TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN DATA MINING**

**FINAL PROJECT**

Người hướng dẫn: **TS LÊ CUNG TƯỞNG**

Người thực hiện: **TRƯƠNG ĐÌNH ÁNH - 51702062**

Lớp **: 17050201**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN DATA MINING**

**FINAL PROJECT**

Người hướng dẫn: **TS LÊ CUNG TƯỞNG**

Người thực hiện: **TRƯƠNG ĐÌNH ÁNH - 51702062**

Lớp **: 17050201**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

LỜI CẢM ƠN

Cảm ơn mọi người đã giúp đỡ tôi hoàn thành đồ án này.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của TS Lê Cung Tưởng;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 15 tháng 5 năm 2021*

*Tác giả*

*Trương Đình Ánh*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Task 1 sử dụng Selenium WebDriver để cào dữ liệu và dán nhãn tay.

Task 2 sử dụng File có sẵn để train và test theo 3 model LinearSVC, SVC, RandomForest. Và đánh giá các điểm Acc, Precision, Recall, F1 – score.

Task 3 Sử dụng model ở task 2 để test file ở task 1 và đánh giá các điểm Acc, Precision, Recall, F1 – score

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc72076465)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc72076466)

[TÓM TẮT iv](#_Toc72076467)

[MỤC LỤC 1](#_Toc72076468)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 2](#_Toc72076469)

[CHƯƠNG 1 – COLLECTING DATA 4](#_Toc72076470)

[1.1 Selenium 4](#_Toc72076471)

[1.2 Cào dữ liệu 6](#_Toc72076472)

[1.3 Ghi file CSV 6](#_Toc72076473)

[1.4 Đọc file csv 6](#_Toc72076474)

[CHƯƠNG 2 – EMOTION RECOGNITION FOR VIETNAMESE SOCIAL MEDIA TEXT 9](#_Toc72076475)

[1.1 Load data Train 9](#_Toc72076476)

[1.2 Xử lý tiền dữ liệu 10](#_Toc72076477)

[1.3 Vectorization 13](#_Toc72076478)

[1.4 Model và đánh giá 14](#_Toc72076479)

[CHƯƠNG 3 – APPLYING THE TRAINED MODEL IN TASK 2 TO THREE DATASETS IN TASK 1 16](#_Toc72076480)

[3.1 Load File Test: 16](#_Toc72076481)

[3.2 Model và đánh giá 16](#_Toc72076482)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1: Cài đặt bộ Selenium tools 4](#_Toc72076483)

[Hình 2: Module cần thiết 4](#_Toc72076484)

[Hình 3: Set up đường dẫn cho tool 5](#_Toc72076485)

[Hình 4: Set up đường dẫn video cho tool 5](#_Toc72076486)

[Hình 5: Giao diện của browser tự động 5](#_Toc72076487)

[Hình 6: Tự động cuộn 5](#_Toc72076488)

[Hình 7: Tự động lấy comment 6](#_Toc72076489)

[Hình 8: Tạo file csv và ghi vào file 6](#_Toc72076490)

[Hình 9: File csv sau khi kết thúc quá trình crawl data 6](#_Toc72076491)

[Hình 10: File csv hiển thị 7](#_Toc72076492)

[Hình 11: Set up định dạng cho file 7](#_Toc72076493)

[Hình 12: File hiển thị theo định dạng UTF-8 9](#_Toc72076494)

[Hình 13: Load file có sẵn để train 9](#_Toc72076495)

[Hình 14: Một số câu trong data train 9](#_Toc72076496)

[Hình 15: Data bị mất cân bằng 10](#_Toc72076497)

[Hình 16: Các bước làm đầy dữ liệu 11](#_Toc72076498)

[Hình 17: Data đã cân bằng 11](#_Toc72076499)

[Hình 18: Bộ từ diển các từ đúng 12](#_Toc72076500)

[Hình 19: Các bước xử lý dữ liệu một câu 13](#_Toc72076501)

[Hình 20: Data sau khi xử lý 13](#_Toc72076502)

[Hình 21: Function của 2-gram 13](#_Toc72076503)

