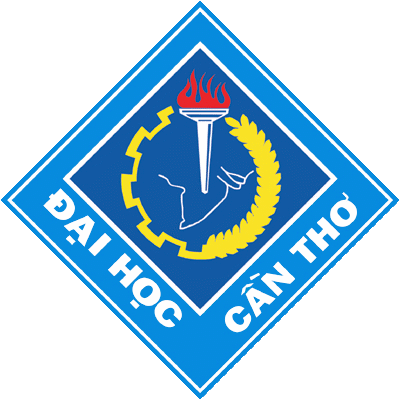
TRƯỜNG ĐẠI HỌC CẦN THƠ

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



**Cần Thơ, 05/2022**

**Sinh viên thực hiện:**

**Vương Cẩm Thanh B1805916**

**Cán bộ hướng dẫn:**

**TS. Lưu Tiến Đạo**

**Đề tài**

KHAI KHOÁNG DỮ LIỆU

ỨNG DỤNG

PHÂN LOẠI BÀI BÁO

MỤC LỤC

Nội dung

[A. Phân lớp văn bản 3](#_Toc104212557)

[1. Tổng quan 3](#_Toc104212558)

[2. Ứng dụng 3](#_Toc104212559)

[3. Chuẩn bị dữ liệu 3](#_Toc104212560)

[B. Tiền xử lý dữ liệu 5](#_Toc104212561)

[1. Làm sạch dữ liệu 6](#_Toc104212562)

[2. Tách từ 7](#_Toc104212563)

[3. Phân tích dữ liệu 8](#_Toc104212564)

[C. Vector hóa văn bản 11](#_Toc104212565)

[1. Tổng quan 11](#_Toc104212566)

[2. Vector hóa văn bản với stop words sau khi phân tích: 11](#_Toc104212567)

[D. Train model 12](#_Toc104212568)

[1. Train model theo 3 thuật toán K-nearest neighbors, Naive Bayes và SVM 12](#_Toc104212569)

[2. Tạo hàm vẽ confusion matrix và tính F1 13](#_Toc104212570)

[E. Deploy Model 15](#_Toc104212571)

[1. Sử dụng Dockerfile 15](#_Toc104212572)

[2. Endpoints để dự đoán 16](#_Toc104212573)

[3. Thử api đã deploy lên heroku bằng Postman 17](#_Toc104212574)

[4. Giao diện cho người dùng 17](#_Toc104212575)

# Phân lớp văn bản

## Tổng quan

Phân lớp văn bản được coi là quá trình phân loại một văn bản bất kì vào một hay nhiều lớp cho trước. Quá trình này gồm hai bước. Ở bước thứ nhất, một mô hình phân lớp (classfication model) được xây dựng dựa trên tri thức kinh nghiệm. Ở đây, tri thức kinh nghiệm chính là một tập dữ liệu huấn luyện (training dataset) được cung cấp bởi con người bao gồm một tập văn bản và phân lớp tương ứng của chúng. Bước này còn gọi là bước xây dựng huấn luyện (training process) hay ước lượng mô hình phân lớp. Ở bước thứ hai, mô hình phân lớp xây dựng ở bước đầu sẽ được sử dụng để phân lớp cho những văn bản (chưa được phân loại) trong tương lai. Bước đầu tiên được xem như là việc học có giám sát mà chúng ta có thể sử dụng rất nhiều các kỹ thuật học máy đã có như: Naïve Bayes, k láng giềng gần nhất (kNN), cây quyết định (Decision Tree), Support Vector Machine (SVM).Mục tiêu của bài toán phân lớp là nhằm xây dựng mô hình có khả năng gán nhãn cho một bài báo bất kì với độ chính xác cao nhất có thể.

