**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***

***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

MỤC LỤC

[1. Mô phỏng bài toán và quy trình làm việc: 1](#_Toc121323820)

[1.1 Mô phỏng bài toán: 1](#_Toc121323821)

[1.2 Quy trình làm việc: 2](#_Toc121323822)

[2. Chi tiết thực hiện: 3](#_Toc121323823)

[2.1 Phần 1: Xử lý dữ liệu để đưa ra phỏng đoán về thông tin được truyền vào: 3](#_Toc121323824)

[2.1.1: Giới thiệu các thư viện sử dụng: 3](#_Toc121323825)

[2.2.2 Chi tiết thực hiện: 4](#_Toc121323826)

[2.2 Phần 2 : Xây dựng trang web với PyScript 13](#_Toc121323827)

[2.2.1 : Giới thiệu chung về PyScript. 13](#_Toc121323828)

[2.2.2: Chương trình đầu tiên với PyScript: Hello World 15](#_Toc121323829)

[2.2.3: Sử dụng PyScript để tìm và thao tác các phần tử HTML 16](#_Toc121323830)

[2.3 Phần 3: Một số phương pháp khác để lấy dữ liệu cần dự đoán: 20](#_Toc121323831)

[2.3.1 Xử lý dữ liệu hàng loạt từ các tệp .csv, tương tự như cách tập training được nhận vào, xem tệp predict\_csv.py: 20](#_Toc121323832)

[2.3.2. Crawl dữ liệu web 21](#_Toc121323833)

[3. Tổng kết: 24](#_Toc121323834)

[4. Tài liệu tham khảo: 24](#_Toc121323835)

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

## 1.1 Mô phỏng bài toán:

Đặt vấn đề: Khi các phương tiện truyền thông xã hội trỗi dậy, phát triển mạnh mẽ đã phá vỡ nhiều ranh giới ngăn chặn các tin tức giả mạo lan truyền. Nếu trước đây, tin tức được kiểm duyệt chặt chẽ trước khi đăng tải chính thức trên phương tiện truyền thông, thì truyền thông mạng xã hội trao quyền cho bất cứ ai đều có thể tạo ra và truyền bá thông tin, đặc biệt là những người được coi là có sức ảnh hưởng trên mạng xã hội. Việc phân biệt được tin tức thật giả trở nên cần thiết hơn, thực dụng hơn và đang được nghiên cứu, phát triển. Đó là một bài toán dành cho nhiều lĩnh vực, khoa học, xã hội học, kinh tế học và tin học, dự án này sẽ mô phỏng một cách đơn giản cách dùng thuật toán để dự đoán tin giả. Dự đoán đưa ra có thể không chính xác 100% nhưng sẽ cho người đọc khả năng dự đoán và cẩn thận hơn với tin giả. Vì lời giải được sử dụng dựa trên ngôn ngữ lập trình Python và sử dụng thư viện ngôn ngữ nttk chưa có một cơ sở dữ liệu tiếng Việt đủ mạnh nên bài toán sẽ tạm thời được phát triển với ngôn ngữ tiếng Anh.

Web sẽ giúp người dùng phỏng đoán tin tức được nhập vào theo cách là: Nhập tin tức bằng Input Box có trên trang web. Bài toán này 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, ngoài ra phần 3 sẽ giúp lấy dữ liệu theo một số các khác.

**Phần 1:** Xử lý dữ liệu để đưa ra phỏng đoán về thông tin được truyền vào: Xử lý dữ liệu training và lưu lại kết quả, từ những mẫu training có được sẽ phỏng đoán dữ liệu chứa 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.

* Xây dựng giao diện trang web với html.
* Sử dụng PyScript để chạy code Python ngay trên Web Browser.

**Phần 3:** Một số phương pháp khác để lấy dữ liệu cần dự đoán: Dự án này cung cấp 2 cách lấy dữ liệu khác cách nhập thủ công từ người dùng:

* Lấy thông tin từ các trang web, báo mạng thông qua url nhập vào.
* Xử lý dữ liệu hàng loạt từ các tệp .csv, tương tự như cách tập training được nhận vào.

## 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.

[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, tiền tố, hậu tố chỉ để lại từ gốc.

[Scikit-learn (sklearn)](https://scikit-learn.org/stable/) là một mô-đun Python dành cho máy học mã nguồn mở hỗ trợ học tập có giám sát và không giám sát. Nó cũng cung cấp nhiều công cụ khác nhau để điều chỉnh mô hình, tiền xử lý dữ liệu, lựa chọn mô hình, đánh giá mô hình và nhiều tiện ích khác.. 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 văn bản, 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.

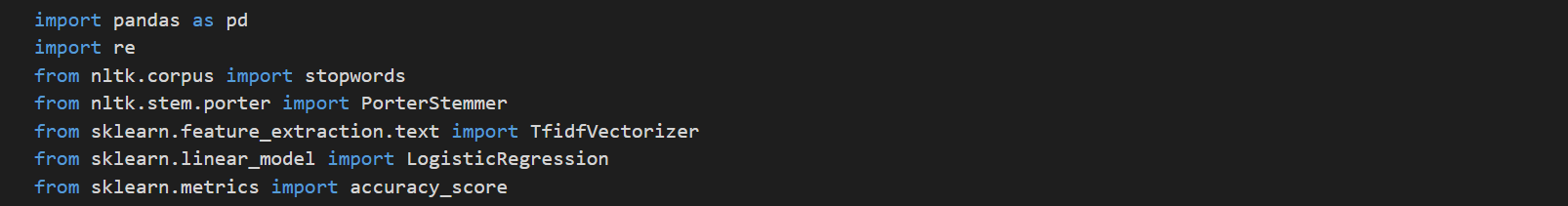
[Regex](https://docs.python.org/3/library/re.html) Biểu thức chính quy (Regular Expressions) trong Python có thể được định nghĩa là chuỗi các ký tự được sử dụng để tìm kiếm một mẫu trong chuỗi. Mô-đun re cung cấp hỗ trợ để sử dụng regex trong chương trình python.

[Pickle](https://docs.python.org/3/library/pickle.html) Mô pickle-đun thực hiện các giao thức nhị phân để tuần tự hóa và hủy tuần tự hóa cấu trúc đối tượng Python. “Pickling” là quá trình theo đó hệ thống phân cấp đối tượng Python được chuyển đổi thành luồng byte và “unpickling” là thao tác nghịch đảo, theo đó luồng byte (từ tệp nhị phân hoặc đối tượng giống byte) được chuyển đổi trở lại thành hệ thống phân cấp đối tượng.

