**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN HÀ NỘI**

**KHOA TOÁN – CƠ – TIN HỌC**

**----\*\*\*----**



**BÀI BÁO CÁO**

***Đề tài: Tạo web phỏng đoán tin tức thật giả bằng Python***

***Giảng viên hướng dẫn: PGS.TS. Lê Trọng Vĩnh***

***Nhóm sinh viên thực hiện : Nhóm 5***

***Lớp học phần: Lập trình nâng cao***

***Hà Nội, tháng 11/2022.***

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN HÀ NỘI**

**KHOA TOÁN – CƠ – TIN HỌC**

**----\*\*\*----**



**BÀI BÁO CÁO**

***Đề tài: Tạo web dự đoán tin tức thật giả bằng Python***

**Thành viên nhóm 5:**

1. Đỗ Thị Hường – 20001929
2. Ngô Thị Thu Hằng – 20001913
3. Nguyễn Thị Thu Hiền – 20001914

# 1. Mô phỏng bài toán và quy trình làm việc:

## 1.1 Mô phỏng bài toán:

Web sẽ giúp người dùng phỏng đoán tin tức được nhập vào theo 2 cách là: Nhập url của trang web chứa tin tức hoặc nhập tin tức thủ công bằng Input Box có trên trang web. Bài toán sẽ được chia nhỏ thành 2 bài toán con: Xử lý dữ liệu để đưa ra phỏng đoán về thông tin được truyền vào và tạo lập trang web với PyScript.

**Phần 1:** Xử lý dữ liệu để đưa ra phỏng đoán về thông tin được truyền vào: Xử lý dữ liệu traning và lưu lại kết quả, từ những traning có được sẽ phỏng đoán dữ liệu chứ tin tức mới nhập được từ người dùng là thật hay giả.

Tập dữ liệu trainning của bài toán được lấy từ <https://www.kaggle.com/c/fake-news/data?select=train.csv>. **train.csv** là một tập dữ liệu training đầy đủ với các thuộc tính sau:

id: id duy nhất cho mỗi tin tức

title: tiêu đề của tin tức

author: tác giả của tin tức

text: văn bản của bài viết, có thể không đầy đủ

label: nhãn đánh dấu bài viết có khả năng không đáng tin cậy

+ 1: không đáng tin cậy

+ 0: đáng tin cậy

**Phần 2:** Xây dựng trang web với PyScript.

**<Viết thêm vào>**

**Phần 3:** Một số phương pháp khác để lấy dữ liệu cần dự đoán.

## 1.2 Quy trình làm việc:

- Chuẩn hóa dữ liệu đầu vào, để tối ưu hóa phân tích dữ liệu.

- Chuyển đổi dữ liệu văn bản thành các vector để máy tính có thể hiểu được.

- Ghi các kết quả bước vừa rồi vào model để phỏng đoán.

- Lưu trữ lại các kết quả training được.

- Tạo lập web để nhận các dữ liệu cần phỏng đoán.

- Chuẩn hóa dữ liệu và chuyển nó thành các vector.

- Truyền nó vào model đã tạo ở các bước trên để model đưa ra phỏng đoán.

- Xây dựng web cho phép nhập nhập dữ liệu và đưa ra phỏng đoán.

- Xử lý để lấy dữ liệu cần dự đoán khi dữ liệu được truyền theo dạng khác.

# 2. Chi tiết thực hiện:

## 2.1 Phần 1: Xử lý dữ liệu để đưa ra phỏng đoán về thông tin được truyền vào:

### 2.1.1: Giới thiệu các thư viện sử dụng:

[Pandas](https://pandas.pydata.org/) cung cấp các chức năng và cấu trúc dữ liệu cấp cao được thiết kế để làm việc với dữ liệu có cấu trúc hoặc dạng bảng nhanh chóng, dễ dàng và rõ ràng. Nó cung cấp chức năng lập chỉ mục để giúp dễ dàng định hình lại, cắt, tổng hợp và chọn tập hợp con dữ liệu.

[Regex](https://docs.python.org/3/library/re.html) Biểu thức chính quy (Regular Expressions) trong Python có thể được định nghĩa là chuỗi các ký tự được sử dụng để tìm kiếm một mẫu trong chuỗi. Mô-đun re cung cấp hỗ trợ để sử dụng regex trong chương trình python.

