|  |
| --- |
| **Trường Đại học Công Nghệ Thông Tin – Đồ án Cuối Kì** |
|  |
| Phân loại  văn bản |
| **Sử dụng thuật toán máy học** |

*Giảng viên hướng dẫn: TS Nguyễn Vinh Tiệp*

*Nhóm sinh viên thực hiện:*

* *Phan Thanh Nghĩa – 18521148*
* *Phan Lê Phú – 18521247*

**Mục lục**

1. [Giới thiệu 3](#_TOC_250024)
2. [Dataset 4](#_TOC_250023)
3. [Phương pháp](#_TOC_250022) 4
   1. [Khởi tạo dữ liệu ban đầu](#_TOC_250021) 4
   2. [Exploratory Data Analysis](#_TOC_250020) 5
   3. [Tiền xử lí dữ liệu](#_TOC_250019) 6
      1. [Chuẩn hóa Unicode tiếng Việt](#_TOC_250019) 6
      2. [Chuẩn hóa kiểu gõ dấu](#_TOC_250017) 6
      3. [Tách từ tiếng Việt](#_TOC_250016) 6
      4. [Xóa các stopword tiếng Việt](#_TOC_250015) 6
   4. [Feature engineering](#_TOC_250014) 7
      1. [Biểu diễn tập văn bản](#_TOC_250013) 7
      2. Mã hóa nhãn 8
      3. [Phân chia tập train-test](#_TOC_250011) 9
   5. [Mô hình dự đoán](#_TOC_250006) 9
      1. [Machine learning model](#_TOC_250019) 9
      2. [Best Selection Model](#_TOC_250019) 9
4. [Kết quả](#_TOC_250003) 10

# Giới thiệu

Phân loại văn bản (Text classification) là một bài toán phổ biến trong xử lý ngôn ngữ tự nhiên (Nature language processing). Đối với phân loại văn bản tiếng Việt sẽ có những khác biệt so với phân loại văn bản tiếng Anh.

Phân loại văn bản (Text Classification) 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.

Diagram

Description automatically generated

# Dataset

Dataset sử dụng trong project này nhóm em tự crawl trên:

https://vnexpress.net/

Dataset lúc đầu bao gồm 2747 articles, thuộc 6 loại sau:

* Đời sống
* Giải trí
* Giáo dục
* Kinh doanh
* Pháp luật
* Sức khỏe

# Phương pháp

Tại mục này nhóm chúng em đề cập đến: khởi tạo dataset ban đầu, phân tích dataset, tiền xử lí dữ liệu, feature engineering, training models

## Khởi tạo tập dữ liệu ban đầu

Dataset của nhóm em được tạo theo cấu trúc:

|  |  |  |
| --- | --- | --- |
|  | **Content** | **Category** |
| Document 1 STT | Document 1 Content | Document 1 Category |
| … | … | … |

Mỗi hàng sẽ đại diện cho một document riêng biệt và mỗi cột sẽ lưu trữ STT, nội dung và thể loại.

## Phân tích dữ liệu

Đối với việc phát triển một mô hình phân loại thì các lớp dữ liệu có cân bằng hay không là một điều đáng quan tâm.

Ví dụ, nếu chúng ta có hai lớp và 95% thuộc về một trong số chúng, thì một mô hình phân loại dù có tệ đến đâu thì cũng luôn xuất ra lớp đa số sẽ có độ chính xác 95%, cho dù nó sẽ thất bại tất cả các dự đoán của lớp thiểu số.

Chart, bar chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

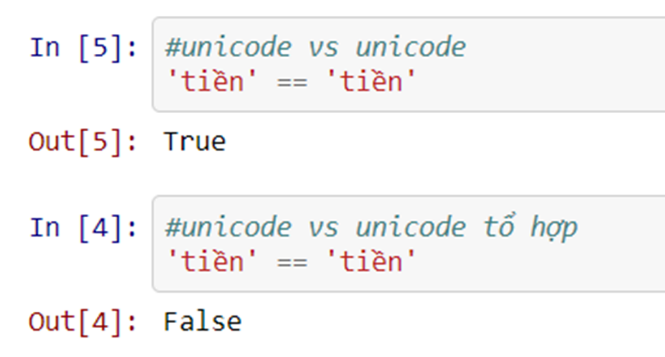
Ở đây thì dataset của chúng ta khá cân bằng về số lượng lẫn độ dài.

