Báo Cáo Đồ Án

Khai Khoáng Dữ Liệu

Đề tài 3: Xây dựng ứng dụng phân loại văn bản của 10 lĩnh vực

# Train Model

## Tổng quan

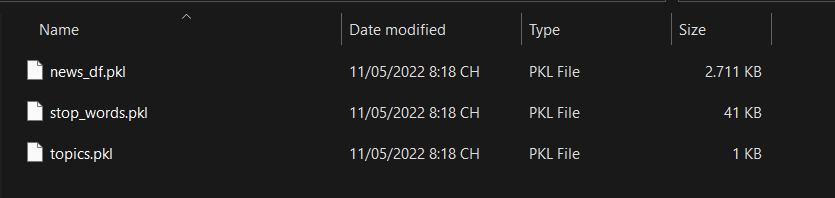
Cho 10 lĩnh vực của các bài báo, hãy viết 1 web service để phân loại văn bản đầu vào. Tạo thêm một website để tương tác với web service.

## Chuẩn bị dữ liệu

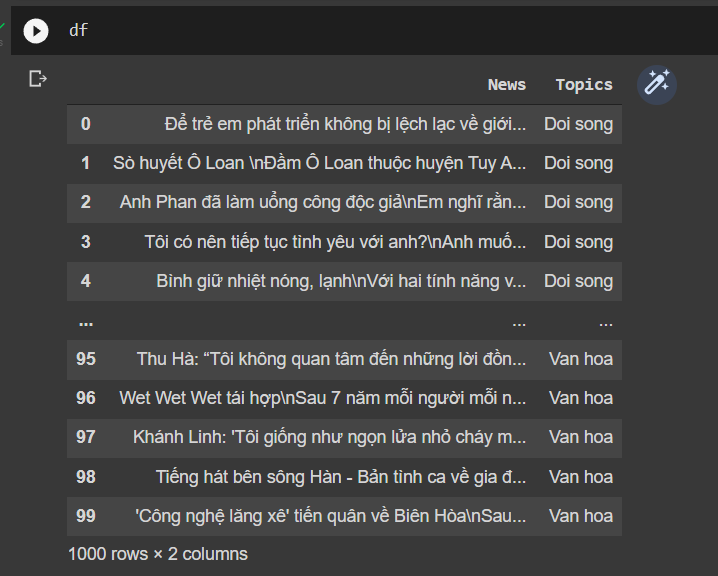
Link github: <https://github.com/ThanhB1805916/DataMining_CT312>

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

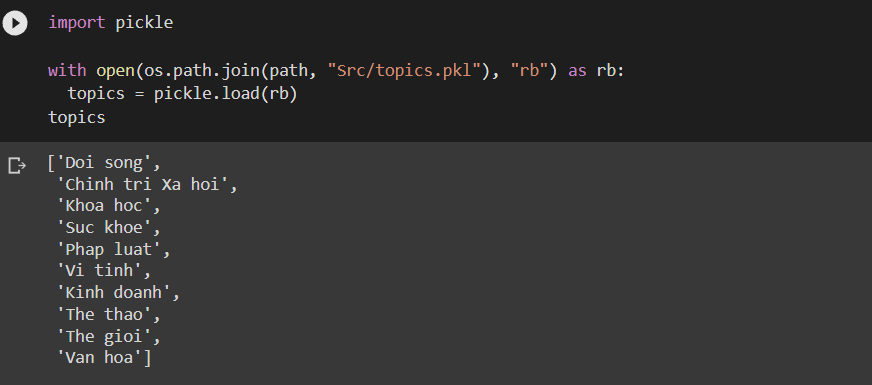
Chạy các file này trên google colab và có yêu cầu quyền truy cập drive.

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

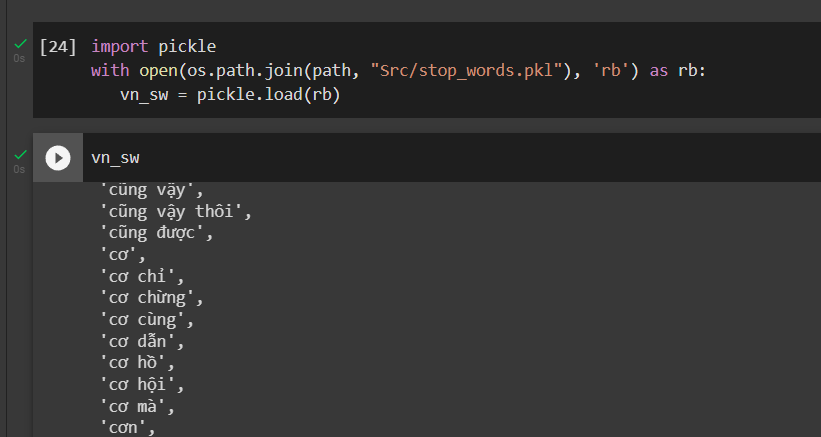
1. news\_df.pkl: là data frame chứa các bài báo vào topics