[NLTK](https://www.nltk.org/) (Bộ công cụ ngôn ngữ tự nhiên - Natural Language Toolkit): là một nền tảng hàng đầu để xây dựng các chương trình Python hoạt động với dữ liệu ngôn ngữ tự nhiên. Nó cung cấp các giao diện dễ sử dụng cho hơn 50 kho ngữ liệu và tài nguyên từ vựng như WordNet, cùng với một bộ thư viện xử lý văn bản để phân loại, thông báo, tạo gốc, gắn thẻ, phân tích cú pháp và lý luận ngữ nghĩa...Các thư viện con sử dụng:

+ [NLTK corpus readers(nltk.corpus)](https://www.nltk.org/_modules/nltk/stem/porter.html): Các mô-đun trong gói này cung cấp các chức năng có thể được sử dụng để đọc các tệp văn bản ở nhiều định dạng khác nhau. Các chức năng này có thể được sử dụng để đọc cả tệp kho văn bản được phân phối trong gói kho văn bản NLTK và các tệp văn bản là một phần của kho văn bản bên ngoài.

+ [nltk.stem.porter](https://www.nltk.org/_modules/nltk/stem/porter.html): Loại bỏ các phụ tố hình thái từ các từ, chỉ để lại từ gốc.

[Scikit-learn (sklearn)](https://scikit-learn.org/stable/) là một mô-đun Python dành cho máy học được xây dựng dựa trên SciPy. Các thư viên con đã sử dụng:

+ [sklearn.feature\_extraction:](https://scikit-learn.org/stable/modules/feature_extraction.html) mô-đun này có thể được sử dụng để trích xuất các tính năng ở định dạng được thuật toán máy học hỗ trợ từ các bộ dữ liệu bao gồm các định dạng như văn bản và hình ảnh. Chương trình này sử dụng định dạng hình ảnh, nên mô-đun sử dụng sẽ là sklearn.feature\_extraction.text.

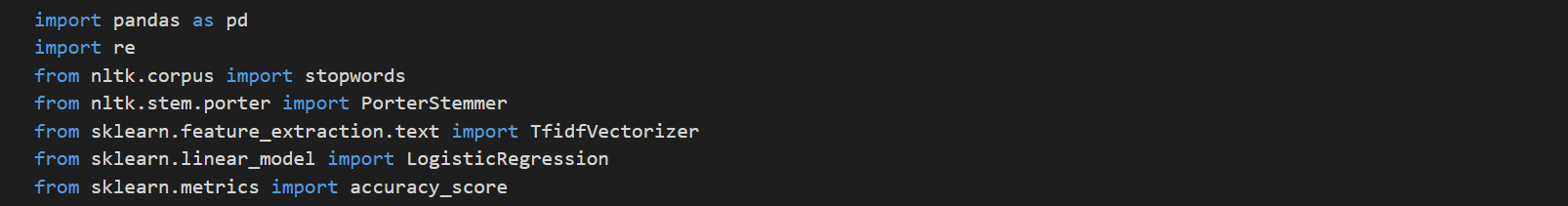
+ [sklearn.linear\_model](https://scikit-learn.org/stable/modules/linear_model.html): một tập hợp các phương thức dành cho hồi quy, trong đó giá trị mục tiêu dự kiến ​​là sự kết hợp tuyến tính của các feature.

+ [sklearn.metrics](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics): định lượng chất lượng dự đoán theo cặp.

### 2.2.2 Chi tiết thực hiện:

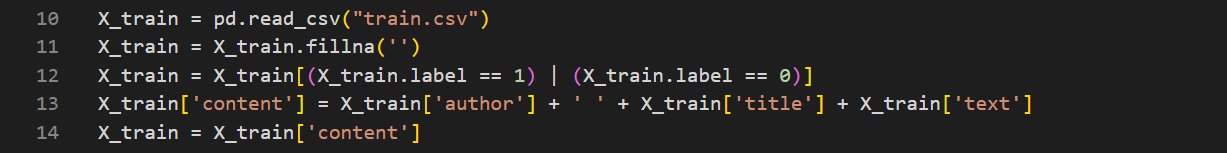
**Tạo model dự đoán và lưu trữ các model này, file đính kèm** [**training.py**](Part1/training.py)

+ Import các thư viện cần thiết:



Công dụng của các thư viện này đã được giải thích cơ bản ở trên, khi được xử dụng sẽ nói chi tiết về việc dùng thư viện đó trong bài toán như thế nào.