## Ứng dụng

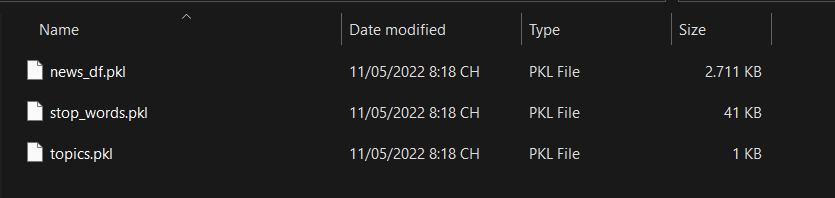
Ứng dụng lớn nhất của bài toán phân lớp văn bản là áp dụng vào bài toán phân loại hay lọc nội dung. Trong bài toán lọc nội dung: một văn bản được phân loại vào nhóm: có ích hoặc không có ích, các ứng dụng cụ thể như: lọc email rác, … . Trong đề tài này sẽ là phân loại bài báo theo 10 chủ đề khác nhau. Với tất cả ý nghĩa thực tế trên, một lần nữa có thể khẳng định rằng trong thời đại Internet được coi là một phần không thể thiếu trong cuộc sống, phân lớp văn bản luôn là vấn đề đáng được quan tâm để có thể phát triển và xây dựng được những công cụ ngày càng hữu dụng hơn. Dựa trên nhu cầu cấp thiết đó, em chọn đề tài "Phân loại bài báo" để có thể nghiên cứu và phát triển ứng dụng này

## Chuẩn bị dữ liệu

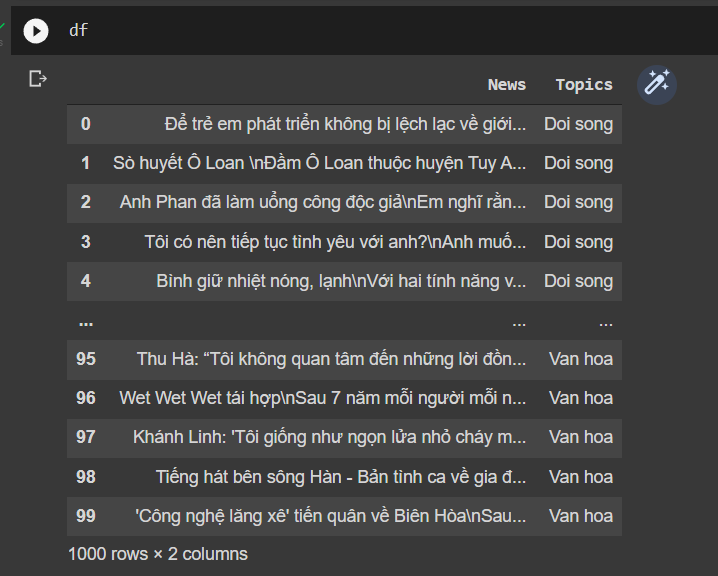
Dữ liệu là yếu tố quan trọng nhất và cũng là vấn đề mà chúng ta cần quan tâm nhất. Trong quá trình xây dựng một hệ thống phân lớp văn bản, bước chuẩn bị và tiền xử lý dữ liệu quyết định tới thành bại của hệ thống hơn cả. Tất cả dữ liệu đã có sẵn tại đây: <https://github.com/ThanhB1805916/DataMining_CT312>

Các file này nằm trong thư mục: project/google\_colab/

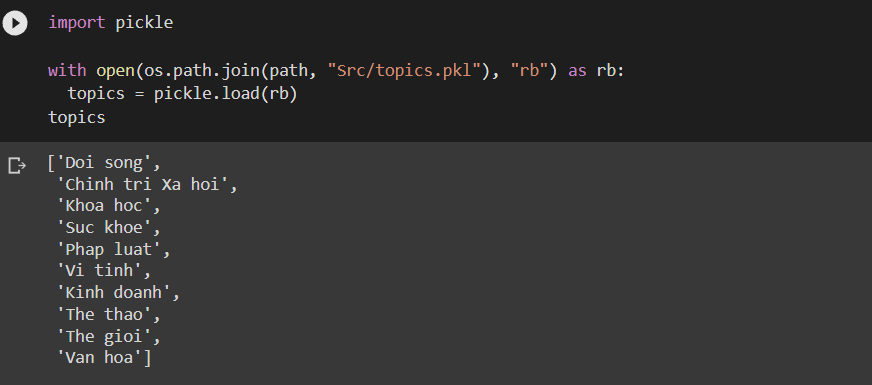
Chạy các file này trên google colab và có yêu cầu quyền truy cập drive.