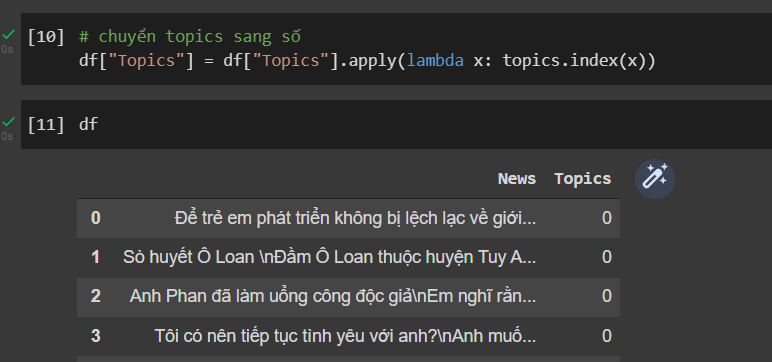
1. topics.pkl: là list 10 topics

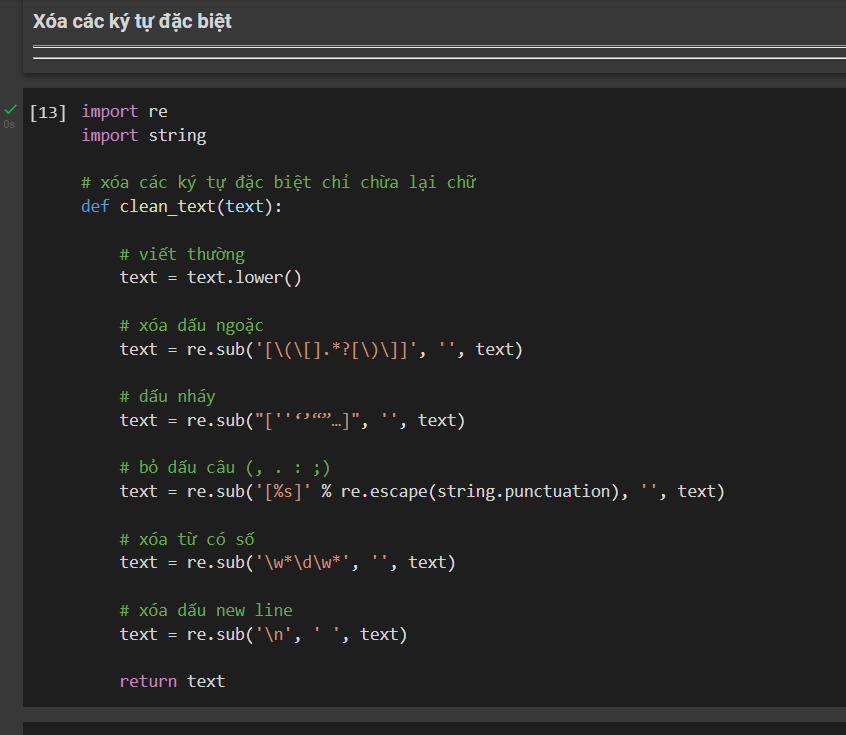


1. stop\_words.pkl là danh sách các từ dừng

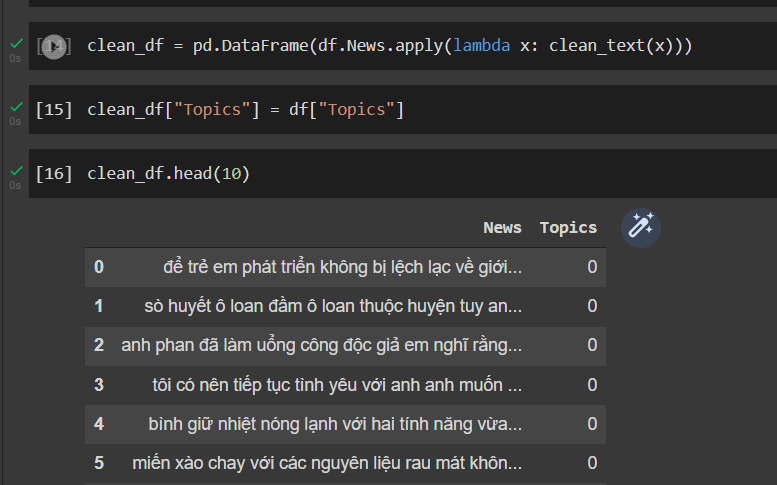


## Làm sạch dữ liệu

1. Chuyển các topics sang số để chạy các thuật toán máy học. Số là vị trí của tiêu đề trong topics 
2. Tạo hàm dọn dữ liệu đầu vào:



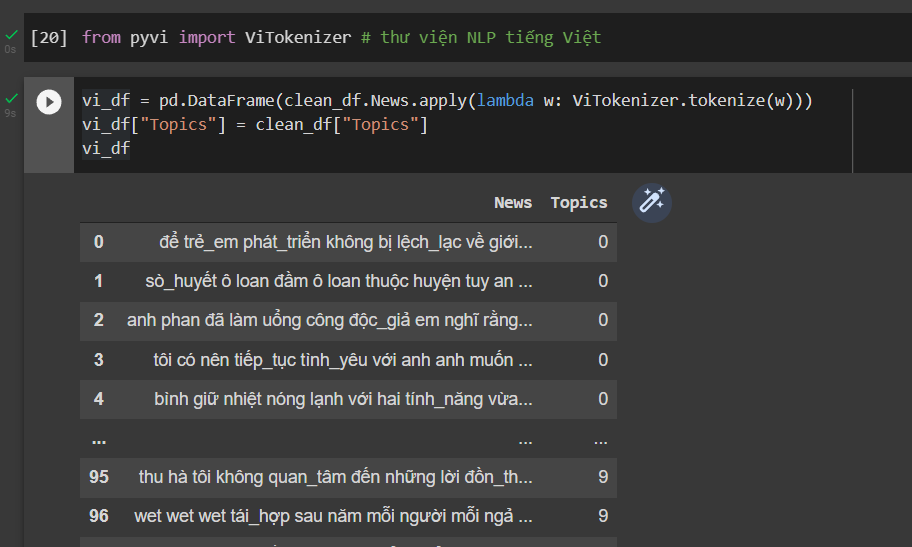
1. Dọn dữ liệu trong News:

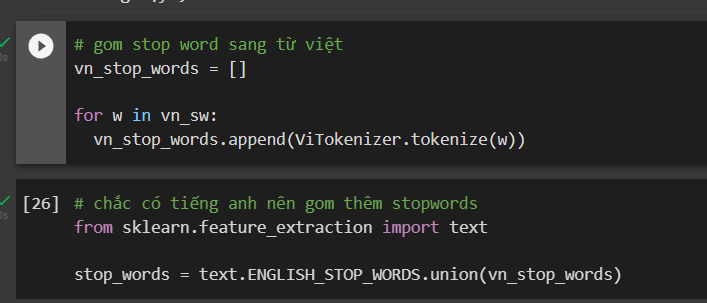


1. Tách từ

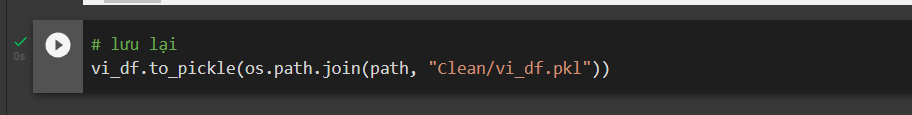
Sử dụng thư viện pyvi để tìm các từ tiếng việt gom các từ ghép thành dạng trẻ em -> trẻ\_em để tạo 1 từ gọi là token, vì tiếng anh các từ cách nhau bởi khoảng trắng.

