TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ**

**NHẬP MÔN**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

*Người hướng dẫn*: **PGS.TS. LÊ ANH CƯỜNG**

*Người thực hiện*: **NGUYỄN THỊ CẨM THÙY – 52100844**

**Đinh Phương My – 52100703**

**NGUYỄN PHÚC TRỌNG – 52100857**

*Lớp:* **21050201**

**21050401**

*Khoá:* **25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ**

**NHẬP MÔN**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

*Người hướng dẫn*: **PGS.TS. LÊ ANH CƯỜNG**

*Người thực hiện*: **NGUYỄN THỊ CẨM THÙY – 52100844**

**Đinh Phương My – 52100703**

**NGUYỄN PHÚC TRỌNG – 52100857**

*Lớp:* **21050201**

**21050401**

*Khoá:* **25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

LỜI CẢM ƠN

Lời đầu tiên, nhóm 10 xin gửi lời cảm ơn đến khoa Công nghệ thông tin đã tạo ra môn học này cho nhóm 10 có thể học tập cùng nhau và thực hành môn xử lý ngôn ngữ tự nhiên. Nhóm 10 xin chân thành cảm ơn PSG.TS.Lê Anh Cường đã giảng dạy, truyền đạt lại kiến thức một cách rất nhiệt tình và tạo ra bầu không khí học tập thoải mái cho nhóm 10 học tập tốt.

Nhóm 10 biết bài báo cáo này còn nhiều điều còn thiếu sót, nhóm 10 mong thầy có thể góp ý cho nhóm em.

Nhóm 10 xin chân thành cảm ơn!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Nhóm 10 xin cam đoan đây là sản phẩm đồ án của riêng nhóm và được sự hướng dẫn của PGS.TS.Lê Anh Cường. 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 nhóm 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 nhóm gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 25 tháng 03 năm 2024*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

**

 *Nguyễn Thị Cẩm Thùy*

* Đinh Phương My*

*Nguyễn Phúc Trọng*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

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

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

**Phần đánh giá của GV chấm bài**

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

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

TÓM TẮT

Báo cáo này được chia thành 3 chương, mỗi chương tập trung vào một phần quan trọng của nghiên cứu. Dưới đây là một mô tả nhanh về từng chương:

Chương 1. Tổng quan về Word2Vec, bao gồm các phương pháp CBOW và Skip-gram, được sử dụng để tạo ra các biểu diễn vector cho từ vựng trong văn bản. Chương này khám phá cách mà các phương pháp này hoạt động và cung cấp một cái nhìn tổng quan về vai trò quan trọng của Word2Vec trong xử lý ngôn ngữ tự nhiên và các ứng dụng của nó.

Chương 2. Tập trung vào việc xây dựng một mô hình phát hiện văn bản có thuộc các nhãn đã cho (như harassment, hate speech, sextually explicit, dangerous content) và so sánh hiệu suất của các mô hình phân loại văn bản khác nhau. Chương này đi sâu vào quá trình tiền xử lý dữ liệu, vector hóa văn bản và lựa chọn mô hình, cung cấp một cái nhìn rõ ràng về các phương pháp hiện đại nhất trong phân loại văn bản.

Chương 3. Đánh giá khả năng thực hiện của mô hình, bao gồm cả những thách thức và giới hạn của quá trình nghiên cứu. Chương này là nơi tóm tắt kết quả, đưa ra đề xuất cho các nghiên cứu tiếp theo và thảo luận về những khó khăn gặp phải trong quá trình nghiên cứu.

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc163262376)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc163262377)

[TÓM TẮT iv](#_Toc163262378)

[DANH MỤC CÁC CHỮ VIẾT TẮT 3](#_Toc163262379)

[DANH CHỨC NĂNG CÁC BẢNG BIỂU, HÌNH ẢNH 4](#_Toc163262380)

[CHƯƠNG 1 – MÔ HÌNH WORD2VEC 6](#_Toc163262381)

[1.1 Sơ lược về Word2Vec 6](#_Toc163262382)

[1.1.1 Mô hình CBOW 6](#_Toc163262383)

[1.1.1.1 Ý nghĩa 6](#_Toc163262384)

[1.1.1.2 Mô hình 6](#_Toc163262385)

[1.1.1.3 Huấn luyện mô hình 8](#_Toc163262386)