Dữ liệu có sẵn là các file pickle dưới đây nằm trong thư mục Src:

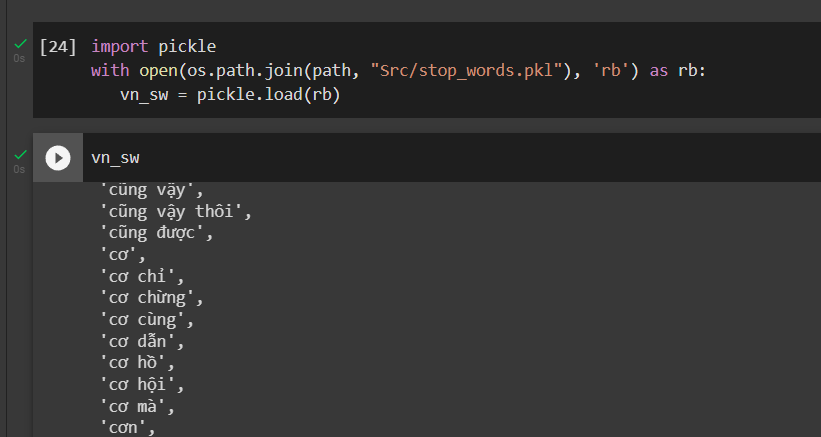
1. news\_df.pkl: là data frame chứa các bài báo vào topics



1. topics.pkl: là list 10 topics



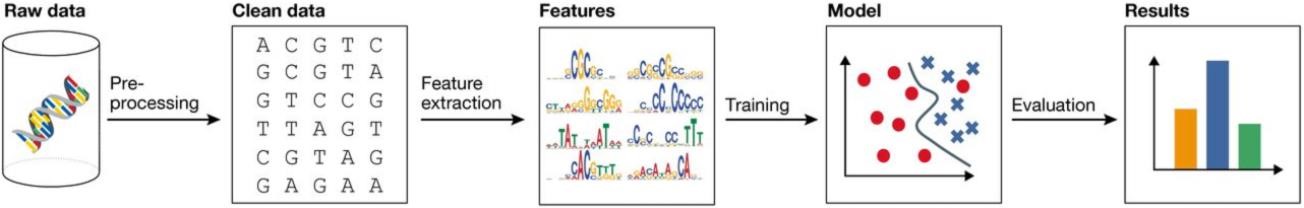
1. stop\_words.pkl là danh sách các từ dừng



# Tiền xử lý dữ liệu

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp[[1]](#footnote-1).

Bước đầu tiên trong xử lý ngôn ngữ tự nhiên là tiền xử lý dữ liệu. Vì văn bản trong ngôn ngữ tự nhiên không có cấu trúc, làm cho chương trình không thể hiểu được đâu là dữ liệu có giá trị. Quy trình tiền xử lý dữ liệu để có dữ liệu phù hợp được tổng kết như hình sau:

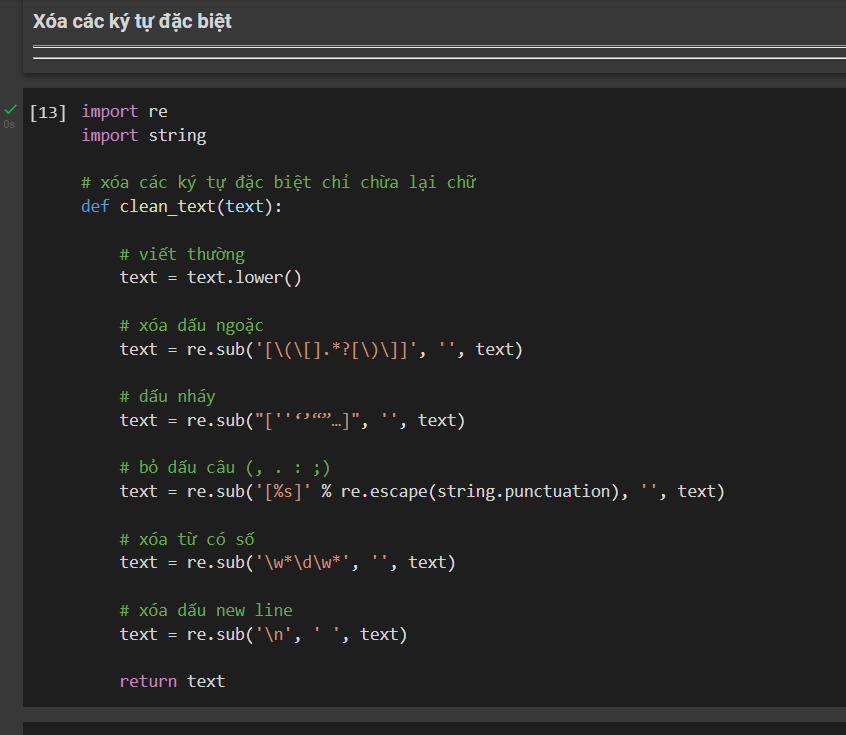


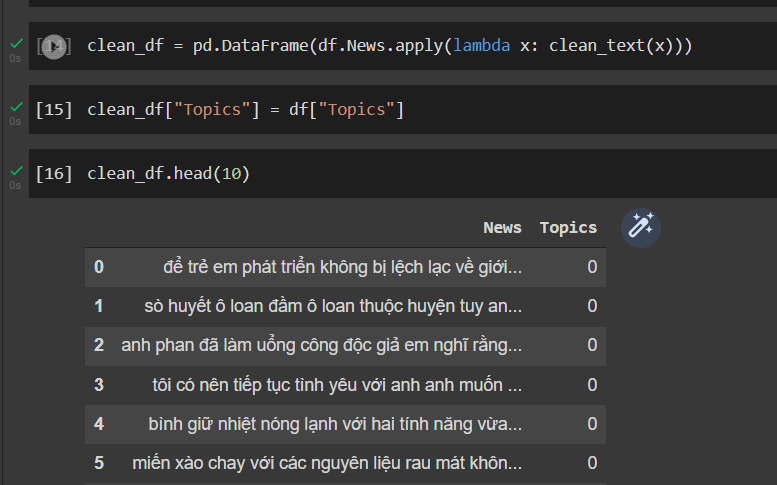
Các bước tiền xử lý được sử dụng cho máy học thường gồm các bước: làm sạch dữ liệu, tách từ, chuẩn hóa từ, loại bỏ stopword… và được thể hiện như hình sau:



## Làm sạch dữ liệu

Các chuỗi văn bản sẽ được chuyển sang viết thường, bỏ các ký tự đặc biệt, các dấu câu và số.



Dữ liệu News sau khi được làm sạch:

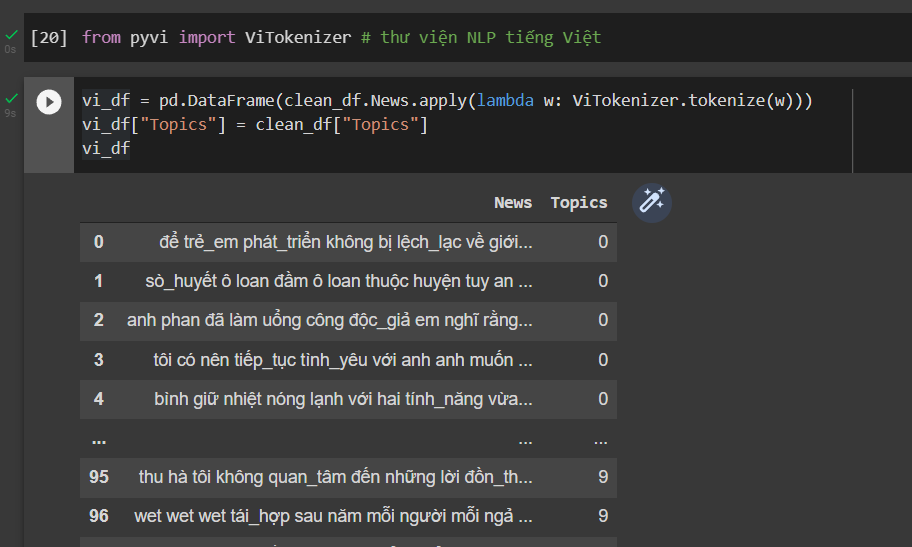
## Tách từ

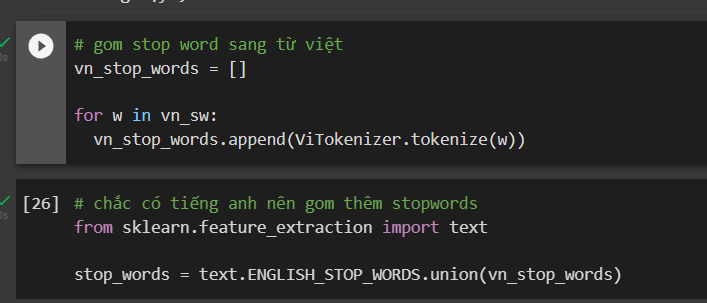
Đơn vị từ trong tiếng Việt bao gồm từ đơn và từ ghép. Nên chúng ta cần phải nói cho mô hình học máy biết đâu là từ đơn, đâu là từ ghép. Nếu không thì từ nào cũng sẽ là từ đơn hết.

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.

Sử dụng thư viện pyvi để tìm các từ tiếng việt gom các từ ghép thành dạng trẻ em -> trẻ\_em để tạo 1 từ gọi là token, vì tiếng anh các từ cách nhau bởi khoảng trắng.

* 1. Đầu tiên sẽ token hóa các News:



* 1. Sau đó sẽ token các stop\_words tiếng việt và thêm stop\_words của tiếng anh:

## Phân tích dữ liệu

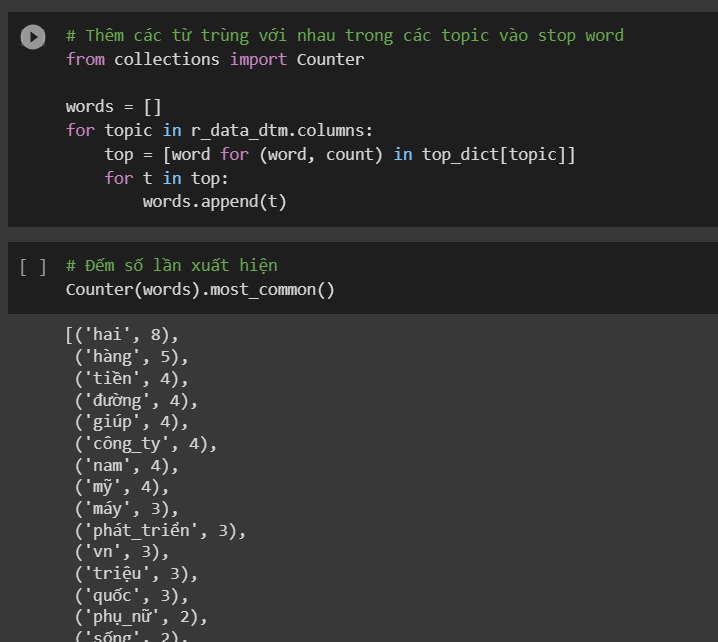
Load các dữ liệu bên clean

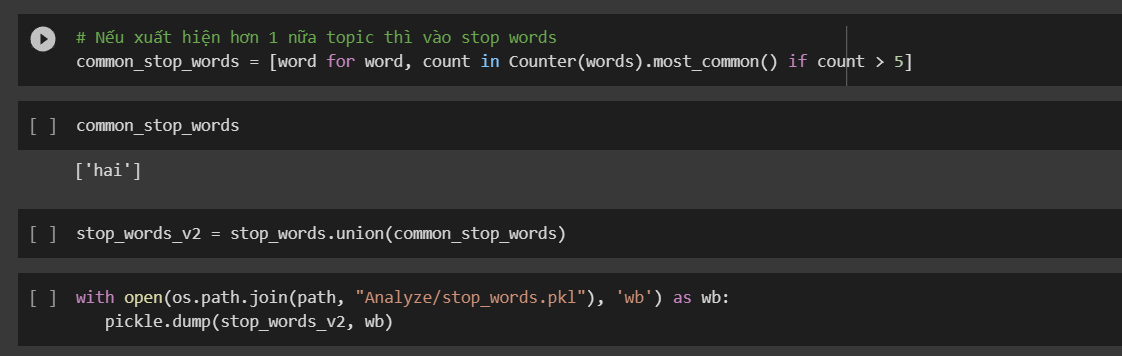
1. Dùng word cloud để phân tích
2. Top các từ xuất hiện nhiều trong các tiêu đề