[Hình 22: Khi áp dụng bi-gram 14](#_Toc72076504)

[Hình 23: DictVectorizer 14](#_Toc72076505)

[Hình 24: Model SVC có độ acc là 0.49 14](#_Toc72076506)

[Hình 25: Model LinearSVC có độ acc là 0.52 15](#_Toc72076507)

[Hình 26: Model RandomForest có độ acc là 0.47 15](#_Toc72076508)

[Hình 27: Biểu đồ thể hiện độ Acc của 3 Model 15](#_Toc72076509)

[Hình 28: File data crawl đã dán nhãn 16](#_Toc72076510)

[Hình 29: Load và gộp file 16](#_Toc72076511)

[Hình 30: Model SVC có độ acc là 0.42 17](#_Toc72076512)

[Hình 31: Model LinearSVC có độ acc là 0.38 17](#_Toc72076513)

[Hình 32: Model RandomForest có độ acc là 0.31 17](#_Toc72076514)

[Hình 33: Biểu đồ độ chính xác của 3 Model ở Task 3 18](#_Toc72076515)

CHƯƠNG 1 – COLLECTING DATA

1.1 Selenium

Selenium là một bộ công cụ kiểm thử tự động dành cho các web application trên các trình duyệt và các nền tảng khác nhau. Nó có 4 công cụ khác nhau:

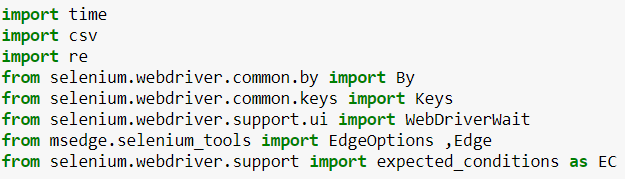
* Selenium Integrated Development Environment (IDE)
* Selenium Remote Control (RC)
* Selenium WebDriver (WD)
* Selenium Grid

Trong đó ta dùng Selenium WebDriver để giải quyết bài toán cào bình luận từ trang Youtube. Ta sẽ cài đặt 2 module này về:



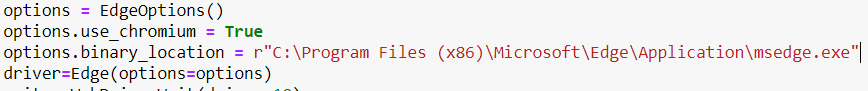
Hình 1: Cài đặt bộ Selenium tools

Sau đó ta sẽ import các module cần thiết để giải quyết bài toán. Trong đó có csv tạo và ghi file CSV, re để xử lý dữ liệu text, Edge và EdgeOptions để chọn browser là MS Edge, và các module khác dùng để lấy comment tự động.



Hình 2: Module cần thiết

Bởi vì ta dùng trình duyệt MS Edge nên chúng ta sẽ phải set up đường dẫn đến file “msedge.exe” để nó tự động mở MS Egde lên.