[1.1.2 Mô hình Skip-gram 9](#_Toc163262387)

[1.1.1.1 Ý nghĩa 9](#_Toc163262388)

[1.1.1.2 Mô hình 9](#_Toc163262389)

[1.1.1.3 Huấn luyện mô hình 10](#_Toc163262390)

[1.2 Ứng dụng vào bài toán 12](#_Toc163262391)

[CHƯƠNG 2 – MÔ HÌNH PHÂN LOẠI VĂN BẢN 16](#_Toc163262392)

[2.1 Tập dữ liệu 16](#_Toc163262393)

[2.2 Tiền xử lí dữ liệu 18](#_Toc163262394)

[2.3 Mô hình phân loại văn bản và độ đo 19](#_Toc163262395)

[2.3.1 Mô hình phân loại văn bản 19](#_Toc163262396)

[2.3.1.1 Logistic Regression 19](#_Toc163262397)

[2.3.1.2 Naive Bayes 20](#_Toc163262398)

[2.3.1.3 Random Forest 20](#_Toc163262399)

[2.3.1.4 Support Vector Machine 20](#_Toc163262400)

[2.3.1.5 Multilayer Perceptron 21](#_Toc163262401)

[2.3.2 Độ đo 23](#_Toc163262402)

[2.3.3 Áp dụng mô hình và so sánh kết quả 24](#_Toc163262403)

[2.4 Phân tích dự đoán nhãn 27](#_Toc163262404)

[2.5 Phân tích dữ liệu văn bản 28](#_Toc163262405)

[CHƯƠNG 3 – KẾT LUẬN 30](#_Toc163262406)

[3.1 Kết quả đạt được 30](#_Toc163262407)

[3.2 Thuận lời và khó khăn 30](#_Toc163262408)

[3.2.1 Thuận lợi 30](#_Toc163262409)

[3.2.2 Khó khăn 30](#_Toc163262410)

[3.3 Đánh giá 30](#_Toc163262411)

[TÀI LIỆU THAM KHẢO 31](#_Toc163262412)

DANH MỤC CÁC CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

CBOW Continuous Bag of Words.

LR Logistic Regression

MLP Multilayer Perceptron

NB Naive Bayes

NLP Natural Language Processing

SGD Stochastic Gradient Descent.

SVM Support Vector Machine

TF-IDF Term Frequency-Inverse Document

RF Random Forest

DANH CHỨC NĂNG CÁC BẢNG BIỂU, HÌNH ẢNH

**DANH CHỨC NĂNG BẢNG**

[Bảng 4.3 Đánh giá độ hoàn thành đồ án 15](#_Toc162996617)

**DANH CHỨC NĂNG HÌNH ẢNH**

[Hình 1.1.1.1 Mô hình CBOW dạng tổng quát 7](#_Toc163261407)

[Hình 1.1.1.2 Ví dụ minh họa theo mô hình CBOW 7](#_Toc163261408)

[Hình 1.1.2.1 Mô hình Skip-gram dạng tổng quát 9](#_Toc163261409)

[Hình 1.1.2.2 Ví dụ minh họa theo mô hình Skip-gram 10](#_Toc163261410)

[Hình 1.2.2.1 Code cài đặt các thư viện cần thiết 12](#_Toc163261411)

[Hình 1.2.2.2 Code hàm tính vector trung bình cho một câu 12](#_Toc163261412)

[Hình 1.2.2.3 Code hàm tìm câu tương tự nhất 13](#_Toc163261413)

[Hình 1.2.2.4 Code khởi tạo tập dữ liệu các câu S 14](#_Toc163261414)

[Hình 1.2.2.5 Code xây dựng mô hình theo CBOW và Skip-gram 14](#_Toc163261415)

[Hình 1.2.2.6 Code nhập câu X và kiểm tra kết quả 15](#_Toc163261416)

[Hình 1.2.2.7 Kết quả trả về 15](#_Toc163261417)

[Hình 2.1 Tập dữ liệu facebook\_comment\_2k7 17](#_Toc163261418)

[Hình 2.2.1 Code tải dữ liệu 18](#_Toc163261419)

[Hình 2.2.2 Code đọc dữ liệu 18](#_Toc163261420)

[Hình 2.2.3 Code tách nhãn chính cho từng dòng dữ liệu 18](#_Toc163261421)