1. Tìm từ xuất hiện nhiều giữa các topics

Vì các từ này đều có trong các topics nên ta sẽ bỏ nó vào stop\_words



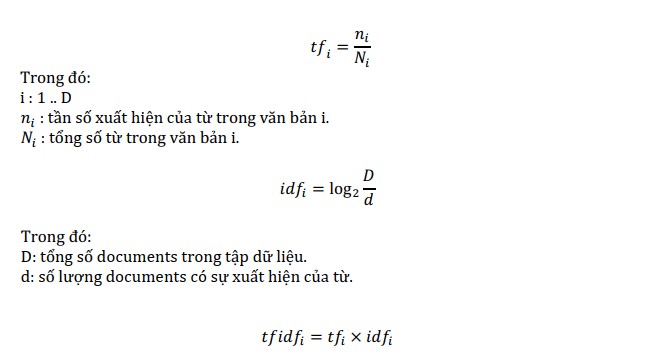
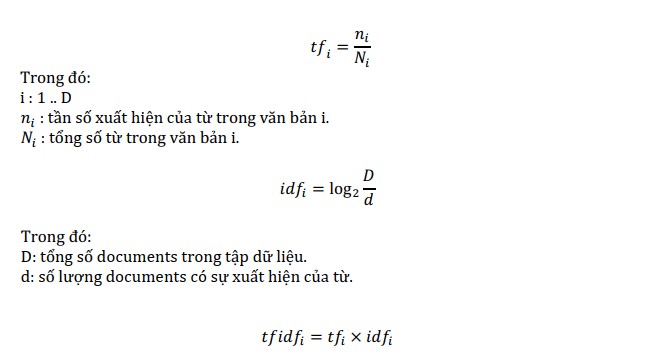
1. Thêm các từ xuất hiện trên 5 topics vào stop\_words vào lưu vào Analyze

# Vector hóa văn bản

## Tổng quan

Thông thường, máy tính không thể hiểu được ý nghĩa các từ. Như vậy, để xử lý được ngôn ngữ tự nhiên, ta cần có một phương pháp để biểu diễn văn bản dưới dạng mà máy tính có thể hiểu được. Phương pháp tiêu chuẩn để biểu diễn văn bản đó là biểu diễn các văn bản theo vector. Trong đó, các từ/cụm từ thuộc kho tài liệu ngôn ngữ được ánh xạ thành những vector trên hệ không gian số thực. Ở đây em sẽ sử dụng thuật toán TF-IDF

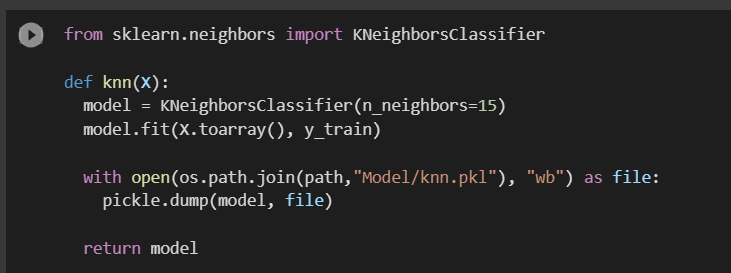
TF- IDF (term frequency–inverse document frequency) – tần suất- tần suất đảo nghịch từ. Đây là một phương pháp thống kê, nhằm phản ánh độ quan trọng của mỗi từ hoặc n-gram đối với văn bản trên toàn bộ tài liệu đầu vào. TF-IDF thể hiện trọng số của mỗi từ theo ngữ cảnh văn bản. TF-IDF sẽ có giá trị tăng tỷ lệ thuận với số lần xuất hiện của từ trong văn bản và số văn bản có chứa từ đó trên toàn bộ tập tài liệu. Phương pháp này giúp cho TF-IDF có tính phân loại cao hơn.