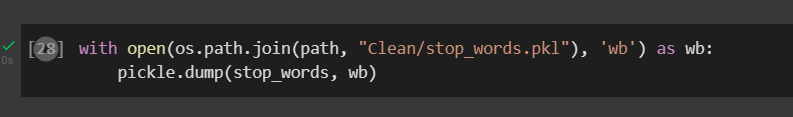
* 1. Đầu tiên sẽ token hóa các News:



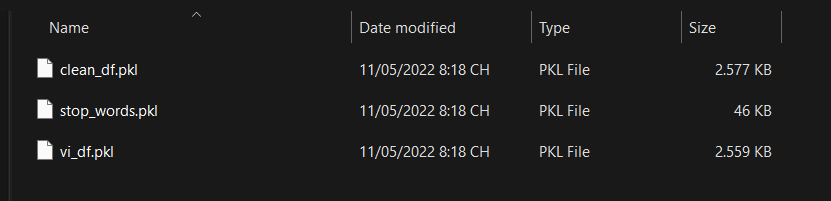
* 1. Sau đó sẽ token các stop\_words tiếng việt và thêm stop\_words của tiếng anh:
  2. Lưu files

Các file sẽ được lưu lại vào thư mục Clean để chuẩn bị cho bước tiếp theo:





Lưu thêm các file của các bước Src



## Phân tích dữ liệu

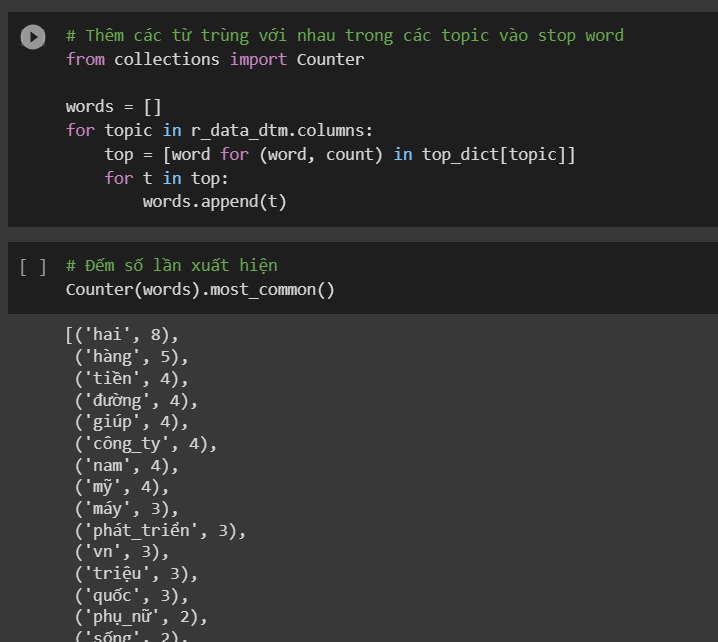
Load các dữ liệu bên clean

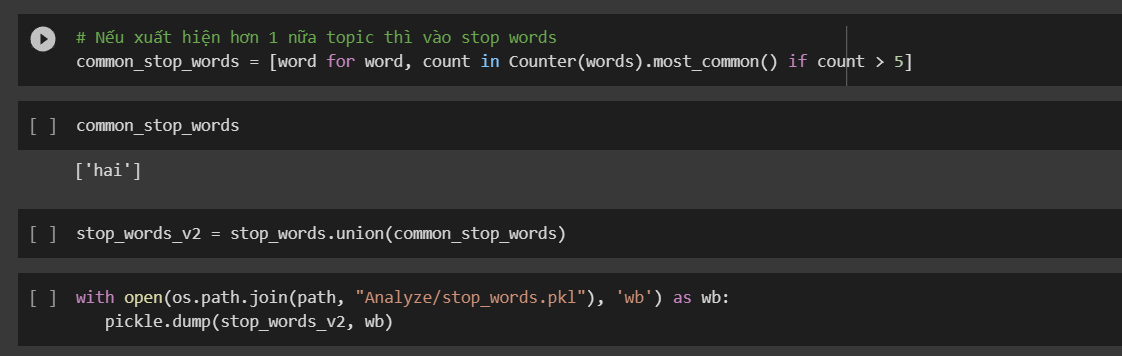
1. Dùng word cloud để phân tích
2. Top các từ xuất hiện nhiều trong các tiêu đề