[Hình 2.2.4 Kết quả so sánh tỉ lệ số lượng các nhãn 19](#_Toc163261422)

[Hình 2.3.3.1 Code chia dữ liệu thành tập huấn luyện 24](#_Toc163261423)

[Hình 2.3.3.2 Code vector hóa văn bản 24](#_Toc163261424)

[Hình 2.3.3.3 Code huấn luyện mô hình LR 24](#_Toc163261425)

[Hình 2.3.3.4 Code huấn luyện mô hình NB 24](#_Toc163261426)

[Hình 2.3.3.5 Code huấn luyện mô hình RF 25](#_Toc163261427)

[Hình 2.3.3.6 Code huấn luyện mô hình MLP 25](#_Toc163261428)

[Hình 2.3.3.7 Code huấn luyện mô hình SVM 25](#_Toc163261429)

[Hình 2.3.3.8 Code đánh giá 5 loại mô hình 25](#_Toc163261430)

[Hình 2.3.3.9 Kết quả đánh giá 5 loại mô hình 26](#_Toc163261431)

[Hình 2.3.3.10 Kết quả so sánh các metric 5 loại mô hình 26](#_Toc163261432)

[Hình 2.4.1 Code phân tích dự đoán nhãn 27](#_Toc163261433)

[Hình 2.4.2 Kết quả dự đoán nhãn 27](#_Toc163261434)

[Hình 2.5.1 Code phân tích dữ liệu văn bản 28](#_Toc163261435)

[Hình 2.5.1 Kết quả phân tích dữ liệu văn bản 29](#_Toc163261436)

CHƯƠNG 1 – MÔ HÌNH WORD2VEC

1.1 Sơ lược về Word2Vec

Word2Vec là một mạng neural 2 lớp với duy nhất 1 tầng ẩn, lấy đầu vào là một corpus lớn và sinh ra không gian vector (với số chiều khoảng vài trăm), với mỗi từ duy nhất trong corpus được gắn với một vector tương ứng trong không gian. Các word vectors được xác định trong không gian vector sao cho những từ có chung ngữ cảnh trong corpus được đặt gần nhau trong không gian. Dự đoán chính xác cao về ý nghĩa của một từ dựa trên những lần xuất hiện trước đây.

Word2Vec có hai phương pháp chính là CBOW và Skip-gram. CBOW cố gắng dự đoán từ vựng hiện tại dựa trên ngữ cảnh xung quanh nó, trong khi Skip-gram cố gắng dự đoán các từ vựng xung quanh từ một từ đích cho trước. Thuật toán CBOW tốn ít thời gian huấn luyện mô hình hơn Skip-gram. Tuy nhiên, Skip-gram có độ chính xác cao hơn và có chứa cả những từ ít xuất hiện.

Word2Vec sử dụng một mô hình mạng nơ-ron đơn giản để học các biểu diễn từ vựng. Quá trình huấn luyện được thực hiện thông qua việc tối ưu hóa hàm mất mát dựa trên dữ liệu văn bản đầu vào. Các thuật toán tối ưu hóa như SGD thường được sử dụng để điều chỉnh các trọng số của mạng nơ-ron.