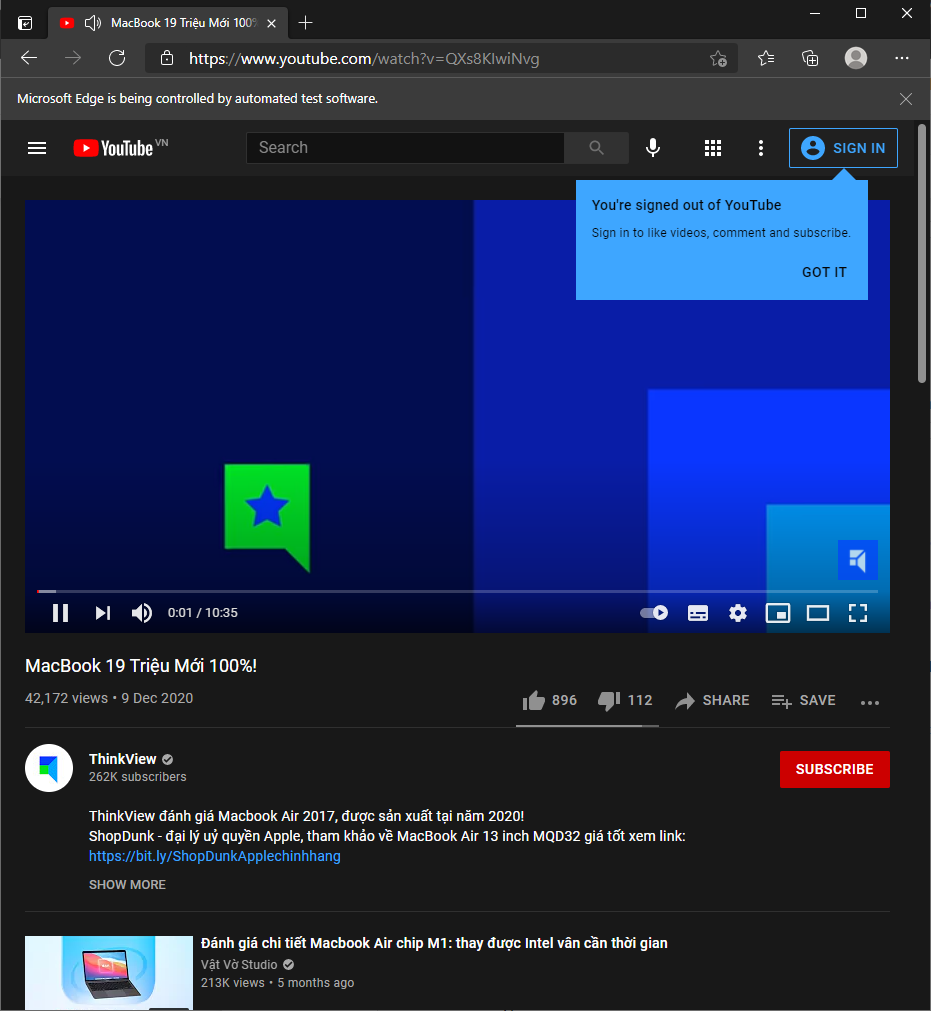
Hình 3: Set up đường dẫn cho tool

Sau đó ta sẽ gán đường link video mà ta muốn cào comment. Và chỉnh thời gian cho nó load dữ liệu



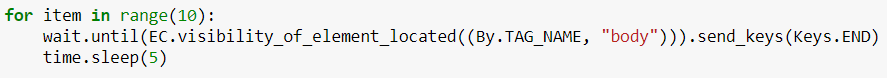
Hình 4: Set up đường dẫn video cho tool

Khi đó, giao diện của browser sẽ mở lên.



Hình 5: Giao diện của browser tự động

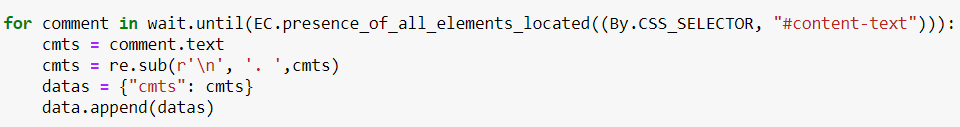
Tuy nhiên ta phải cho nó tự động cuộn thì mới gọi là Auto tool.



Hình 6: Tự động cuộn

1.2 Cào dữ liệu

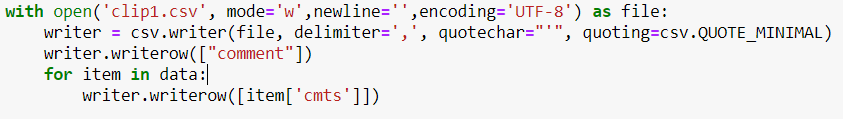
Sau 10 lần cuộn, thì sẽ bắt đầu lấy tất cả dữ liệu sau 10 lần load đó. Bỏ đi ký tự “/n” và sau đó đưa về một List



Hình 7: Tự động lấy comment

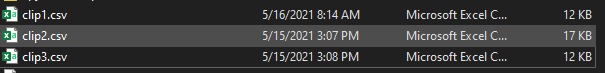
1.3 Ghi file CSV

Khi có dữ liệu đã cào rồi, ta bắt đầu tạo file csv và định dạng là UTF-8, và ghi từng dòng vào file