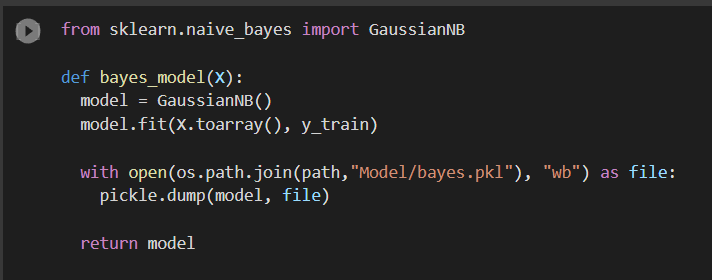


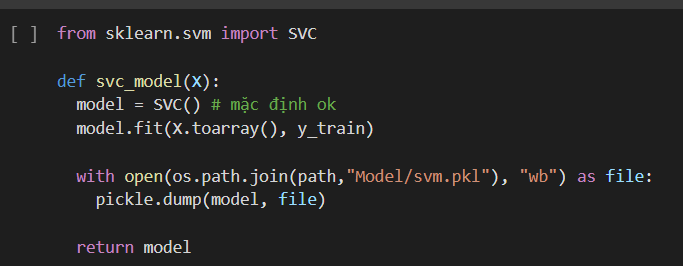
## Vector hóa văn bản với stop words sau khi phân tích:

# Train model

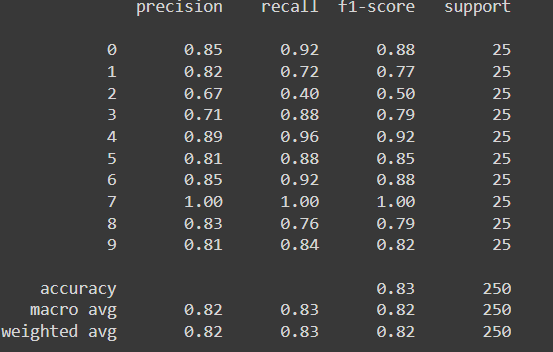
## Train model theo 3 thuật toán K-nearest neighbors, Naive Bayes và SVM

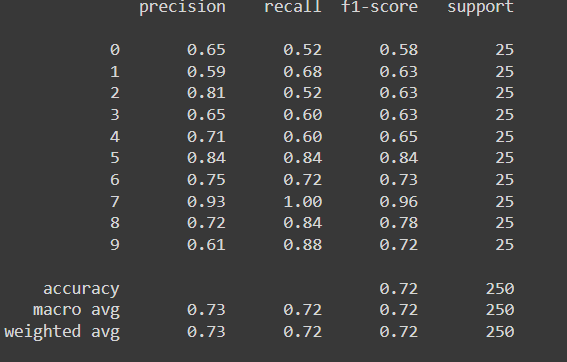


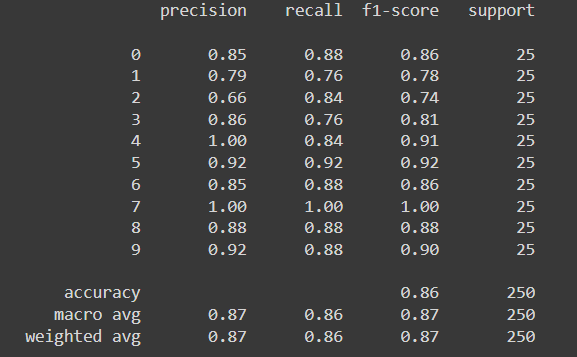




## Tạo hàm vẽ confusion matrix và tính F1

* 1.  K-nearest neighbors
  2. Naive Bayes



* 1. SVM

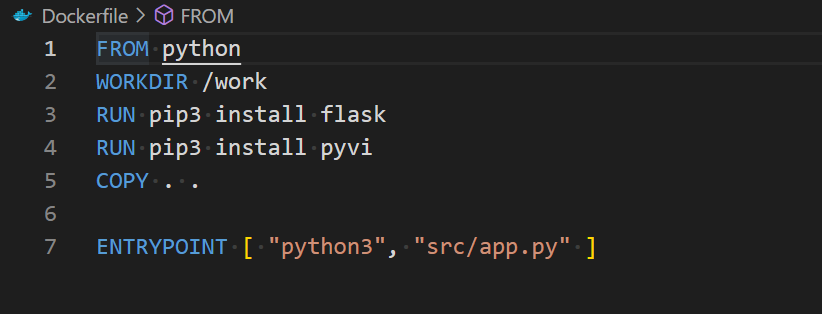
1. Bảng so sánh chỉ số F1

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Naive Bayes | K-nearest neighbors | SVM |
| F1 | 0.72 | 0.83 | 0.86 |