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

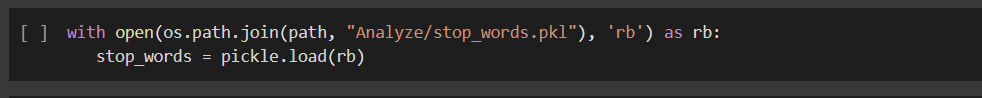


1. Thêm các từ xuất hiện trên 5 topics vào stop\_words vào lưu vào Analyze

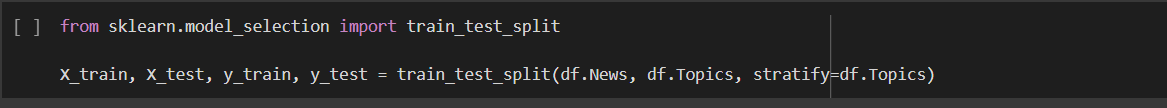
## Train model

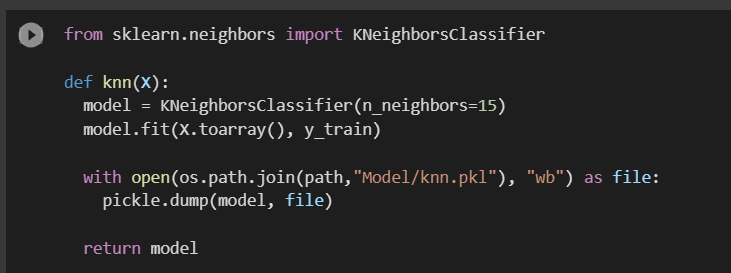
1. Load các dữ liệu topics và vi\_df bên Clean và stop\_words vừa phân tích

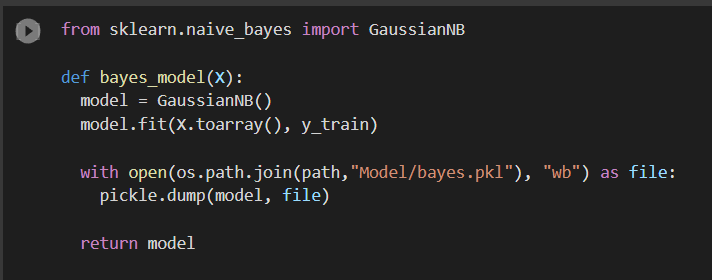


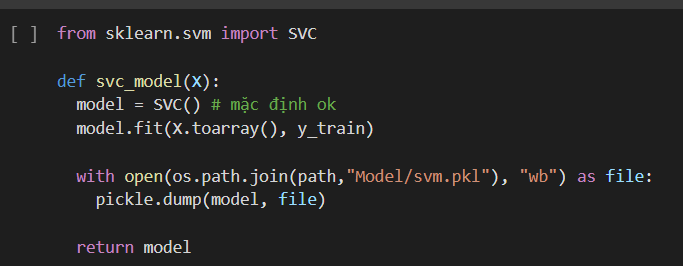


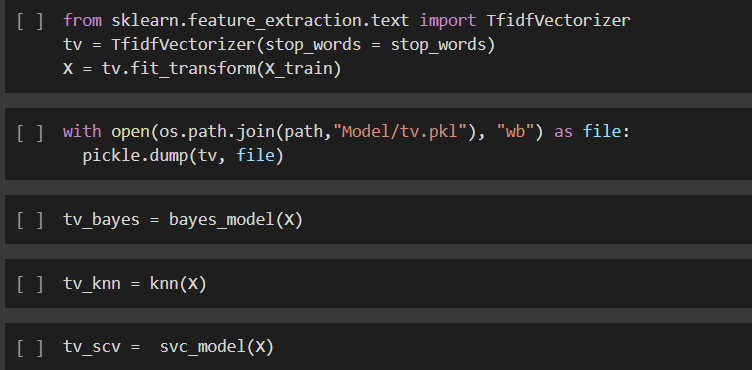
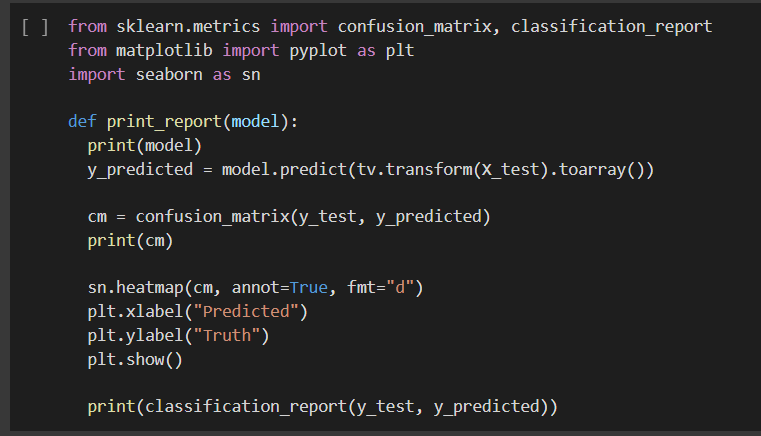
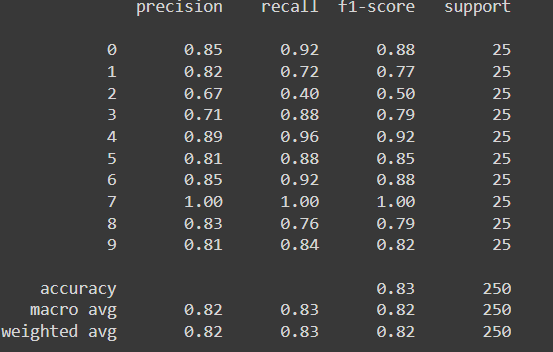
1. Chia test và train