### 2.2.2 Chi tiết thực hiện:

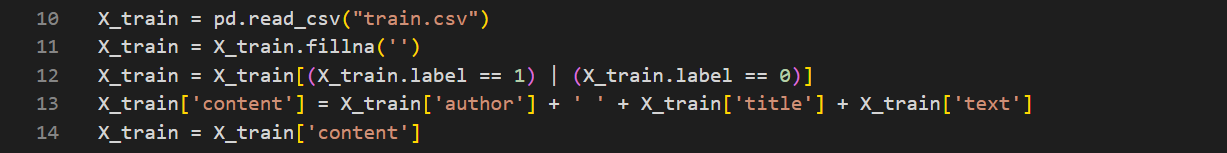
**Tạo model dự đoán và lưu trữ các model này, file đính kèm training.py**

\* 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.

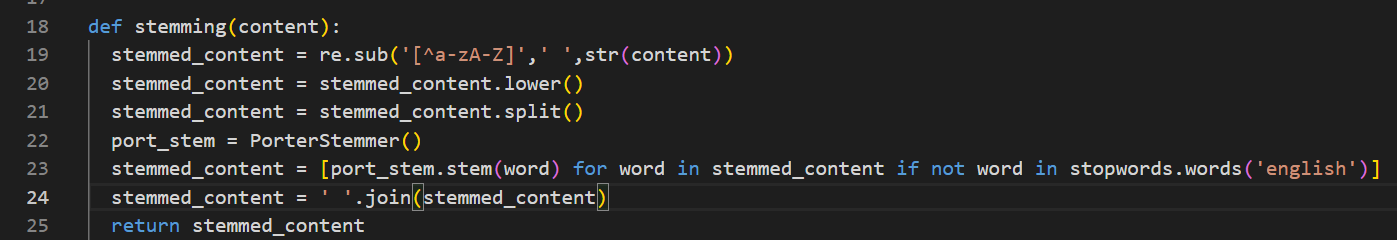
\* Đọc dữ liệu và bỏ đi các bản ghi không phù hợp với pandas:



* 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 training.
* Hợp nhất các trường mang lại giá trị đó là: author, title và text để sau chuyển đổi chúng trở thành các feature value (giá trị dữ liệu đặc trưng) của bài toán.

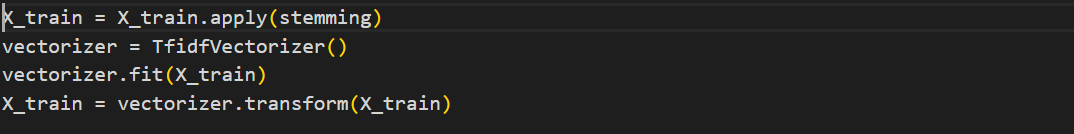
\* Lưu trữ trường label riêng vì chúng đã là dạng số mà máy tính có thể hiểu được, và đồng thời là predict value của các bản ghi training của bài toán.

\* Hàm stemming(content) sau sẽ giúp chuẩn hóa các feature: Đây là một bước quan trọng trong bất cứ phân tích nào nhất là trong machine learning, giữ nguyên dữ liệu gốc (raw documentation) thì khả năng những dữ liệu không cần thiết sẽ gây rối cho quá trình phân tích của máy tính, dẫn đến mất thời gian lâu hơn thậm chí dẫn đến những sai lệch không đáng có.



* 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. Hàm dựng của TfidVectorizer có rất nhiều thông số bổ sung, nhưng ở đây chúng ta chỉ cần một hàm dựng không đối số. Chi tiết về hàm dựng có thể xem ở [đây](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer.fit).Trước tiên giải thích về TfidVectorizer:

+ TF-IDF (Term Frequency - Inverse Document Frequency): là một thuật toán thống kê số học nhằm phản ánh tầm quan trọng của một từ đối với một văn bản trong một tập hợp hay một ngữ liệu văn bản. Đây là thuật toán rất phổ biến để chuyển đổi văn bản thành một biểu diễn có ý nghĩa của các con số, được sử dụng trong máy học.

+ TF- term frequency – tần số xuất hiện của 1 từ trong 1 văn bản. Số lần xuất hiện của một từ trong một văn bản càng lớn đương nhiên nó sẽ càng quan trọng. Cách tính:

Thương của số lần xuất hiện 1 từ trong văn bản và số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản đó. (giá trị sẽ thuộc khoảng [0, 1])

f(t,d) - số lần xuất hiện từ t trong văn bản d.

max{f(w,d):w∈d} - số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản.

+ IDF – inverse document frequency. Tần số nghịch của 1 từ trong tập văn bản (corpus). Tính IDF để giảm giá trị của những từ phổ biến. Mỗi từ chỉ có 1 giá trị IDF duy nhất trong tập văn bản. Ở trên ta đã khẳng định tính quan trọng của những từ xuất hiện nhiều lần tuy nhiên, nếu từ đó xuất hiện nhiều lần ở tất cả các văn bản thì có khả năng đó là từ phổ biến. Ví dụ: nếu đề cập đến chủ đề World Cup, thì từ này sẽ được đề cặp đến rất nhiều trong nhiều văn bản nhưng lại không mang đến quá nhiều giá trị. Cách tính:

|D|: tổng số văn bản trong tập D

: số văn bản chứa từ nhất định, với điều kiện t xuất hiện trong văn bản d. Nếu từ đó không xuất hiện ở bất cứ 1 văn bản nào trong tập thì mẫu số sẽ bằng 0 => phép chia cho không không hợp lệ, vì thế người ta thường thay bằng mẫu thức 1+.

Cơ số logarit trong công thức này không thay đổi giá trị của 1 từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Tuy nhiên việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng để dùng cho công thức TF-IDF như bên dưới.

+ Giá trị TF-IDF:

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

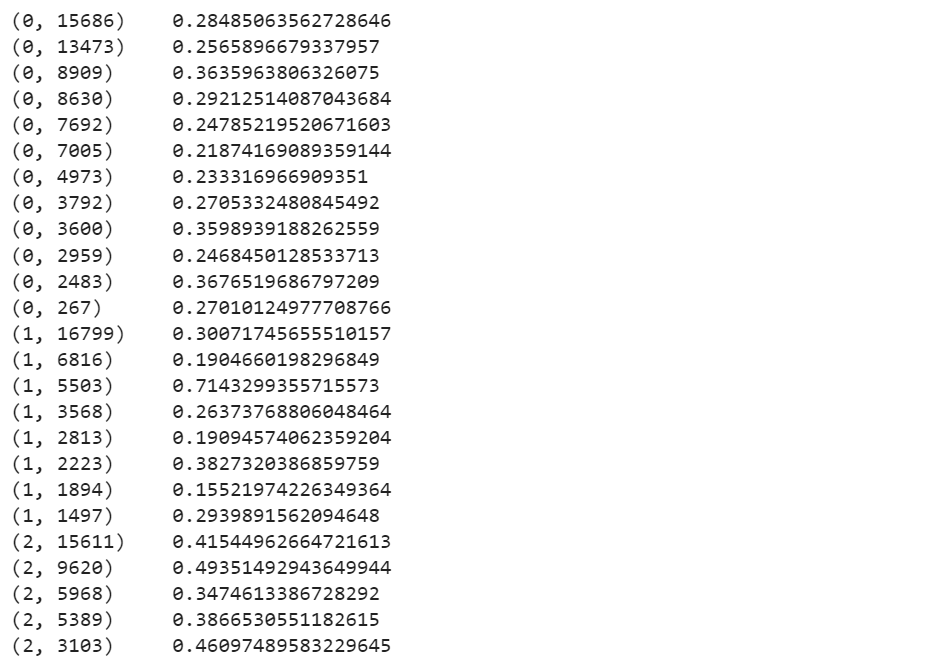
* Quay lại với TfidVectorizer được cung cấp sắn bởi Python, nhờ nó mà chúng ta có thể giải quyết những tính toán phức tạp ở trên một cách nhanh chóng:

+ TfidVectorizer.fit(raw\_documents): Học từ vựng và IDF từ bộ đào tạo, nó sẽ trả về TfidVectorizer phù hợp với tập training được đưa vào.

+ transform(raw\_documents): Chuyển đổi tài liệu thành ma trận tài liệu bởi những thứ mà nó học được từ hàm fit(raw\_documents) ở trên.

+ Ngoài ra có thể dùng hàm fit\_transform(raw\_documents) để thực hiện cả hai chức năng trên.

* Kết quả mà chúng ta nhận được sẽ có dạng như sau:



\* Xây dựng model dự đoán với LogisticRegression của sklearn.linear\_model:

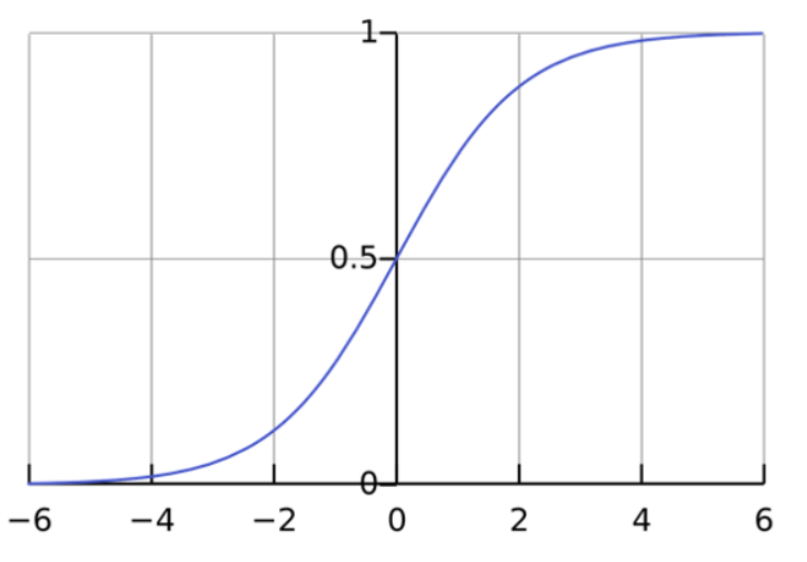
* Đầu tiên, ta sẽ tìm hiểu về thuật toán Logistic Regression (Hồi quy logicstic), là một thuật toán thuộc nhóm thuật toán phân lớp, thuật toán trả về một giá trị xác suất để ánh xạ đến hai hoặc nhiều lớp rời rạc

+ Hàm Sigmoid là hàm thường được sử dụng trong LogisticRegression (Python cũng dùng hàm này). Công thức của nó là:

Trong đó:

* y là xác suất dự đoán, nó luôn nằm trong khoảng từ [0,1] theo tính chất của hàm Sigmoid
* s = WX + b, với w là trọng số và X là input features, b là hệ số tự do, với bài toán có nhiều feature như bài toán của chúng ta WX = w1x1 + w2x2+… wnxn.

+ Nếu vẽ phương trình hồi quy logistic này, bạn sẽ có một đường cong hình chữ S như hình dưới đây:



+ Khi đó dự đoán phân lớp được đưa ra dựa trên các Threshold Value. Trong bài toán phân biệt tin tức thật/giả của chúng ta, quy ước Threshold Value = 0.5, y <0.5 là tin tức thật, y >= 0.5 là tin tức giả.

* Quay trở lại với LogisticRegression đã được dựng sẵn bởi Python. Hàm dựng tổng quát của LogisticRegression có thể xem ở [đây](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html). Chúng ta sẽ sử dụng hàm dựng không có đối số và xử lý dữ liệu như sau:



Hàm fit(X, y, sample\_weight=None)

Tham số:

* X: {array-like, sparse matrix} of shape (n\_samples, n\_features): Vectơ training, trong đó n\_samples là số lượng mẫu và n\_features là số lượng feature.
* y: array-like of shape (n\_samples,): Vectơ mục tiêu liên quan đến X.
* sample\_weightarray-like of shape (n\_samples,) default=None: Mảng trọng số được gán cho các mẫu riêng lẻ. Nếu không được cung cấp, thì mỗi mẫu được cung cấp trọng lượng đơn vị.

Returns: self, công cụ ước tính được trang bị.

\* Model đã được xây dựng nhưng công việc của chúng ta vẫn còn: kiểm tra độ chính xác của model. Việc kiểm tra này được dựa trên sự ăn khớp giữa predict chúng ta tìm được khi áp dụng model với tập X\_train và giá trị Y\_train thực tế đã có:

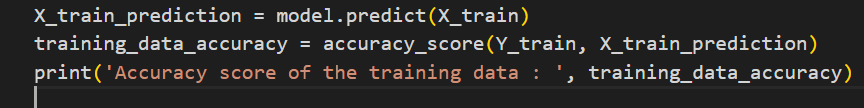
* Chúng ta sẽ dựa vào hàm sklearn.metrics.accuracy\_score(y\_true, y\_pred, \*, normalize=True, sample\_weight=None): Điểm phân loại độ chính xác. Trong phân loại nhiều nhãn, hàm này tính toán độ chính xác của tập hợp con: tập hợp nhãn được dự đoán cho một mẫu phải khớp chính xác với tập hợp nhãn tương ứng trong y\_true:

+ Tham số:

* y\_true: 1d array-like hoặc mảng nhãn/ma trận thưa, đó là nhãn chân lý (chính xác).
* y\_pred: 1d array-like hoặc mảng nhãn/ma trận thưa, nhãn dự đoán
* normalize: bool, mặc định=True. Nếu False, trả lại số lượng mẫu được phân loại chính xác. Nếu không, hãy trả lại số phần mẫu được phân loại chính xác.
* sample\_weight array-like of shape (n\_samples,), default=None, trọng lượng mẫu.

+ Return: float, dựa vào tham số normalize.

* Chúng ta sử dụng hàm này như thế nào:

Hàm predict(X): Dự đoán nhãn lớp cho các mẫu trong X:

+ Tham số: X {dạng mảng, ma trận thưa thớt} (n\_samples, n\_features), Ma trận dữ liệu muốn nhận dự đoán.

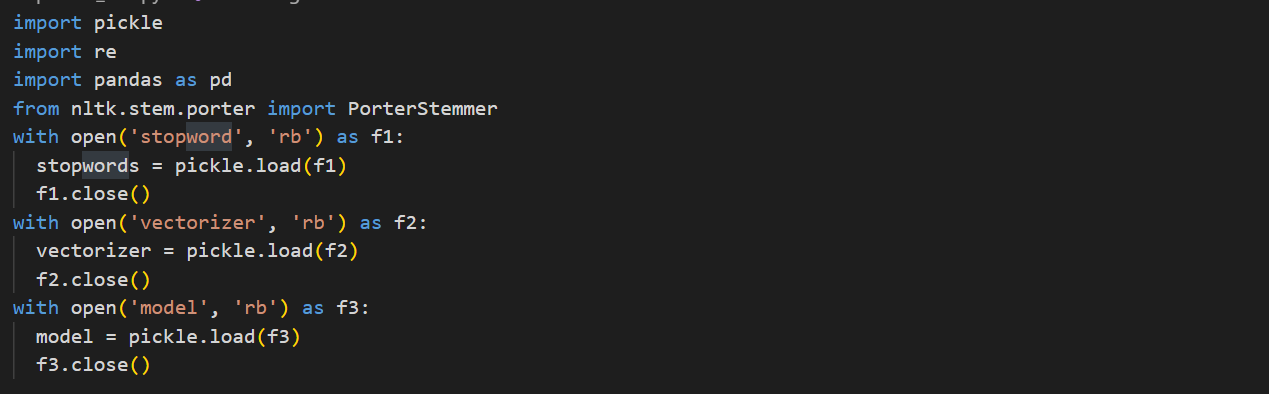
+ return: y\_pred nd-array dạng (n\_samples,), Vector chứa các nhãn lớp cho mỗi mẫu.

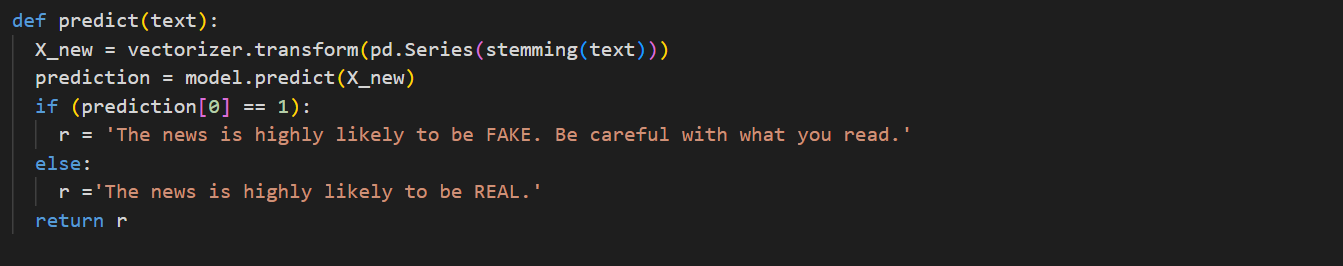
* Như vậy chỉ cần tính accuracy\_score của Y\_train, X\_train\_prediction, ta sẽ thấy được phần nào sự chính xác của model. Giá trị được tính ra là 0.9795192307692308, một độ chính xác khá cao.

\* Để tiết kiệm thời gian sử dụng model cho những phân tích sau này nhất là khi chúng ta sử dụng PyScript để chạy code Python sẽ gặp bất lợi về thời gian, chúng ta sẽ lưu lại giá trị của vectorizer để chuyển dữ liệu input thành các vector và model để dự đoán kết quả với việc sử dụng thư viện pickle. Ngoài ra ta cũng lưu lại giá trị của các stopwords để sử dụng dễ dàng hơn.



\* Hàm để sử dụng model khi có dữ liệu được truyền vào (dữ liệu dạng str) - theo dõi tệp predict\_text.py đính kèm:



* Import các thư viện cần thiết cũng như lấy lại các kết quả đã có với pickle.
* Chuẩn hóa text đầu vào và biến nó thành các vector với TfidVectorizer đã được training, chú ý phải chuyển text thành dạng pd.Series để tránh TypeError đối với hàm transform.
* Vì chỉ đưa vào model 1 bản ghi duy nhất lên prediction = model.predict(X\_new) mà chúng ta cần sẽ ở chỉ số 0 và giá trị prediction[0] để đưa ra phỏng đoán.