Hình 8: Tạo file csv và ghi vào file

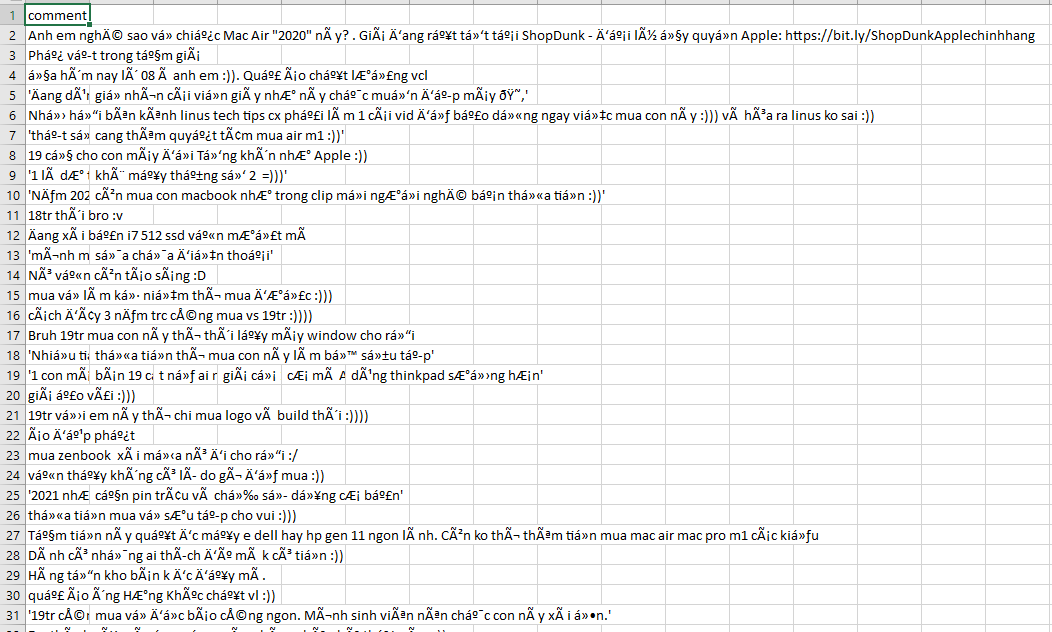
Tương tự, cho 2 video cũng làm theo quy trình như trên. Khi quá trình ghi vào file csv đã kết thúc thì chúng ta sẽ có 3 file csv



Hình : File csv sau khi kết thúc quá trình crawl data

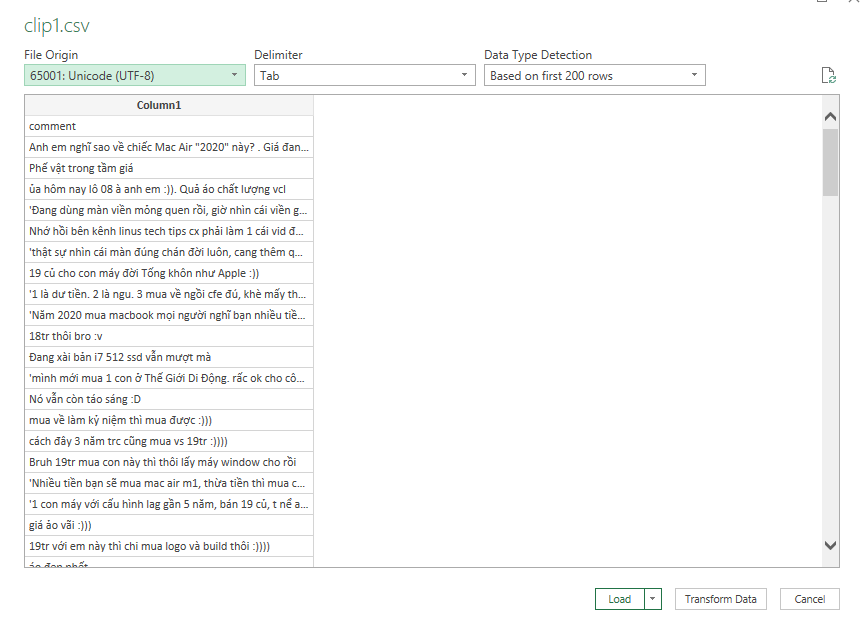
1.4 Đọc file csv

File csv khi mở bằng excel thì sẽ không định dạng UTF-8 nên ta sẽ không thể nào có thể đọc được để dán nhãn