## Tiền xử lí văn bản

Việc tiền sử lý dữ liệu là quá trình **chuẩn hóa dữ liệu**và **loại bỏ các thành phần không có ý nghĩa** cho việc phân loại văn bản.

### Chuẩn hóa Unicode tiếng Việt

Hiện nay, có 2 loại mã Unicode được sử dụng phổ biến, Unicode tổ hợp và Unicode dựng sẵn. Hướng xử lý: Đưa về 1 chuẩn Unicode dựng sẵn.



### Chuẩn hóa kiểu gõ dấu

Kiểu gõ dấu khác nhau thì bạn nhìn mắt thường cũng sẽ thấy được sự khác nhau: òa với oà lần lượt là kiểu gõ cũ (phổ biến hơn) và kiểu gõ mới. Ở đây chúng ta sẽ chuyển về kiểu gõ dấu cũ.

### Tách từ tiếng Việt

Trong tiếng Việt của chúng ta, đơn vị từ bao gồm từ đơn và từ ghép. Nên chúng ta cần phải nói cho mô hình máy học đâu là từ đơn, đâu là từ ghép. Nếu không thì từ nào cũng là từ đơn hết.

Bởi vì mô hình của chúng ta sẽ coi các từ là đặc trưng, tách nhau theo dấu cách. Do đó, chúng ta phải nối các từ ghép lại thành một từ để không bị tách sai:



### Xóa các stop word tiếng Việt

Stopword là các từ xuất hiện nhiều ở tất cả các chuyên mục cần phân loại. Do đó, chúng là các đặc trưng không có tác dùng cho việc phân loại văn bản.

Các stopword thường là các từ nối (của, là, có, được, những,…) và các từ đặc trưng của

dữ liệu (ví dụ như các từ “máy bay, tiếp viên” là các stopword nếu làm bài phân loại đánh giá khách hàng của doanh nghiệp vận tải hàng không.

Để xây dựng bộ stopword thì chúng ta chỉ cần thông kê các từ xuất hiện nhiều trong tất cả các mục và lấy top đầu là được.

Với tập dữ liệu của em có khoảng gần 2500 documents, với số lượng từ nhiều thì nhóm em chọn 100 từ đầu trong kết quả thu được.

## Feature Engineering

Feature engineering là một phần thiết yếu trong xây dựng bất kì một hệ thống thông minh nào. Andrew Ng đã từng nói:

*“Coming up with features is difficult, time-consuming, requires expert knowledge.*

*Applied machine learning‟ is basically feature engineering.”*

Feature engineering là một quá trình chuyển đổi dữ liệu trở thành features để sử dụng như input cho các mô hình máy học sao cho cải thiện được hiệu suất mô hình.

Khi xử lí dữ liệu văn bản, có nhiều cách để lấy features đại diện cho data. Nhóm em sẽ đề cập đến một số phương pháp phổ biến nhất và sau đó sẽ chọn phương pháp phù hợp nhất cho project của nhóm.

### Biểu diễn văn bản

Có nhiều phương pháp để biểu diễn văn bản như Word Count Vectors, TF-IDF Vectors, Word Embeddings, Text based or NLP based features. Ở đây chúng em chọn TF-IDF Vectors bởi vì mô hình này đơn giản nhưng kết quả mang lại rất tốt, cũng như quá trình thực hiện không quá phức tạp.

**TF–IDF Vectors**

Là một phương pháp thể hiện tầm quan trọng của một thuật ngữ trong một tài liệu và toàn bộ ngữ liệu. TF là viết tắt của *Term Frequency* (Tần suất xuất hiện của thuật ngữ trong tài liệu) và IDF là viết tắt của *Inverse Document Frequency* (nghịch đảo của tần suất của tài liệu chứa thuật ngữ trong toàn bộ ngữ liệu)

Text

Description automatically generated

Trong đó:

* 𝑡: term
* 𝑑: document
* 𝑇𝐹(𝑡): term frequency (tần suất term, vd như từ “Cây” xuất hiện bao nhiêu lần trong một tài liệu)
* 𝑁: số lượng tài liệu trong ngữ liệu
* 𝐷𝐹 𝑡 : số lượng tài liệu trong ngữ liệu chứa thuật ngữ t

Nguồn: [*https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-*](https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/)[*understand-and-implement-text-classification-in-python/*](https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/)

Giá trị TF-IDF tăng tỉ lệ thuận với số lần một từ xuất hiện trong tài liệu và được bù lại bởi số tài liệu trong kho ngữ liệu chứa từ này, điều này giúp điều chỉnh thực tế là có một số từ có thể không quan trọng nhưng lại xuất hiện quá nhiều lần.