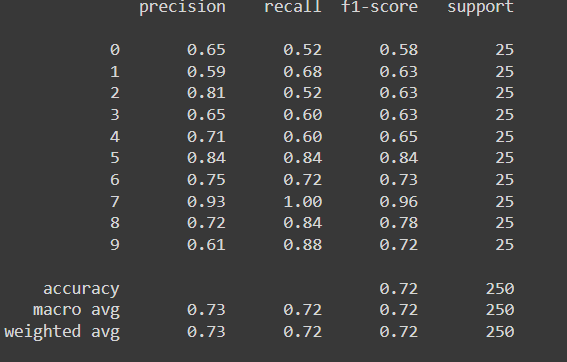


1. Train model theo 3 thuật toán K-nearest neighbors, Naive Bayes và SVM

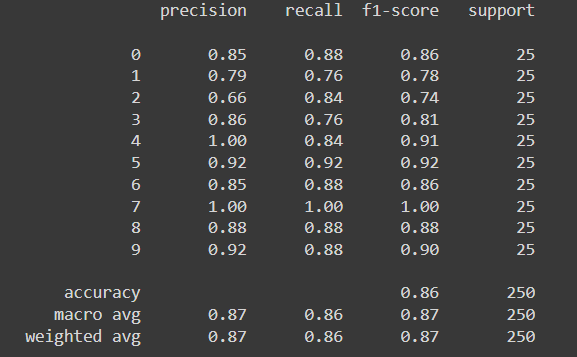




1. Chuyển các news sang số sử dụng tf-idf và train các model
2. Tạo hàm vẽ confusion matrix và tính F1
   1.  K-nearest neighbors
   2. Naive Bayes