## 2.2 Phần 2 : Xây dựng trang web với PyScript

### 2.2.1 : Giới thiệu chung về PyScript.

* **PyScript** là một framework được xây dựng bằng Pyodide (một trình thông dịch CPython được biên dịch bằng Emscripten thành WebAssembly), cho phép chạy mã Python trong trình duyệt.
* Trong file [README.md](https://github.com/pyscript/pyscript) của PyScript mô tả PyScript là một siêu dự án nhằm mục đích kết hợp nhiều công nghệ mở vào một framework cho phép người dùng tạo ra các ứng dụng trình duyệt phức tạp bằng Python. Nó tích hợp hoàn hảo với cách thức hoạt động của DOM trong trình duyệt và cho phép người dùng thêm logic Python một cách tự nhiên.

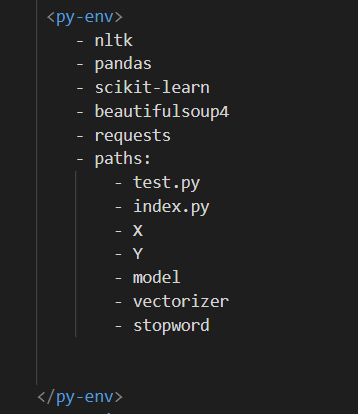
***(DOM (Document Object Model)*** *là một chuẩn được định nghĩa bởi W3C - Tổ chức Web toàn cầu được dùng để truy xuất và thao tác trên các tài liệu có cấu trúc dạng HTML hay XML bằng các ngôn ngữ lập trình thông dụng****)***

* Một trong những mục tiêu của PyScript là làm cho Web trở thành một nơi thân thiện cho bất kỳ ai muốn học cách viết mã. Framework này đã đạt được mục tiêu đó bằng cách không yêu cầu bất kỳ quy trình cài đặt nào ngoài trình soạn thảo văn bản và trình duyệt hiện tại.
* **Cách sử dụng PyScript:**
  + Để sử dụng PyScript thì cách đơn giản nhất là thêm đoạn code sau vào thẻ <head> trong file HTML :

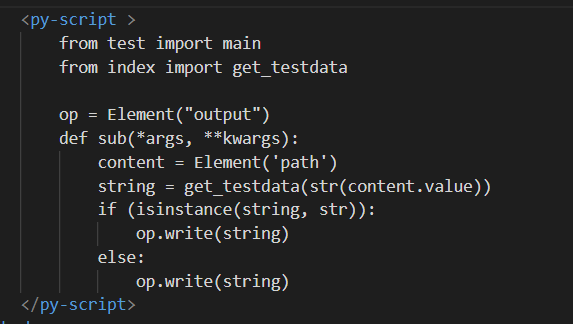
<link rel="stylesheet" href="https://pyscript.net/alpha/pyscript.css" />

<script defer src="<https://pyscript.net/alpha/pyscript.js>"></script>

* PyScript cho phép bạn viết Python trong HTML bằng cách sử dụng 3 thành phần chính sau đây :
  + ***py-env*** : xác định các package Python cần thiết để chạy đoạn mã Python.



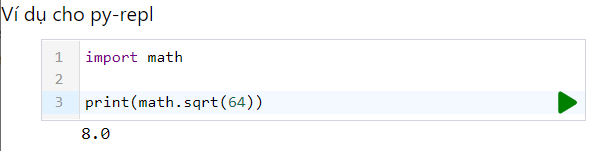
* *paths*: tìm nạp một mô-đun tùy chỉnh vào trình duyệt web.
  + ***py-script*** : là nơi viết mã Python được thực thi trong trang web.



* + ***py-repl***: tạo thành phần REPL (vòng lặp đọc - đánh giá - in) để đánh giá mã người dùng nhập và hiển thị kết quả.

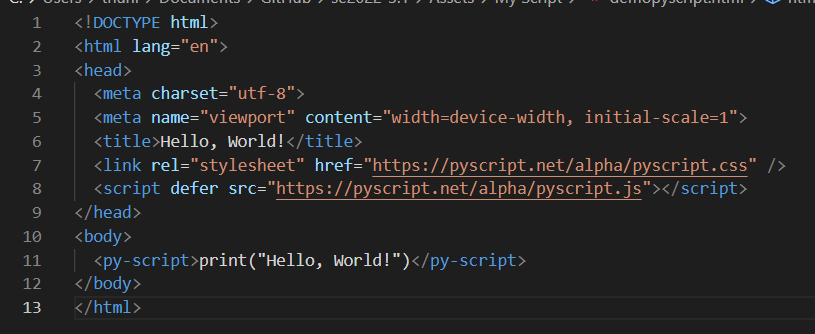


Và màn hình sẽ xuất hiện một repo tương tác có thể kiểm tra câu lệnh nhập vào như dưới đây:



### 2.2.2: Chương trình đầu tiên với PyScript: Hello World

Như đã giới thiệu ở trên, cách nhanh nhất để bắt đầu với PyScript là tạo một tệp HTML và thêm vào đó đoạn code bắt buộc vào thẻ <head>



*Dòng 7, 8 chính là 2 dòng mã cần thiết để có thể bắt đầu với PyScript.*

*Dòng 11 : thẻ <py-script> chứa dòng lệnh Python, chẳng hạn như là hàm print()*

* **Tìm nạp thời gian chạy Python từ Internet**
* Khi mở tệp HTML trong trình duyệt web, sẽ mất thời gian để tải trước khi hiển thị ra dòng chữ “*Hello, World!*”.
* PyScript phải tìm nạp hàng chục tải nguyên cho các dự án nguồn mở, bao gồm cả thời gian chạy Pyodide.
* Tuy nhiên, trình duyệt sẽ lưu trữ hầu hết tài nguyên đó trong bộ nhớ hoặc trên đĩa để trong tương lai, thời gian tải sẽ nhanh hơn đáng kể. Bạn cũng có thể làm việc ngoại tuyến mà không phụ thuộc vào kết nối Internet của mình miễn là bạn đã mở tệp HTML của mình ít nhất một lần.

Một cách tốt hơn là chúng ta nên trích xuất mã Python của mình thành một tệp riêng biệt. Để làm được điều này chỉ cần chỉ định thuộc tính src trong thẻ <py-script>. Ví dụ bạn có một file *hello.py.* Việc bạn cần làm đơn giản là chỉ định *src = “./hello.py”* :

<py-script src = “./hello.py”></py-script>

Điều này sẽ tải và ngay lập tức chạy tập lệnh Python ngay khi trang sẵn sàng.