Nó cũng tính đến thực tế là có thể một số document có thể lớn hơn nhiều so với một số khác, điều này được khắc phục bằng chuẩn hóa TF term.

### Mã hóa nhãn

Machine learning models yêu cầu features và nhãn là số học để dự đoán. Vì vậy nên chúng ta phải tạo một từ điển để ánh xạ mỗi nhãn cho một ID số. Lược đồ ánh xạ như sau:

|  |  |
| --- | --- |
| **Category Name** | **Category Code** |
| Đời sống | 0 |
| Giải trí | 1 |
| Giáo dục | 2 |
| Kinh doanh | 3 |
| Pháp luật | 4 |
| Sức khỏe | 5 |

### Phân chia tập train-test

Chúng ta cần phải chia một test set để có thể chứng minh chất lượng mô hình khi chúng phải dự đoán data chưa từng gặp qua. Chúng em chọn tỉ lệ chia là 80% với tập train và 20% với tập test.

## Mô hình dự đoán

### Machine learning model

Nhóm em đã thử nghiệm một số mô hình máy học để tìm ra mô hình phù hợp nhất với bộ dữ liệu, như:

* Linear Support Vector Classifier
* Gaussian Naive Bayes
* LogisticRegression
* Random Forest Classifier

Nhóm em chỉ sử dụng các mô hình máy học cổ điển thay vì thay vì các mô hình học sâu vì lượng dữ liệu nhóm em có không đủ, điều này có thể dẫn đến các mô hình bị overfit.

**Source\_code:** https://github.com/nghiaphan177/Truyvanttdpt/blob/main/Do\_An\_Truy\_Van\_Thong\_tin\_da\_Phuong\_Tien.ipynb