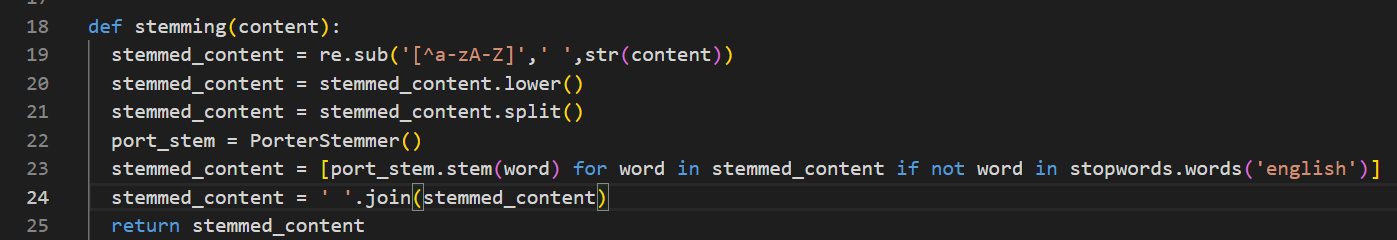
+ Đọc dữ liệu và bỏ đi các bản ghi không phù hợp với pandas:



* Phương thức pandas.read\_cvs() giúp đọc tệp giá trị được phân tách bằng dấu phẩy (csv) vào DataFrame, X\_train có type là DataFrame.
* DataFrame.fillna(‘’) giúp tìm tất cả các giá trị NaN và thay nó thành chuỗi rỗng.
* Loại bỏ các bản ghi với label khác 0 và 1, chúng không mang ý nghĩa gì với quá trình traing.
* Hợp nhất các trường mang lại giá trị đó là: author, title và text để sau chuyển đổi chúng trở thành các feature value của bài toán.

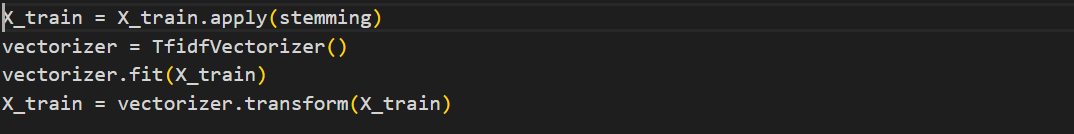
+ Lưu trữ trường label riêng vì chúng đã là dạng số mà máy tính có thể hiểu được, và đồng thời là predict value của các bản ghi training của bài toán.

+ Hàm stemming(content) sau sẽ giúp chuẩn hóa các feature:



* Đầu tiên hàm stemming sẽ loại bỏ hết các kí tự đặc biệt và thay bằng chuỗi trống. Thư viện re được dùng ở đây. Các chữ hoa sẽ được chuyển thành chữ viết thường.
* Dữ liệu được chia nhỏ bởi các space.
* stopwords là một module thuộc nltk.corpus, chúng chứa các từ nối các từ không mang nhiều giá trị, ví dụ như: “and”, “or”, “but”... Ở đây chúng ta làm việc với tiếng Anh, chú ý stopwords cần được download trước khi xử dụng. Chúng ta sẽ loại bỏ tất cả các stopwords này.
* Sử dụng phương thức PorterStemmer.stem(word) để loại bỏ các tiền tố và hậu tố, đưa từ trở về từ gốc. Ví dụ training sẽ trở thành train, visited sẽ thành visit...
* Sau đó các từ được nối lại và trả về dữ liệu đã được chuẩn hóa.

+ Chuyển các feature từ dạng văn bản sang vector với TfidVectorizer:



* X\_train sẽ được chuẩn hóa bởi apply(stemming)
* Ta sẽ sử dụng class TfidVectorizer của sklearn.feature\_extraction. Trước tiên giải thích về TfidVectorizer: