1 Crawl các bài báo mạng

Các mẫu dữ liệu được được thực hiện theo từng bài báo

2 Báo Tuổi trẻ: https://tuoitre.vn/

2.1 Hiện thực

1. Crawl 10000 bài báo
2. Làm sạch dữ liệu (Clean data)
3. Train data
4. Dự đoán các thể loại với nội dung báo bất kỳ
5. Tìm tất cả các báo đã crawl về theo thể loại

2.2 Crawl bài báo

Sử dụng thư viện

Requests

- Module Request dùng để gửi HTTP request, giống như thao tác chúng ta thường làm khi lướt mạng : Vào trình duyệt gõ bất kỳ và enter, bạn sẽ nhận được giao diện của trang web hoặc một dạng dữ liệu khác. Để lấy được dữ liệu trả về thì ta phải sử dụng một module hỗ trợ và Request sẽ giúp chúng ta làm điều đó.
- Cài đặt: *pip install requests (hoặc python –m pip install requests)*

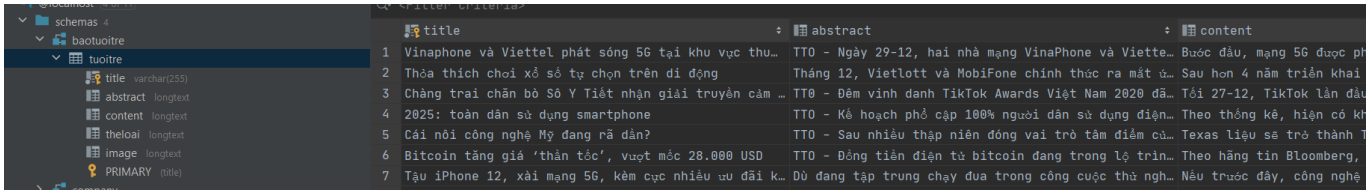
BeautifulSoup

- Có nhiều cách để bóc tách dữ liệu từ một văn bản dài, sử dụng regex (biểu thức chính quy) cũng là một cách nhưng thực tế thì python đã hỗ trợ mạnh hơn. Ở project này mình sử dụng module beautifulSoup4 để tách dữ liệu.
- Để dùng đầu tiên ta lên trang web cần crawl và bấm f12 để xem source và tìm các thẻ và class chứ các nội dung cần crawl ví dụ như:

- Module BeautifulSoup hỗ trợ hàm findAll(...) để tách dữ liệu ra *titles = soup.findAll('h3', class_='title-news')*
- Cài đặt: *pip install beautifulsoup4 (hoặc python –m pip install beautifulsoup4)*

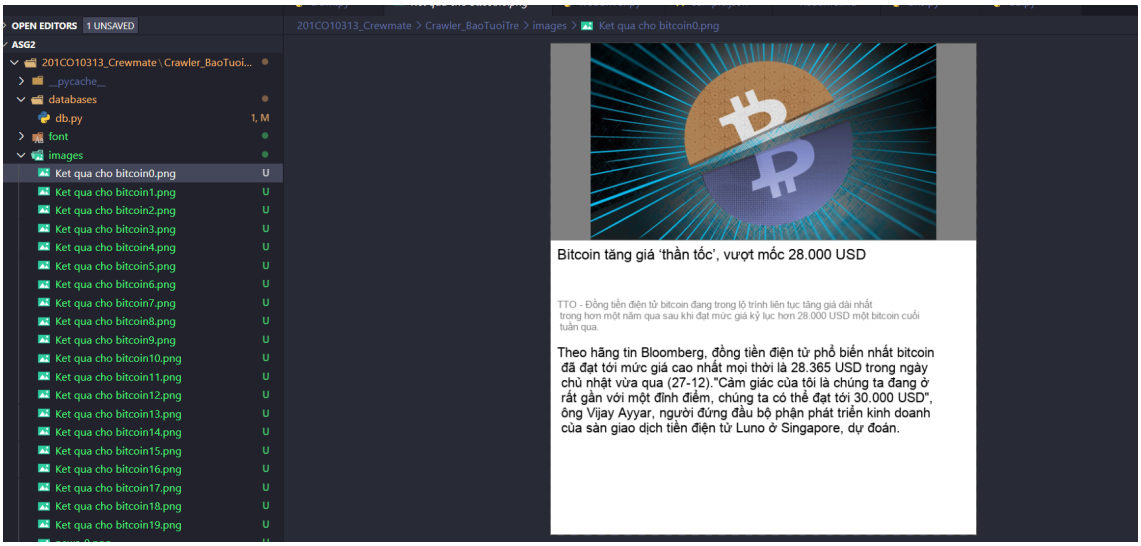
Kết hợp mysql với python

- Sử dụng thư viện mysql để lưu các bài báo crawl được vào database. Để title là primary key để cho các bài báo không trùng nhau
- Kết quả:



PIL

- Pillow là module hỗ trợ xử lí file ảnh khá thân thiện và dễ sử dụng. Mình sử dụng Pillow để đọc và ghi ảnh, viết chữ lên ảnh...
- Với mỗi bài báo crawl về mình sẽ mô phỏng lại bằng thư viện pil và đây là ví dụ:



- Cài đặt: *pip install Pillow (hoặc python –m pip install Pillow)*

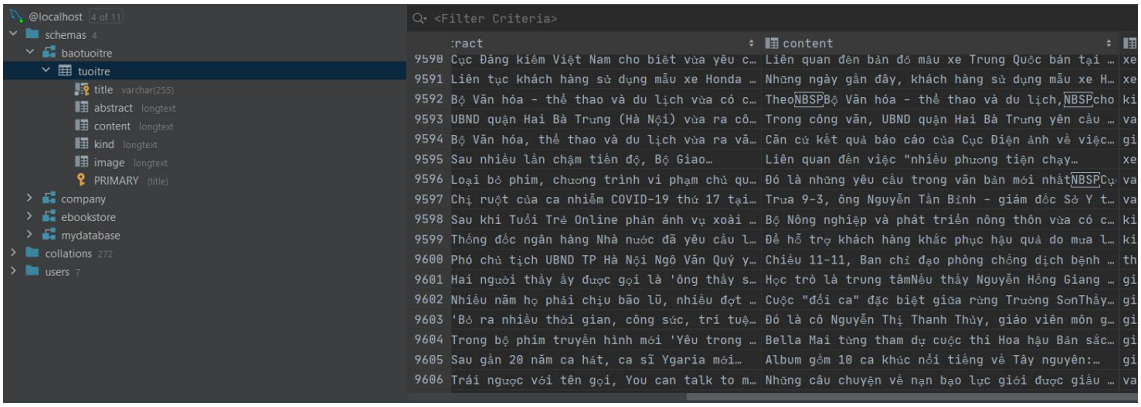
2.3 Dataset

2.3.1 Crawl data

Đã crawl được hơn 10000 bài báo. Các bài báo đảm bảo không trùng nhau vì trường **title** được set là primary key.
Các bài báo được chia ra thành 8 thể loại là:

1. Giải trí: Crawl 1577 bài
2. Giáo dục: Crawl 1245 bài
3. Khoa học: Crawl 1328 bài
4. Kinh doanh: Crawl 1411 bài
5. Thế giới: Crawl 1411 bài
6. Thời sự: Crawl 1412 bài
7. Văn hóa: Crawl 1328 bài
8. Xe: Crawl 300 bài

Dữ liệu đã được lưu ở 2 nơi một là database, hai là file csv (file /csv/paperCrawler_file.csv đã được push lên git).



2.3.2 Clean data

Dữ liệu khi được crawl về đã được làm sạch bằng cách:

1. **Loại bỏ stop words**: Những từ thường xuyên xuất hiện như là 'và', 'của', 'các',v.v, được loại bỏ vì chúng có thể gây nhiễu cho dữ liệu, và chúng không gây ảnh hưởng nhiều đến dữ liệu.
2. **Lemmatization**: Tất cả các từ sẽ được đưa về dạng ký tự thường (không phải HOA).

3. **Loại bỏ non-words**: các dấu câu, và các ký tự đặc biệt đã được loại bỏ.

2.3.3 Train data

Sau khi làm sạch dữ liệu mình tiến hành train data để có thể dự đoán được nội dung của một bài báo bất kỳ sẽ có thể loại là gì.

Sử dụng mô hình **naive_bayes** đã được hỗ trợ bởi module **sklearn**

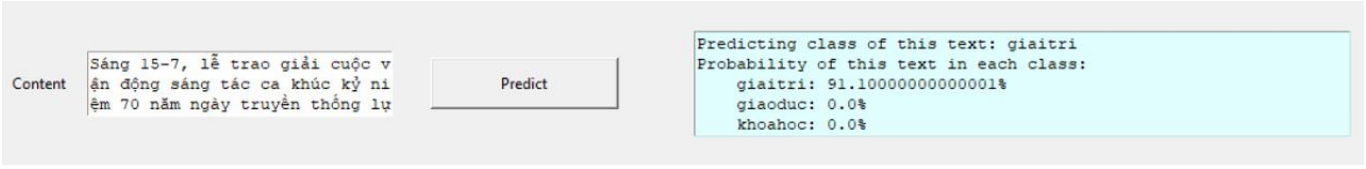
Đầu tiên để sử dụng mô hình **naive_bayes** ta phải có tập từ điển gồm các từ xuất hiện nhiều nhất ở các báo. Để tìm được tập đó mình đã sử dụng một mô hình khác là **DecisionTreeClassifier**. Với mô hình này mình có thể tìm ra các từ xuất hiện nhiều nhất ở một thể loại bài báo. Và ở mỗi thể loại bài báo mình đã tìm từ 20 đến 25 từ xuất hiện nhiều nhất và tổng cộng đã kiếm được 180 từ xuất hiện nhiều nhất ở 8 thể loại bài báo khác nhau.

Sau đó mình dùng data mà mình đã crawl được để làm tập huấn luyện. Để làm tập huấn luyện cho mô hình **naive_bayes** thì mình phải biến dữ liệu chữ ở trường content của các bài báo thành ma trận số. Đầu tiên mình sẽ tạo ra ma trận 8000x180 (8000: độ lớn của tập huấn luyện, 180: độ dài của tập từ điển) có các giá trị là 0. Mình sẽ chọn ngẫu nhiên ra 8000 bài báo ngẫu nhiên và tách content của mỗi bài báo đó ra thành từng chữ và so chúng với tập từ điển nếu từ nào trong bài báo có trong tập từ điển thì vị trí của từ đó trong tập từ điển sẽ được cộng thêm 1.

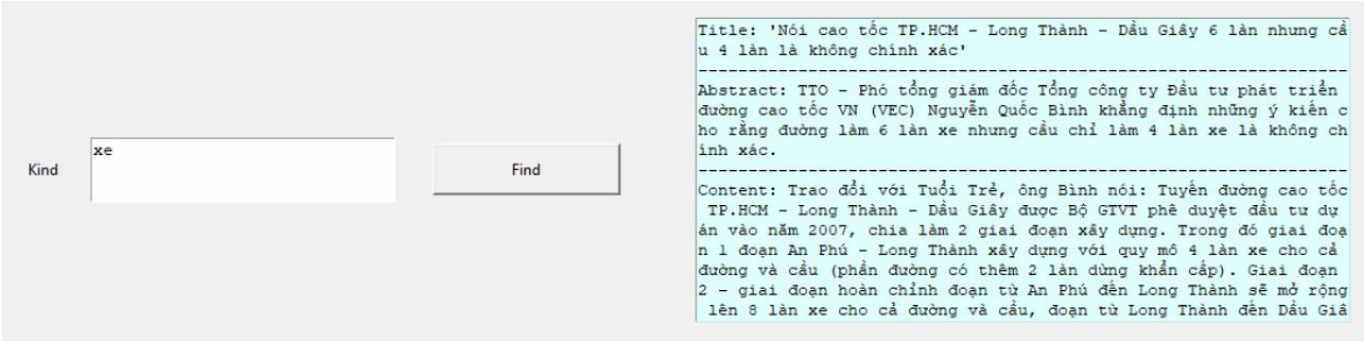
Như vậy mình đã tạo ra được tập huấn luyện và có thể dự đoán các nhãn của các nội dung của các bài báo bất kỳ.

2.4 Tính năng

1. Crawl dữ liệu theo chủ đề: Chương trình sẽ crawl lần lượt theo các trang theo từng chủ đề như: thời sự, giáo dục, văn hóa, giải trí, thể thao,...
2. Crawl dữ liệu theo từ khóa: Chương trình sẽ crawl các bài báo có chứa từ khóa nhập vào.
3. Dự đoán nội dung của một bài báo bất kì thuộc thể loại nào



4. Tìm các bài báo theo thể loại đã được crawl



2.5 Link github: https://github.com/FriedCorn/201CO10313_Crewmate/tree/main