Hình 10: File csv hiển thị

Nên ta sẽ phải load file lại theo định dạng UTF-8 và lưu lại theo đuôi \*.xlsx



Hình 11: Set up định dạng cho file

Sau đó file excel đã hiển thị theo đúng định dạng mà ta có thể đọc và gán nhãn.

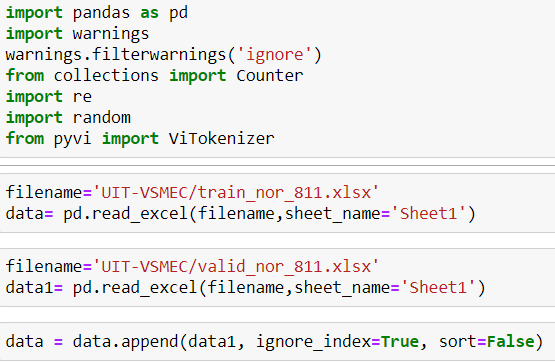


Hình 12: File hiển thị theo định dạng UTF-8

CHƯƠNG 2 – EMOTION RECOGNITION FOR VIETNAMESE SOCIAL MEDIA TEXT

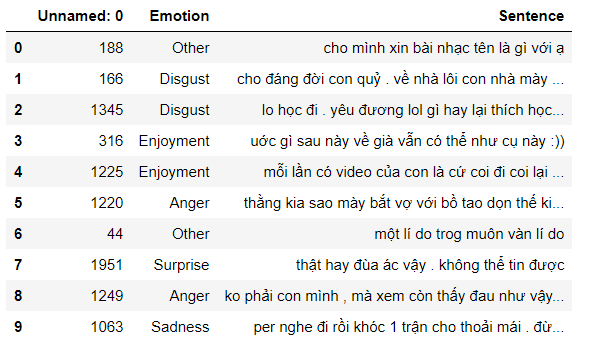
1.1 Load data Train

Ta sẽ phải import một số thư viện cần thiết cho việc load file có sẵn và gộp lại thành 1 file



Hình : Load file có sẵn để train

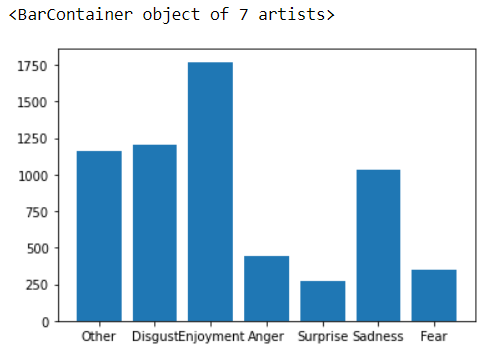
Khi đó data train của ta sẽ có dạng như thế này.



Hình : Một số câu trong data train

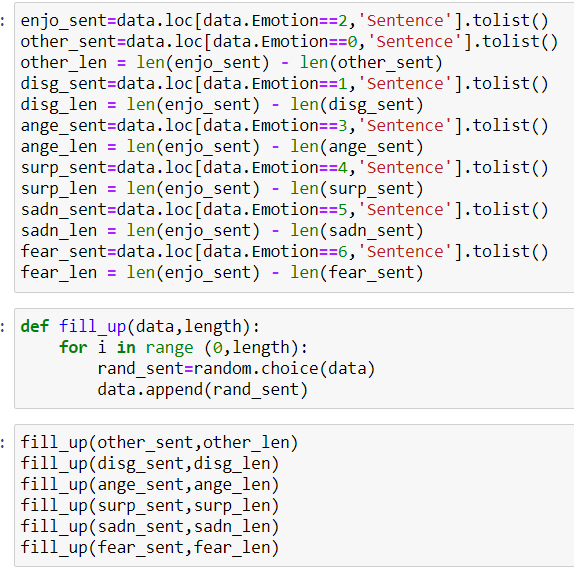
1.2 Xử lý tiền dữ liệu

Ở bước tiền xử lý dữ liệu, ta sẽ xóa bỏ những cột không cần thiết, in ra biểu label và ta nhận ra đây là một bộ data bị mất cân bằng. Và dữ liệu cao nhất là Enjoyment