* 1. SVM



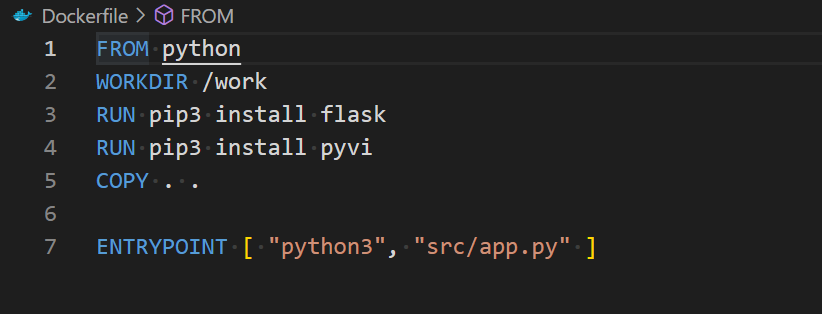
1. Bảng so sánh chỉ số F1

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

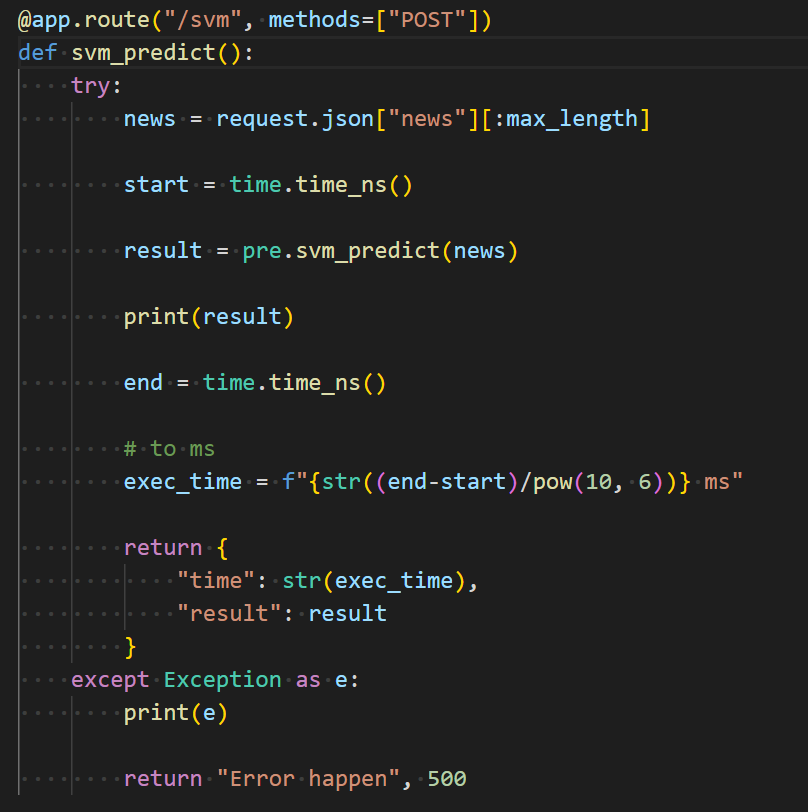
Vì SVM có F1 cao nhất nên sẽ sử dụng model này để deploy