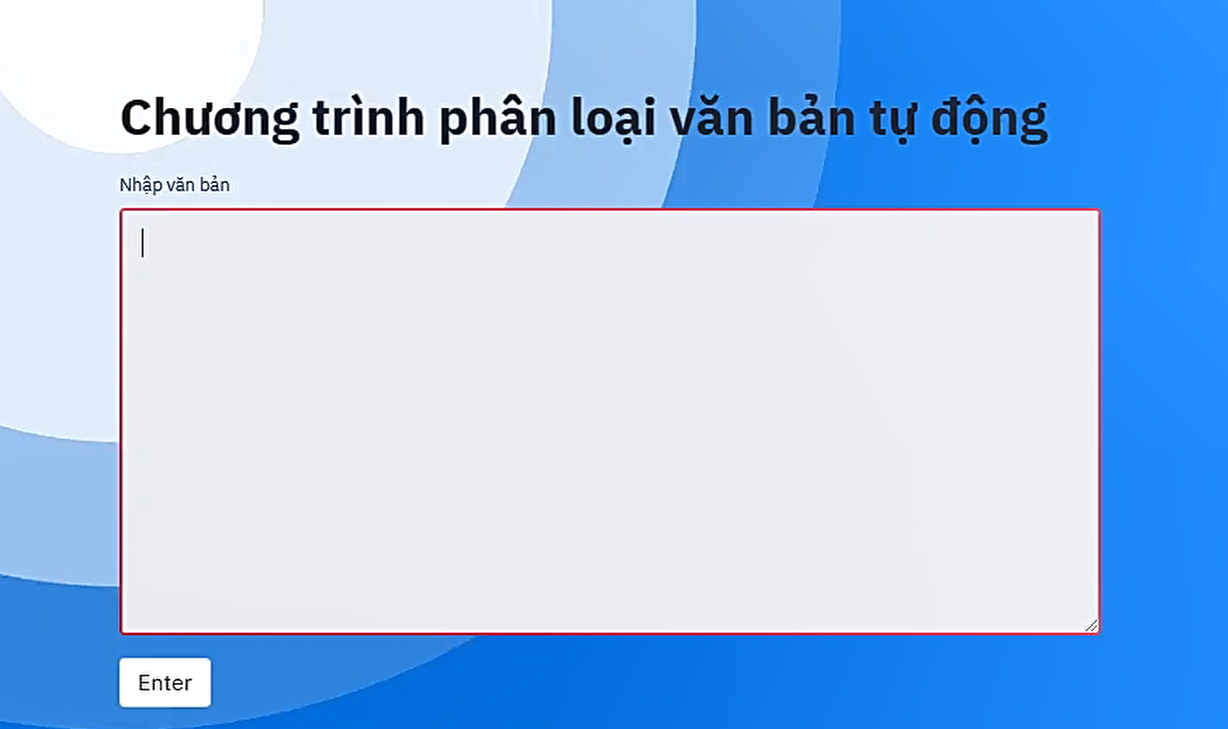
### Best selection model

|  |  |  |
| --- | --- | --- |
| **Model** | **Train accuracy** | **Test accuracy** |
| Linear Support Vector Classifier | 0.9986 | 0.9436 |
| Gaussian Naive Bayes | 0.9845 | 0.8218 |
| LogisticRegression | 0.9767 | 0.9509 |
| Random Forest Classifier | 0.9509 | 0.8581 |

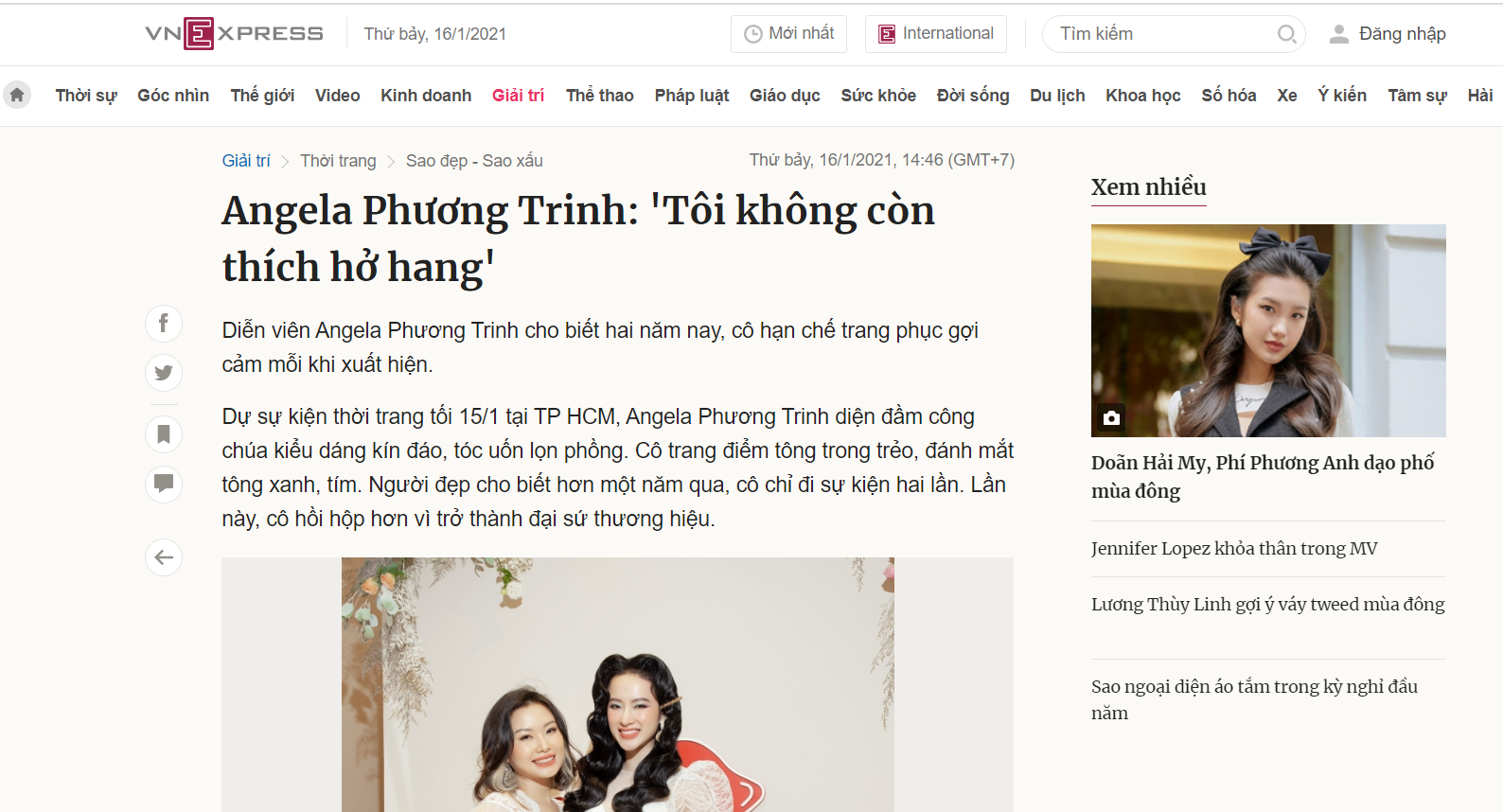
Random Forest Classifier và Gaussian Naïve Bayes có vẻ như bị overfit nên nhóm em chọn mô hình LogisticRegession là mô hình tốt nhất còn lại.