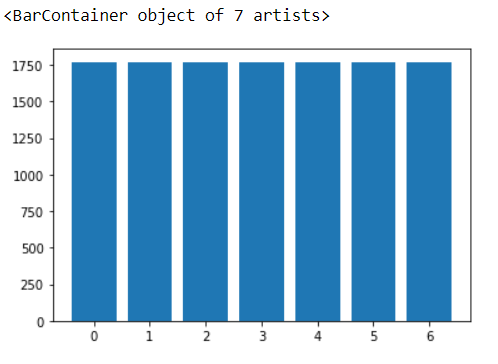


Hình 15: Data bị mất cân bằng

Ta sẽ fill up cho các cột dữ liệu bằng nhau hết bằng cách lấy ra ngẫu nhiên 1 số câu của chính nó để thêm vào nó.



Hình 16: Các bước làm đầy dữ liệu



Hình 17: Data đã cân bằng

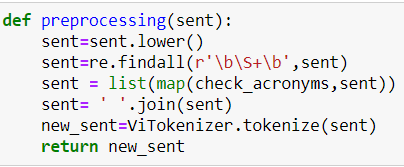
Tiếp đến, ta thấy trong data có rất nhiều chữ viết tắt và từ bị sai chính tả. Nên ta tạo một bộ lọc từ dựa trên data đã có sẵn. Để một lát nữa ta sẽ lọc các từ đó đi thay các từ đúng cú pháp, đúng chính tả



Hình 18: Bộ từ diển các từ đúng

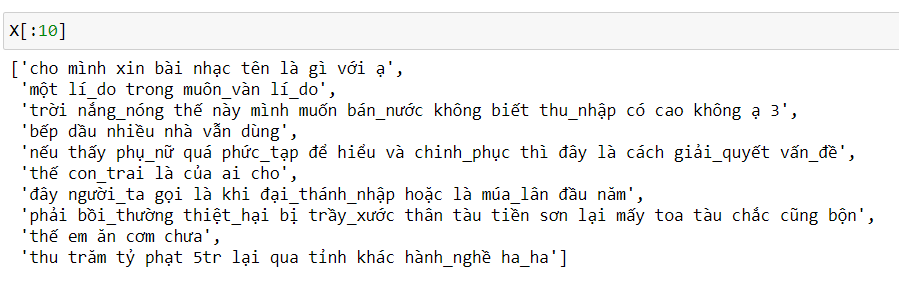
Khi truyền một câu vào ta sẽ xử theo các bước sau:

* B1: ta sẽ chuyển về thành chữ thường
* B2: Sau đó chỉ lấy những chữ cái và số
* B3: Sau đó chúng ta sẽ dùng bộ từ điển ở trên để chuyển các chữ viết tắt thành các từ đúng cú pháp
* B4: Sau đó ta phân từ vựng theo tiếng việt



Hình 19: Các bước xử lý dữ liệu một câu

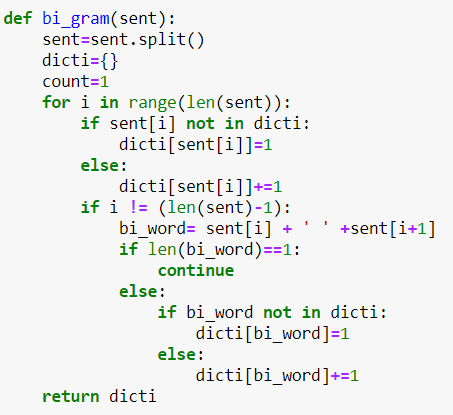
Tập data sau khi xử lý xong sẽ có dạng như thế này.



Hình 20: Data sau khi xử lý

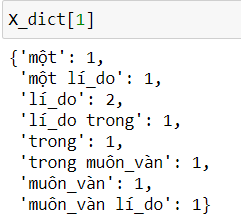
1.3 Vectorization

Vì ta sử dụng DictVectorizer nên là input của ta phải là List chứa số lần lặp lại của các từ trong câu. Nên ta ứng dụng 2-gram vào để tạo ra input đó.



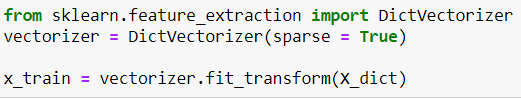
Hình 21: Function của 2-gram

Input của một câu sẽ có dạng thành Dictionary chứa từ và số lần lặp của nó. Với tập dữ liệu từ ta sẽ có 1 List



Hình 22: Khi áp dụng bi-gram

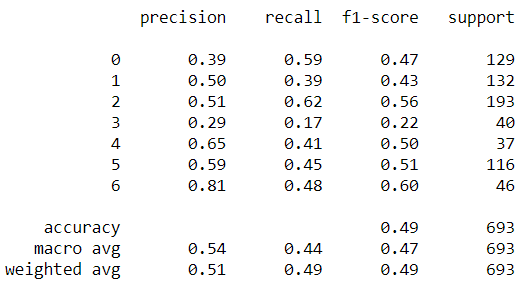
Sau đó ta truyền vào module DictVectorizer để fit và transform



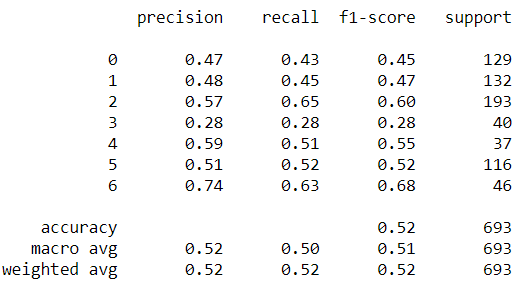
Hình 23: DictVectorizer

1.4 Model và đánh giá

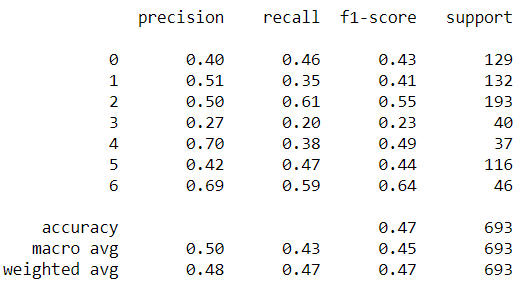
Sau khi load data test vào ta cũng sẽ transform thành Vector và tiến hành test xem độ chính xác của 3 model là LinearSVC, SVC, RandomForest