# Deploy Model

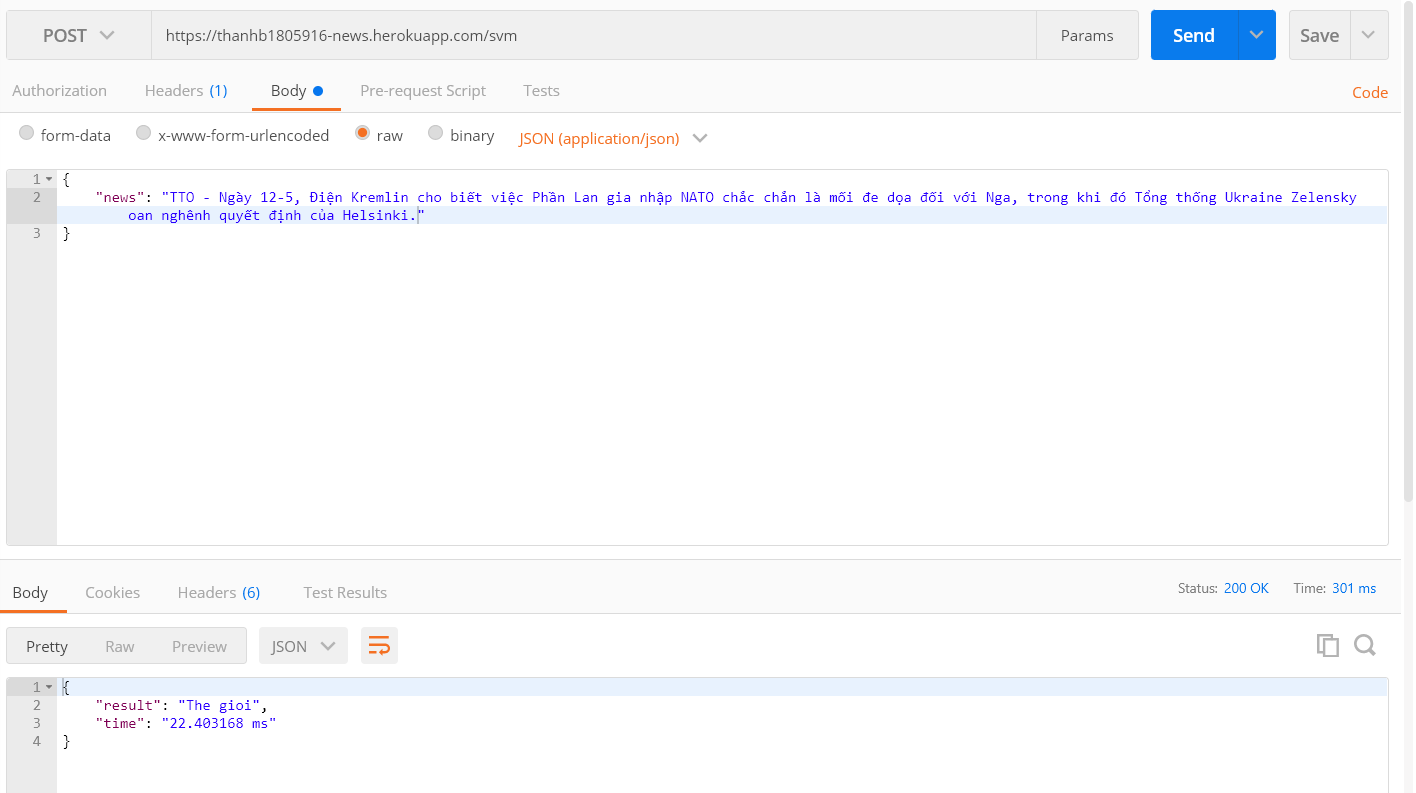
1. Sử dụng docker file

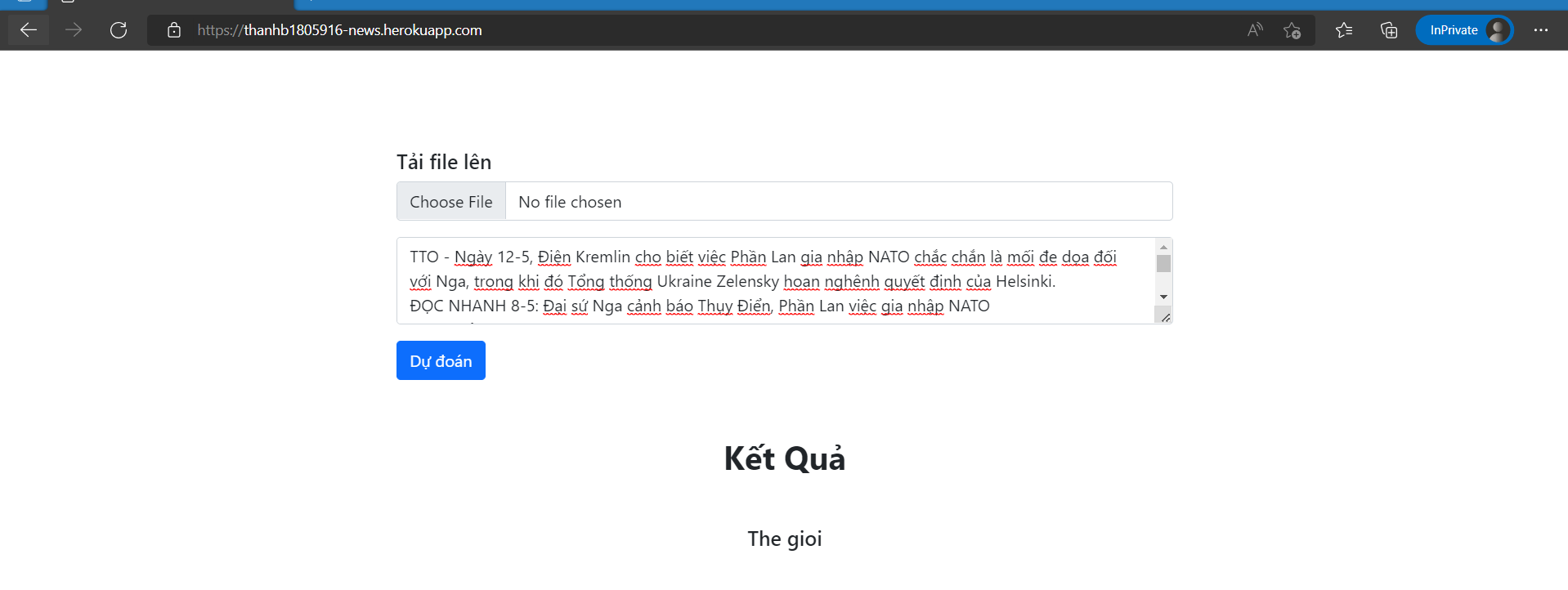


Dùng flask để tạo web service và pyvi để token hóa các dữ liệu đầu vào

1. Endpoints để dự đoán

Thử api đã deploy lên heroku bằng Postman



Giao diện cho người dùng