Vì SVM có F1 cao nhất nên sẽ sử dụng model này để deploy

# Deploy Model

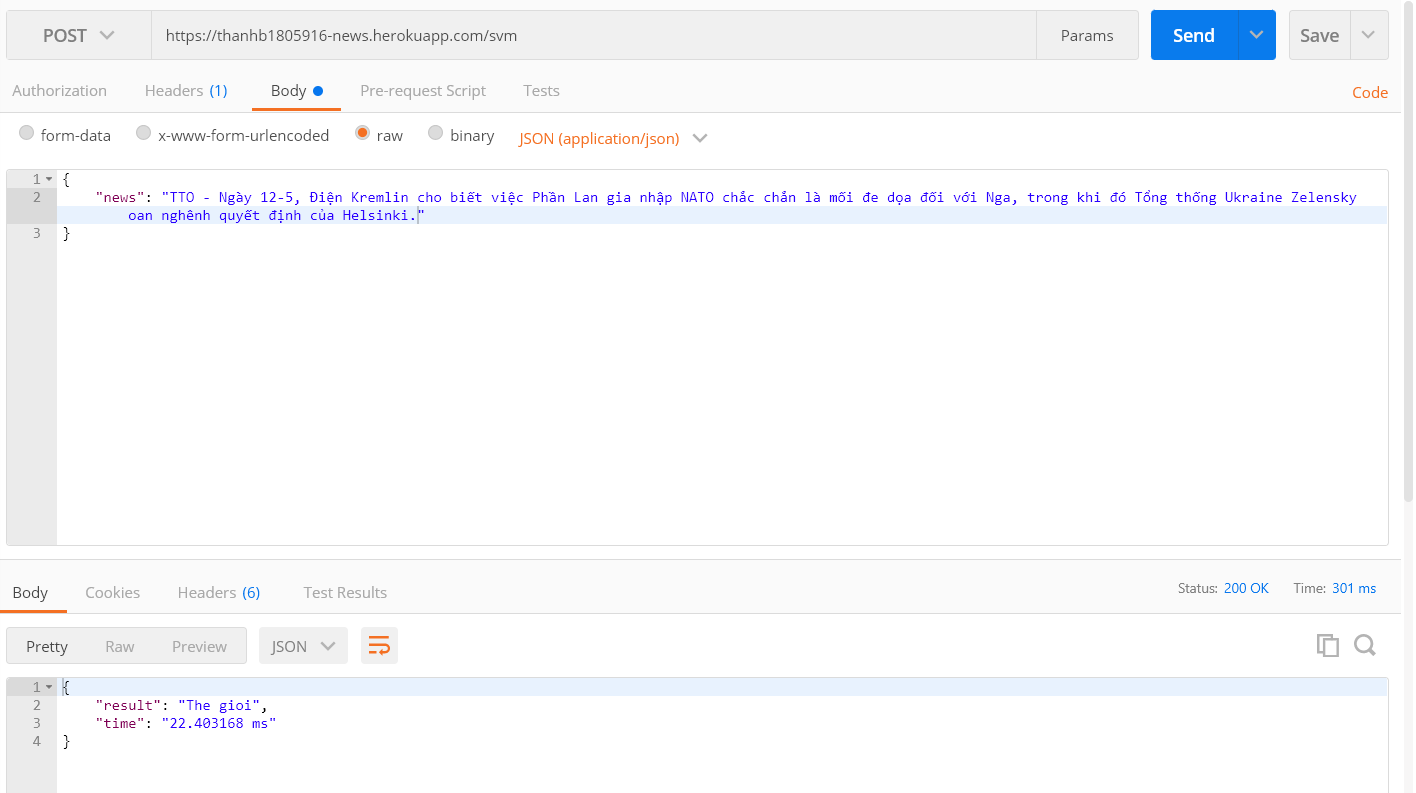
## Sử dụng Dockerfile



Dùng flask để tạo web service và pyvi để token hóa các dữ liệu đầu vào

## Endpoints để dự đoán

## Thử api đã deploy lên heroku bằng Postman



## Giao diện cho người dùng

1. [↑](#footnote-ref-1)