# 4. Kết quả

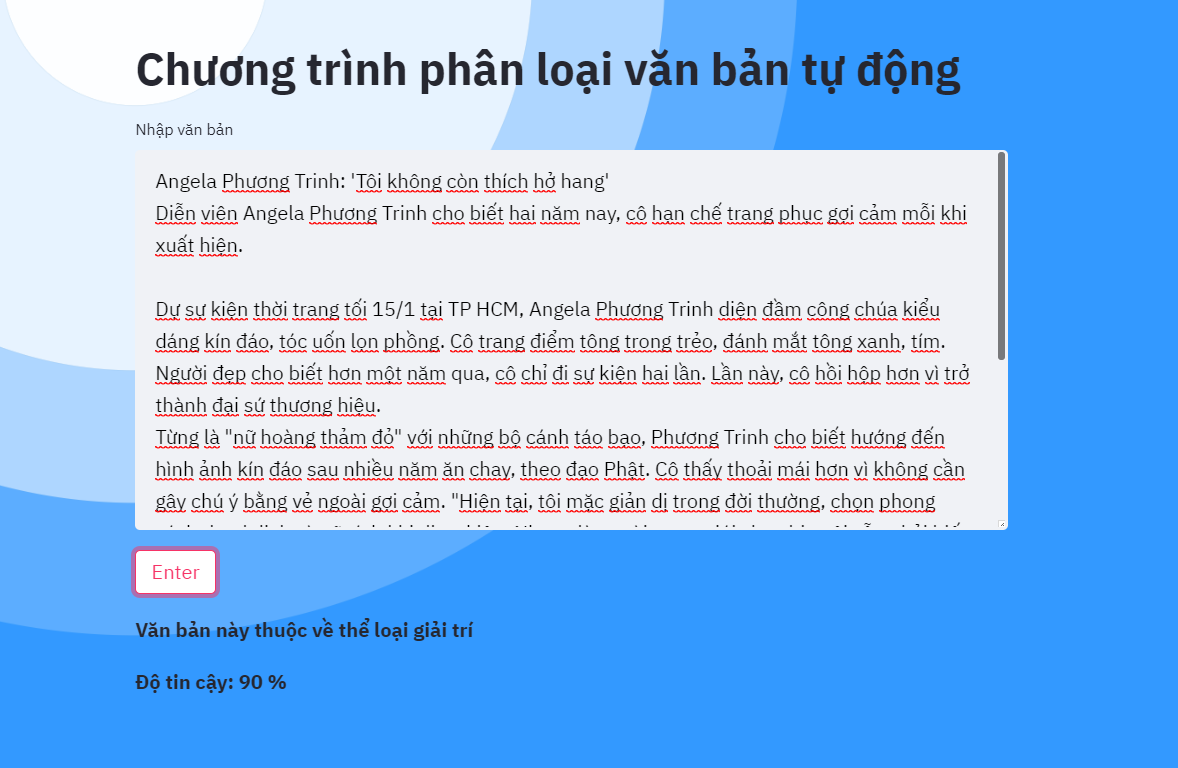
Nhóm em tiến hành xây dựng một webapp như hình để tiến hành biểu diễn kết quả



**Tiến hành chọn một bài báo thuộc lĩnh vực GIẢI TRÍ**



**Kết quả:**



Nếu độ tin cậy của bất kì thể loại nào dưới 50% thì kết quả sẽ hiển thị đó là một loại khác trong 6 thể loại đã xét:

Ví dụ như khi cho một đoạn văn bản thuộc thể loại XE:

