiều (đặc trưng). Thuật toán máy học sẽ học và tối ưu các tham số để đạt được kết quả tốt trên tập dữ liệu này. Nhãn của dữ liệu được dùng để đánh giá việc mô hình học tốt không và dựa vào đó để tối ưu.
* Giai đoạn (b): Dự đoán (prediction), là giai đoạn sử dụng mô hình học máy sau khi nó đã học xong. Ở giai đoạn này, dữ liệu cần dự đoán cũng vẫn thực hiện các bước trích xuất đặc trưng. Mô hình đã học sau đó nhận đầu vào là đặc trưng đó và đưa ra kết quả dự đoán.

# Phân tích dữ liệu

## Thống kê các nhãn

|  |  |
| --- | --- |
| Xuất sắc | 6396 |
| Tốt | 10735 |
| Trung bình | 4569 |
| Kém | 1085 |
| Rất kém | 214 |



## Nhận xét về dữ liệu

* Mặc dù các dữ liệu đã được tiền xử lý nhưng vẫn chưa đồng nhất về định dạng cần phải xử lý tiếp mới đem lại được hiệu quả cao nhất. Ví dụ chưa đồng nhất về kiểu gõ dấu, chưa đồng nhất về dạng Unicode.
* Có khá nhiều dữ liệu khó xử lý . Tiêu biểu như tiếng nước ngoài gồm : Trung, Anh,..Còn một số từ viết tắt cũng làm cho việc phân tích và hiểu dữ liệu trở nên khó khăn.Còn nhiều ký tự thừa không cần thiết gây nhiễu trong việc phân tích và hiểu dữ liệu.
* Có một số dữ liệu mặc dù nhận xét rất tốt nhưng vẫn gán nhãn trung bình, kém, rất kém và ngược lại. Điều này làm khó khăn cho việc hiểu dữ liệu. Dễ làm hiểu sai vấn đề.

# Phương pháp giải quyết

## Dựa theo độ chính xác

Ban đầu chúng em tìm những luật có độ chính xác trên 80%. Ví dụ như:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Từ xuất hiện | label | Số lượng | Xác suất |
| 1 | Xuất sắc | Xuất sắc | 1435 | 100% |
| 2 | Tuyệt hảo | Xuất sắc | 489 | 100% |
| 3 | Excellent | Xuất sắc | 180 | 78.3% |
| 4 | amazing(1) | Xuất sắc | 75 | 81.3% |
| 5 | fantastic(1) | Xuất sắc | 58 | 93.1% |
| 6 | perfect(1) | Xuất sắc | 157 | 80.25% |
| 7 | wonderful(1) | Xuất sắc | 60 | 80.67% |
| 8 | “. Tốt” | Tốt | 1596 | 97.74% |
| 9 | ". Rất tốt" | Tốt | 503 | 89.26% |
| 10 | ". Rất tốt","quay lại","rất","hài lòng" | Tốt | 440 | 96.13% |
| 11 | “. Tuyệt vời" | Tốt | 610 | 82.78% |
| 12 | ". Hài lòng" | Tốt | 254 | 74.41% |
| 13 | ". Dễ chịu" | Trung bình | 481 | 100% |
| 14 | ". binh thuong"(2) | Trung bình | 86 | 100% |
| 15 | ". Tàm tạm" | Trung bình | 256 | 100% |
| 16 | ". Chấp nhận được" | Trung bình | 162 | 100% |
| 17 | ". tam duoc"(2) | Trung bình | 54 | 100% |
| 18 | ". binh thuong"(2) | Trung bình | 86 | 100% |
| 19 | ". tot"(2) | Trung bình | 63 | 100% |
| 21 | ". Kém" | Kém | 75 | 100% |
| 22 | ". Thất vọng" | Kém | 122 | 96.72% |
| 23 | ". Rất tệ" | Rất kém | 42 | 97.16% |

Làm theo cách này chúng em xử lý được khoảng 7000/23000 dữ liệu còn lại dựa vào xác suất xuất hiện của các label chúng em gán cho tất cả nhãn còn lại label tốt. Để test kết quả từ file train bọn em loại bỏ label. Sau khi sử dụng mô hình này để sinh kết quả thì chúng em được kết quả là 0.6022608695652174. Kết quả không khả quan. Và một nguyên nhân chúng em không dùng mô hình này để đưa ra kết quả vì trong file test các từ bọn em xuất hiện rất ít => Không hiệu quả

## Dựa vào gom nhóm

* Cách này nhóm em định chia thành hai nhóm
* Nhóm xuất sắc + tốt : Thường được những lời khen. Chiếm 74.4858%

Trong đó tốt chiếm 62.66% .Các từ xuất hiện nhiều là "very nice", "excellent", "good","great","fantastic","amazing","helpful","help","super","delightful","well", "perfect","agréable","best","nice","beautiful","zhi","geweldige","muito bom", "nothing","fei chang bang","fang bian" ,"wonderful","nothing everything","friendly", "comfortable", "relax", "thank you","very","fabulous","spacious","comfort" ,"rat ok", "excellent", "hospitality", "parfait", "!!!","awesome","tot","dep","tan tinh","chu dao", "ngon","day du","tuyet","ly tuong","o tiep","tien nghi","vui ve","tron ven","xuat sac", " tot",". rat tot",". tuyet voi","hai long","sach se"," ok ","hop ly","yen tinh","tiep tuc", "tuyet voi","may man","lich su","thuan loi","an tuong","thu gian","hoan hao" , "tuyet hao","nhiet tinh","than thien","cam on","thoai mai","rat tuyet","se quay lai", "chuyen nghiep".

* Nhóm Trung bình + kém + rất kém : Thường bị chê. Chiếm 25.5142%. Trong đó trung bình chiếm 77.8629%.Một số từ thường xuất hiện là : ". de chiu",". rat te",". tam tam","te","tam duoc","that vong",". kem",". chap nhan duoc", ". tam tam", ". chua dat chuan",". binh thuong”,". chua dat chuan", ". tam duoc","qua toi te",". rat te",". binh thuong"
* Từ cách phân nhóm trên và các luật có sẵn thì kết quả kỳ vọng : Nhóm 1 : 75% đúng, nhóm 2 đúng 75%. Tuy nhiên, kết quả thực tế không được như kỳ vọng vì việc chia nhóm là việc làm gần như không thử. Như bọn em chia thì bọn em tìm được khoảng 16000 bộ nhóm 1 trong khi nó có 17131 mà trong có hơn 13000 bộ thực sự đúng => cách làm không hiệu quả.
* Dựa trên các mô hình có sẵn

Từ việc nhận thấy việc tự xây dựng các mô hình đem lại kết quả không cao. Chúng em đã áp dụng các mô hình như fastText, Logistic Regression , SVM để xử lý.

## Chuẩn hóa dữ liệu

Để áp dụng các mô hình một cách hiệu quả, đầu tiên chúng em chuẩn hóa dữ liệu.

B1 : Chuẩn hóa Unicode tiếng việt

Hiện nay, có 2 loại mã Unicode được sử dụng phổ biến, Unicode tổ hợp và Unicode dựng sẵn. Chúng em đưa 1 chuẩn là Unicode dựng sẵn (phổ biến hơn). Tránh trường hợp“thắng”!=”thắng”. Từ đó, giúp máy học dễ hơn,chính xác và hiệu quả hơn.

B2 : Chuẩn hóa kiểu gõ dấu

Ví dụ òa khác oà mặc dù ngữ nghĩa là hoàn toàn giống nhau. Tránh trường hợp máy tính học sai mặc dù nghĩa của các từ đó hoàn toàn như nhau.

B3 : Tách từ tiếng việt

Để cho máy tính hiểu được từ nào là từ đơn hay từ ghép . Từ đó đ việc học dữ liệu trở nên dễ dàng hơn và chính xác hơn. Ở đây chúng em dùng thư viện mã nguồn mở underthesea.

B4 : Đưa về chữ viết thường (Vì máy tính coi chữ thường và chữ hoa là khác

nhau nên để tránh trường hợp này thì chúng em đưa về chữ thường để máy tính học dễ hơn và chính xác hơn)

B5 : Xóa các ký tự không cần thiết

Ví dụ như các khoảng trắng thừa.Việc này giúp giảm số chiều đặc trưng, tăng tốc độ học và xử lý. Và tránh làm ảnh hưởng xấu đến kết quả mô hình.

## Xây dựng dữ liệu train/test

Để việc đánh giá mô hình một cách khách quan chúng em đưa ra hai trường hợp:

TH1 : File train/test = 1/1. Cách tạo , từ file train ban đầu chúng em chia đôi theo cách lấy phần dư khi chia cho 2 để đảm đảm tỷ lệ các label. Sau đó chúng em loại bỏ label 1 file để làm file test. File còn lại làm file train.

TH2 : File train/test = 2/1. Cách tạo , từ file train ban đầu chúng em chia theo cách lấy phần dư khi chia cho 3 để đảm đảm tỷ lệ các label. Sau đó, chúng em loại bỏ label 1 file để làm file test. File còn lại làm file train.

## Solution fastText

* Tổng quát về mô hình

Trong word2vec, ta không trực tiếp sử dụng thông tin hình thái học. Trong cả mô hình skip-gram và túi từ (*bag-of-word*) liên tục, ta sử dụng các vector khác nhau để biểu diễn các từ ở các dạng khác nhau. Chẳng hạn, “dog” và “dogs” được biểu diễn bởi hai vectơ khác nhau, trong khi mối quan hệ giữa hai vector đó không biểu thị trực tiếp trong mô hình. Từ quan điểm này, fastText [[Bojanowski et al., 2017]](https://d2l.aivivn.com/chapter_references/zreferences.html#bojanowski-grave-joulin-ea-2017) đề xuất phương thức embedding từ con (*subword embedding*), thông qua việc thực hiện đưa thông tin hình thái học vào trong mô hình skip-gram trong word2vec.

* Nêu kết quả đánh giá

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train | Train | Train | Train | Test | 0.793043478 |
| Train | Train | Train | Test | Train | 0.785652174 |
| Train | Train | Test | Train | Train | 0.792173913 |
| Train | Test | Train | Train | Train | 0.797608695 |
| Test | Train | Train | Train | Train | 0.787779952 |

## Solution Logistic Regression

* Tổng quát về mô hình

Logistic Regression là 1 thuật toán phân loại được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc (như 0, 1, 2, ...). Một ví dụ điển hình là phân loại Email, gồm có email công việc, email gia đình, email spam, ... Giao dịch trực tuyến có là an toàn hay không an toàn, khối u lành tính hay ác tình. Thuật toán trên dùng hàm sigmoid logistic để đưa ra đánh giá theo xác suất. Ví dụ: Khối u này 80% là lành tính, giao dịch này 90% là gian lận, ...

* Nêu kết quả đánh giá

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train | Train | Train | Train | Test | 0.645652173 |
| Train | Train | Train | Test | Train | 0.657608695 |
| Train | Train | Test | Train | Train | 0.661304347 |
| Train | Test | Train | Train | Train | 0.668478260 |
| Test | Train | Train | Train | Train | 0.665796912 |

## Solution SVM

* Tổng quát về mô hình

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồi thị dữ liệu là các điểm trong n chiều ( ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (*hyper-plane*) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

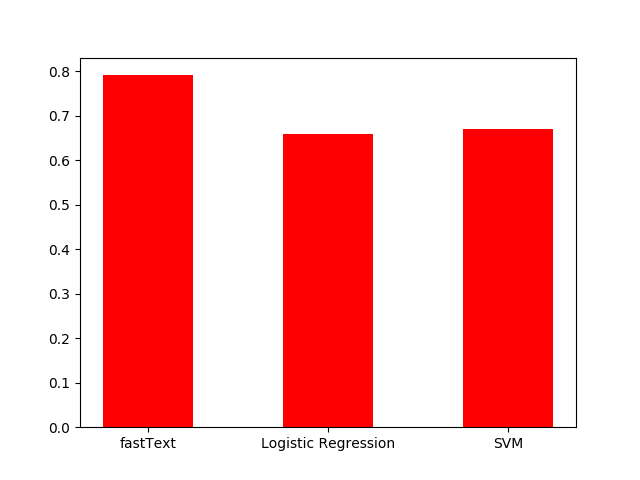
*Support Vectors* hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, *Support Vector Machine* là một biên giới để chia hai lớp tốt nhất.

* Nêu kết quả đánh giá

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train | Train | Train | Train | Test | 0.670869565 |
| Train | Train | Train | Test | Train | 0.662826086 |
| Train | Train | Test | Train | Train | 0.668043478 |
| Train | Test | Train | Train | Train | 0.668043478 |
| Test | Train | Train | Train | Train | 0.683191998 |

# Đánh giá các mô hình

## So sánh kết quả của các mô hình



## Nhận xét và đánh giá

* Mô hình fastText cho kết quả khá khả quan gần 80%(0.7912516425)
* Còn hai mô hình Logistic Regression và SVM cho kết quả kém hơn và gần như tương đương nhau lần lượt 0.6597680774 và 0.6705965174.
* Các kết quả này còn hơi thấp so với kỳ vọng do hai nguyên nhân sau:
* Dữ liệu đầu vào train không đủ nhiều nên việc học của các mô hình trên chưa đạt hiệu quả cao nhất.
* Việc tiền xử lý dữ liệu chưa được tốt nhất có thể.

# Tài liệu tham khảo

<https://fasttext.cc/>

<https://scikit-learn.org/stable/index.html>

<https://nguyenvanhieu.vn/phan-loai-van-ban-tieng-viet/>

<https://github.com/nvlong198/Sell-Detection?fbclid=IwAR1mgg5QySHbXAmSng8T70MnJOh9Idq4a-DaFNIFhYpaINFYOpIooW34iAU>