### 2.2.3: Sử dụng PyScript để tìm và thao tác các phần tử HTML

WebAssugging hiện không cho phép tương tác trực tiếp với DOM mà cung cấp cho bạn quyền truy cập vào các thành phần cơ bản của HTML. Do đó, phải sử dụng JavaScript để truy vấn và duyệt qua cây DOM. Có thể truy cập các đối tượng JavaScript trong Python nhờ các đối tượng proxy của Pyodide, tạo điều kiện thuận lợi cho việc dịch giữa hai ngôn ngữ.

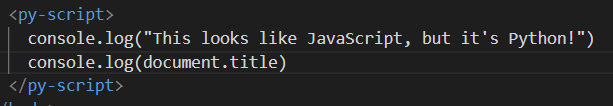
#### **Bộ điều hợp Apdapter của PyScript dành cho JavaScript Proxy**

* Đối với trường hợp sử dụng thông thường, có thể sử dụng lớp Element của PyScrupt. Lớp này cho phép bạn tìm các phần tử HTML chỉ bằng ID và sửa đổi nội dung của chúng ở một mức độ hạn chế.
* Lớp Element có các thuộc tính và phương thức khác như:

|  |  |
| --- | --- |
| .element | Một đối tượng proxy cho phần tử HTML của JavaScriptJavaScript |
| .id | Giá trị chuỗi của thuộc tính id của phần tử HTML nếu nó tồn tại |
| .value | Giá trị chuỗi của thuộc tính value của phần tử HTML nếu nó tồn tại |
| .add\_class() | Thêm một hoặc nhiều lớp CSS |
| .remove\_class() | Xóa một hoặc nhiều lớp CSS |
| .write() | Cập nhật thuộc tính .innerHTML hoặc thêm một phần tử <div> hoàn toàn mới |
| .clear() | Xóa thuộc tính value hoặc innerHTML |
| .clone() | Sao chép phần tử và chèn deep copy của nó vào cây DOM |
| .select() | Tìm phần tử con bằng bộ chọn CSS |

#### **Proxy JavaScript của Pyodide**

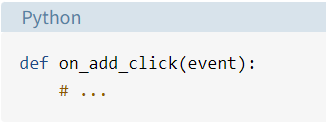
PyScript đã nhập một vài thứ từ mô-đun js vào global namespace trong Python. Cụ thể là các đối tượng ***console*** và ***document*** đã có sẵn :



#### **Python Event Callback Proxy**

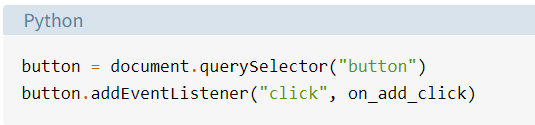
Làm cách nào để xử lý được các sự kiện DOM, chẳng hạn như nhấp chuột trong PySript? Câu trả lời là cần có chức năng callback.

* Đầu tiên, cần đóng gói mã hiện có của mình trong một hàm lấy sự kiện DOM làm đối số

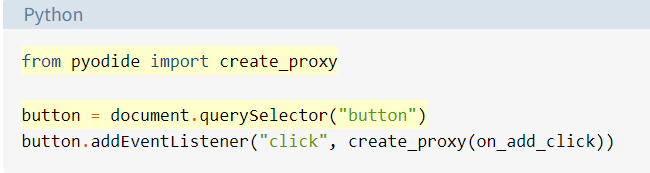


Đối số event sẽ không được sử dụng và bỏ qua nhưng không sao cả.

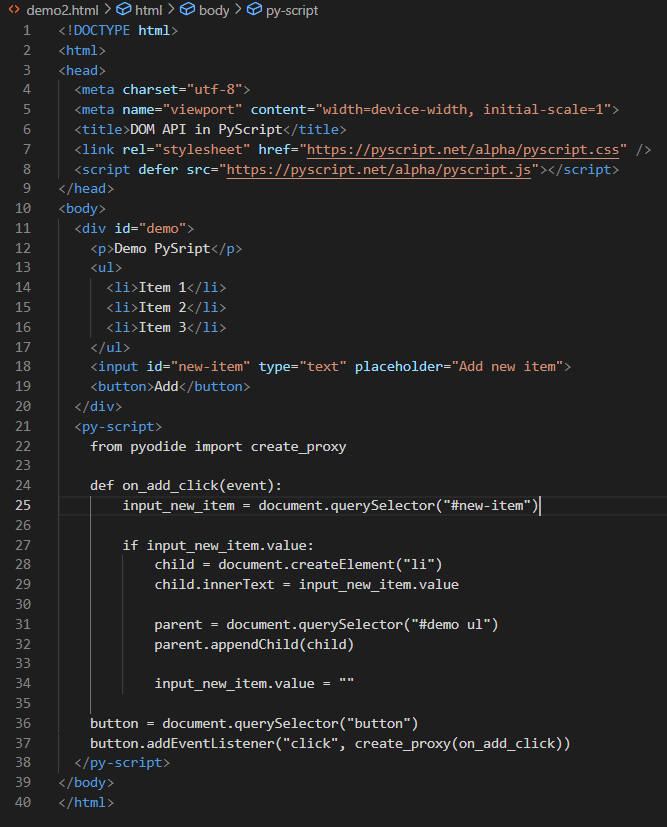
* Giả sử bạn đã có chức năng của mình, đã đến lúc báo cho trình duyệt biết khi nào nên gọi nó. Bạn làm điều đó bằng cách thêm một trình **event listener** vào một phần tử sẽ kích hoạt một hành động. Trong trường hợp này, phần tử đó là nút mà bạn có thể tìm thấy bằng bộ chọn CSS thích hợp và tên sự kiện là "click":



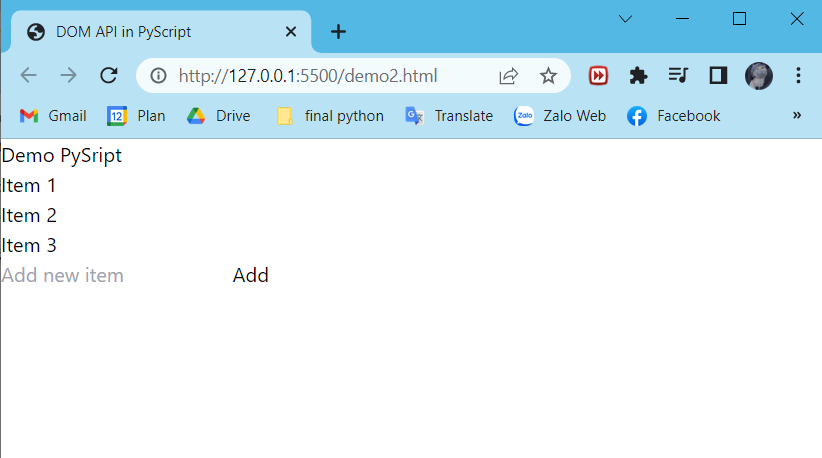
* Trình duyệt web mong đợi một cuộc gọi lại JavaScript, nhưng bạn cung cấp cho nó một chức năng Python. Để khắc phục điều này, bạn có thể tạo một đối tượng proxy, gọi được bằng cách sử dụng một hàm có liên quan từ mô-đun pyodide:



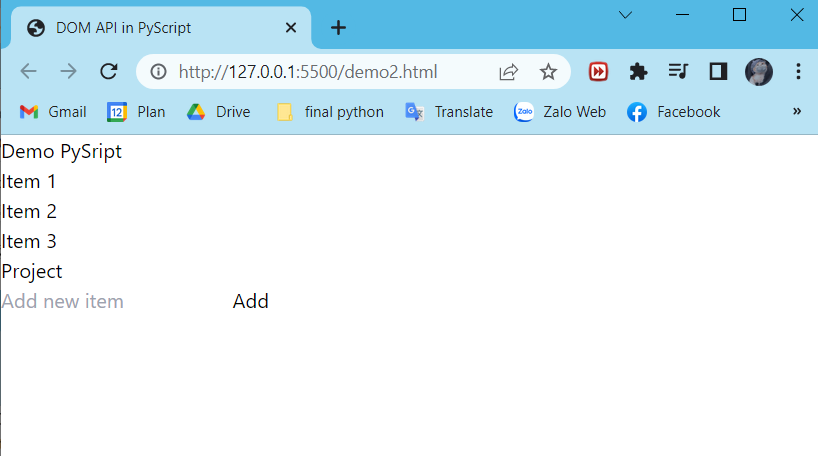
Dưới đây là chương trình đầy đủ, một ví dụ về tương tác đơn giản với PyScript:



Kết quả hiển thị:



Khi nhập “Project” vào ô “Add new item” và nhấp chuột vào Add. Ta được:



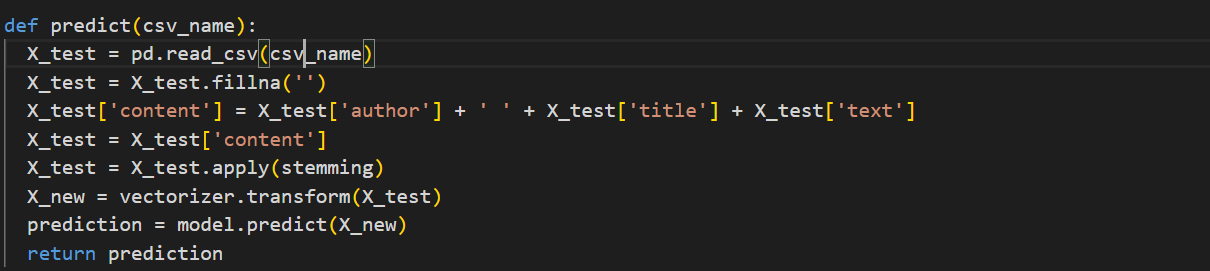
Như vậy có thể thấythấy PyScript chắc chắn là một công nghệ mới thú vị cho phép bạn chạy mã Python trong trình duyệt web nhờ Pyodide và WebAssembly.

## 2.3 Phần 3: Một số phương pháp khác để lấy dữ liệu cần dự đoán:

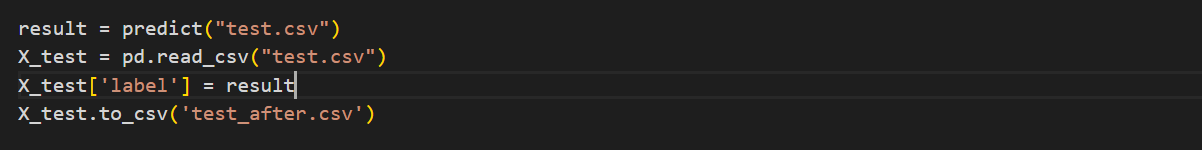
### 2.3.1 Xử lý dữ liệu hàng loạt từ các tệp .csv, tương tự như cách tập training được nhận vào, xem tệp predict\_csv.py:

Tệp test.csv được lấy từ <https://www.kaggle.com/c/fake-news/data?select=test.csv>

Hoàn toàn như cách ta đã lấy dữ liệu từ train.cvs, ta sẽ lấy dữ liệu từ test.cvs một cách tương tự và đưa ra dự đoán như đã làm với text:



Tuy nhiên kết quả trả ra sẽ là một numpy.ndarray - định dạng return ban đầu của hàm predict với dãy các số 0, 1 đại diện cho tính chất thật/giả của từng tin tức được truyền vào.



X\_test đang có kiểu là pandas.core.frame.DataFrame nên chúng ta có thể dễ dàng dùng hàm DataFrame.to\_csv để ghi lại kết quả. Cấu trúc chi tiết của hàm có thể xem tại [pandas.DataFrame.to\_csv — pandas 1.5.2 documentation](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html#pandas.DataFrame.to_csv). Nhưng chúng ta chỉ sử dụng cấu trúc đơn giản nhất đó là cung cấp đường dẫn cho file được tạo thành.

### 2.3.2. Crawl dữ liệu web

#### 2.3.2.1. Các thư viện sử dụng

- [Requests](https://requests.readthedocs.io/en/latest/) là một module Python mà bạn có thể sử dụng để gửi tất cả các loại yêu cầu HTTP. Đây là một thư viện dễ sử dụng với nhiều tính năng khác nhau, từ việc truyền các tham số trong URL cho đến gửi các header tùy biến và xác minh SSL.

- [Beautiful Soup 4 (bs4)](https://pypi.org/project/beautifulsoup4/) là một thư viện Python dùng để lấy dữ liệu từ các file HTML và XML. Nó hoạt động cùng với các parser - trình phân tích cú pháp, cung cấp cho bạn các cách để điều hướng, tìm kiếm và chỉnh sửa trong parse tree - cây phân tích được tạo từ parser.