Hình : Model SVC có độ acc là 0.49

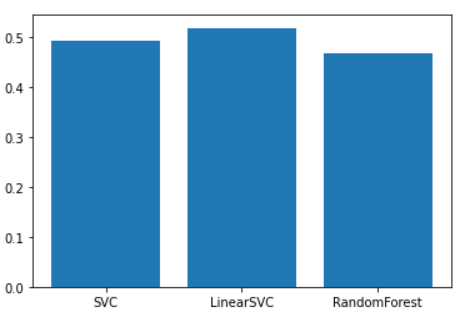


Hình : Model LinearSVC có độ acc là 0.52



Hình : Model RandomForest có độ acc là 0.47

Sau đây là biểu đồ thể hiện độ Accuracy của 3 Model

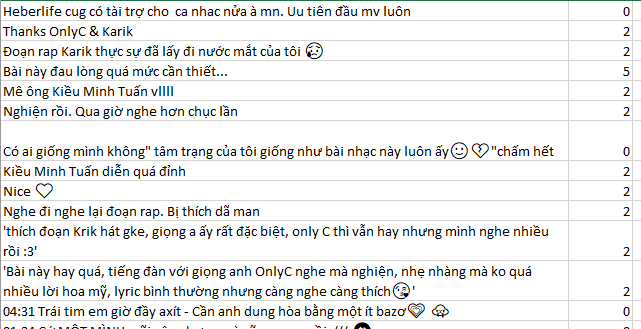


Hình : Biểu đồ thể hiện độ Acc của 3 Model

CHƯƠNG 3 – APPLYING THE TRAINED MODEL IN TASK 2 TO THREE DATASETS IN TASK 1

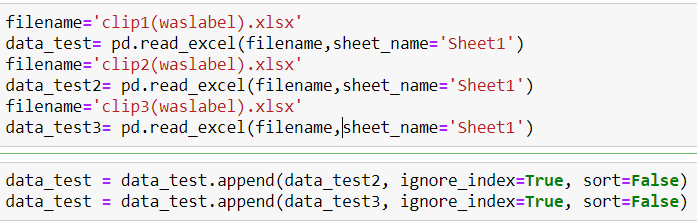
3.1 Load File Test:

File data mà ta đã crawl từ Task 1 sau khi dán nhãn sẽ có dạng



Hình 28: File data crawl đã dán nhãn

Tiếp đến là load 3 file lại và gộp lại thành 1

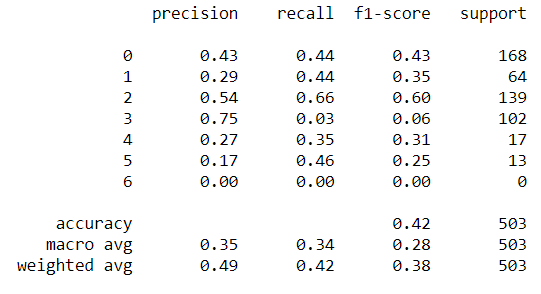


Hình 29: Load và gộp file

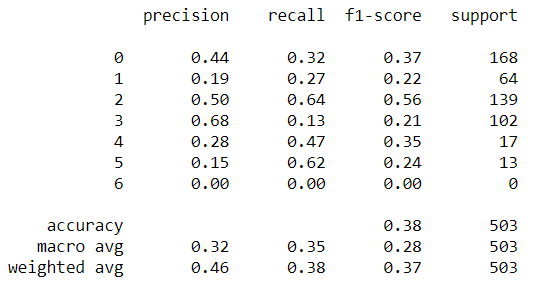
Data test của ta sẽ có được xử lý tiền dữ liệu và transform tương tự file test ở Task 2.

3.2 Model và đánh giá

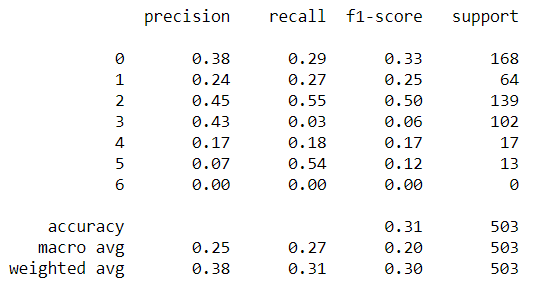
Ta cũng áp dụng 3 Model của Task 2 là LinearSVC, SVC, Randomforest



Hình 30: Model SVC có độ acc là 0.42

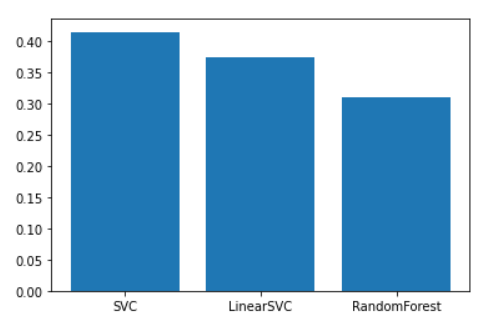


Hình 31: Model LinearSVC có độ acc là 0.38



Hình 32: Model RandomForest có độ acc là 0.31

Ta thấy rằng tuy ở Task 2 độ chính xác của LinearSVC là cao nhất nhưng đến Task 3 có sự thay đổi thành SVC là cao nhất. Còn Randomforest thì vẫn thấp nhất.



Hình 33: Biểu đồ độ chính xác của 3 Model ở Task 3