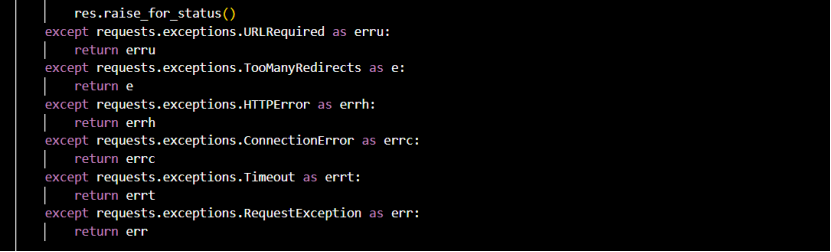
#### 2.3.2.2. Thực hiện

- Để sử dụng được module thì bạn cần phải import nó vào đầu chương trình:

- Gửi yêu cầu đến một URL được chỉ định bằng requests:

- Phương thức requests.get(path) giúp chúng ta gửi yêu cầu đến một trang web với path là đường dẫn đến trang web đó. Khi tạo yêu cầu thành công thì dữ liệu trả về sẽ là một đối tượng Response là res. Bây giờ bạn có thể sử dụng res để lấy dữ liệu của trang web bạn chỉ định.

- Kiểm tra trạng thái của response khi gửi một yêu cầu.

- res.raise\_for\_status() trả về một đối tượng HTTPError nếu xảy ra lỗi trong quá trình này, ngược lại chương trình sẽ tiếp tục mà không có ngoại lệ nào được nêu ra.

+ RequestException - Có lỗi gì đó xảy ra trong quá trình thực thi request của bạn.

+ ConnectionError - Kết nối xảy ra lỗi.

+ HTTPError - Có lỗi HTTP xảy ra.

+ URLRequired - URL của bạn không chính xác.

+ TooManyRedirects - Request redirect quá nhiều lần.

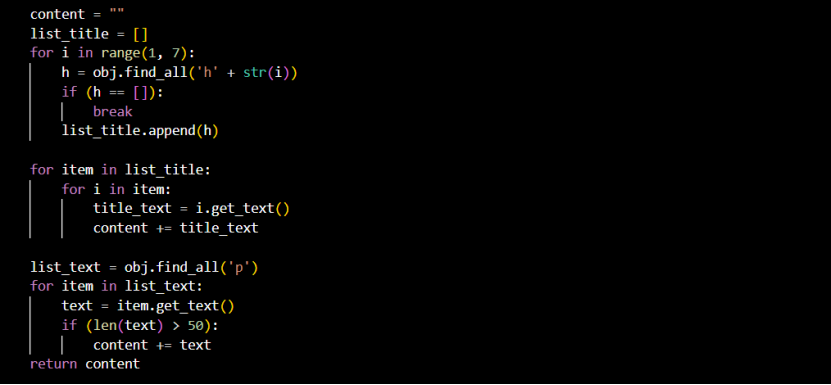
+ Timeout - Request bị timeout.

- Phân tích cú pháp HTML :

 + res.text cho phép bạn lấy dữ liệu HTML của trang web.

+ Hàm BeautifulSoup(res.text, ‘html.parser’) truyền vào dữ liệu HTML và sử dụng trình phân tích cú pháp html.parser, trả về đối tượng BeautifulSoup là obj.

- Lấy nội dung trang web bao gồm tiêu đề và nội dung bài báo:

 + Thực hiện vòng lặp for để lấy tất cả các thẻ <h1> - <h6> có thể có trong obj. Phương thức obj.find\_all(‘h’ + str(i)) cho phép tìm kiếm tất cả các thẻ <hi> trong obj, phương thức trả về một danh sách chứa tất cả kết quả. Sau đó cho kết quả vào một danh sách các tiêu đề list\_title.

+ Lấy ra nội dung trong từng thẻ bằng phương thức get\_text(), và lưu vào biến content.

+ Tiếp tục dùng phương thức obj.find\_all(‘p’) để tìm kiếm tất cả các thẻ <p> trong obj, kết quả lưu vào trong một danh sách list\_text. Thực hiện vòng lặp for và phương thức get\_text() để lấy nội dung của các thẻ <p>, lưu ý chỉ lấy nội dung của những thẻ có độ dài lớn hơn 50 và cho vào biến content.

#### 2.3.2.3. Module requests trong Pyscript

- Module requests không hoạt động trong Pyscript vì module requests sử dụng TCP socket của hệ điều hành. Tuy nhiên trình duyệt web không hỗ trợ TCP socket và nó không thể truy cập đến các module mạng cấp hệ điều hành vì bảo mật. Nên đây là một hạn chế bảo mật của trình duyệt web, không phải hạn chế của Pyscript.

# 3. Tổng kết:

Qua bản báo cáo này, chúng tôi đã trình bày một cách đơn giản để xử lý bài toán để dự đoán tin tức thật giả với Python. Thuật toán mà báo cáo này sử dụng là phân tích thành TF-IDF và Logic Regression (Hồi quy tuyến tính) để xử lý. Ngoài thuật toán này, với bài toán này ta hoàn toàn có thể sử dụng Passive Aggressive Classiifier, Bert For Sequence Classification,... Báo cáo còn cung cấp một số thông tin về PyScript, một framework còn rất mới mẻ (được công bố vào cuối tháng 4/2022) nhưng có tiềm năng cực lớn trong lĩnh vực xây dựng web. Ngoài ra, báo cáo còn sử dụng một số phương pháp khác để lấy input cho quá trình dự đoán.

# 4. Tài liệu tham khảo:

* Nguồn từ trang web:
* <https://docs.python.org/3/library/>
* <https://pandas.pydata.org/>
* <https://www.nltk.org/>
* <https://scikit-learn.org/>
* <https://pyscript.net/>
* <https://realpython.com/pyscript-python-in-browser/>
* <https://pypi.org>
* Các tài liệu khác:

+ Tripepi G, Jager KJ, Stel VS, Dekker FW, Zoccali C. How to Deal with Continuous and Dichotomic Outcomes in Epidemiological Research: Linear and Logistic Regression Analyses. Nephron Clin Pract. 2011 Feb 23;118(4):c399-c406

+ Rajaraman, A.; Ullman, J.D. (2011). "Data Mining" (PDF). Mining of Massive Datasets. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2.

+ Tài liệu tham khảo môn Lập trình nâng cao của ***PGS.TS. Lê Trọng Vĩnh - Đại học Khoa học tự nhiên - Đại học Quốc gia Hà Nội.***