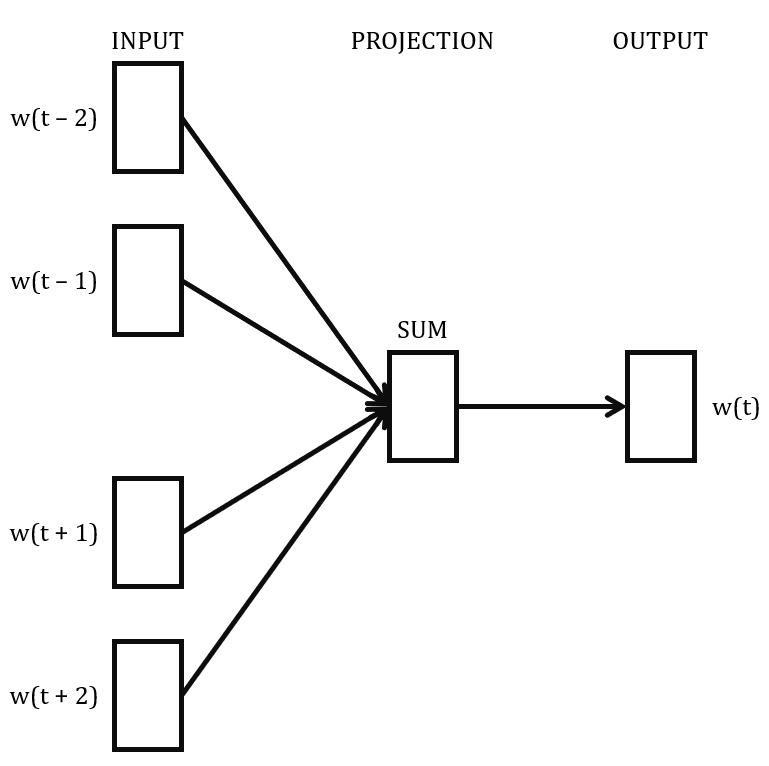
1.1.1 Mô hình CBOW

1.1.1.1 Ý nghĩa

Mô hình CBOW, sử dụng từ ngữ cảnh (từ xung quanh) để dự đoán từ đích (targetword). Mô hình này giả thiết vị trí của các từ ngữ cảnh không ảnh hưởng tới việc dự đoán từ target

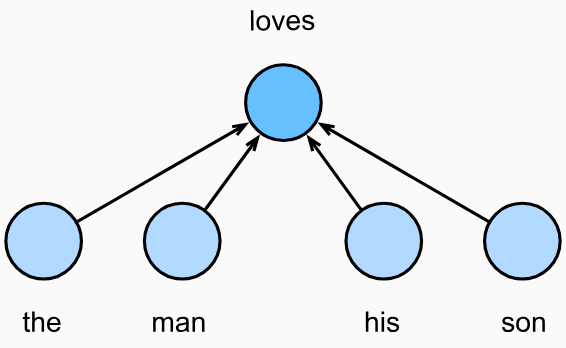
1.1.1.2 Mô hình

Trong CBOW, một mô hình neural network có một lớp ẩn và hai lớp đầu vào và đầu ra được sử dụng. Đầu vào là một cửa sổ các từ xung quanh từ cần dự đoán, còn đầu ra là từ cần dự đoán. Mô hình này cố gắng học các biểu diễn từ để dự đoán từ mục tiêu dựa trên ngữ cảnh.



Hình 1.1.1.1 Mô hình CBOW dạng tổng quát

Ví dụ, giả sử chuỗi văn bản là “the”, “man”, “loves”, “his” và “son”. Ta sử dụng “loves” làm từ đích trung tâm và đặt kích thước cửa sổ ngữ cảnh bằng 2. Mô hình CBOW quan tâm đến xác suất có điều kiện để sinh ra từ đích “love” dựa trên các từ ngữ cảnh “the”, “man”, “his” và “son” như sau:



Hình 1.1.1.2 Ví dụ minh họa theo mô hình CBOW

Vì có quá nhiều từ ngữ cảnh trong mô hình CBOW, ta sẽ lấy trung bình các vector từ của chúng và sau đó sử dụng phương pháp tương tự như trong mô hình skip-gram để tính xác suất có điều kiện.

Giả sử và là vector từ ngữ cảnh và vector từ đích trung tâm của từ có chỉ số i trong từ điển. Gọi c là chỉ số của từ đích trung tâm và *, … ,*  là chỉ số các từ ngữ cảnh *, … ,*  trong từ điển. Do đó, xác suất có điều kiện sinh ra từ đích trung tâm dựa vào các từ ngữ cảnh cho trước:

Để rút gọn, ký hiệu  *và* . Trong đó, tập chỉ số trong bộ từ vựng là . Phương trình trên được đơn giản hóa thành:

Cho một chuỗi văn bản có độ dài , ta giả định rằng từ xuất hiện tại bước thời gian là và kích thước của cửa sổ ngữ cảnh là . Hàm hợp lý của mô hình CBOW là xác suất sinh ra bất kỳ từ đích trung tâm nào dựa vào những từ ngữ cảnh.

1.1.1.3 Huấn luyện mô hình

Uớc lượng hợp lý cực đại của mô hình CBOW tương đương với việc cực tiểu hóa hàm mất mát:

Lưu ý rằng:

Thông qua phép đạo hàm, ta có thể tính log của xác suất có điều kiện của gradient của bất kỳ vector từ ngữ cảnh nào trong công thức trên.

Sau đó, sử dụng cùng phương pháp đó để tính gradient cho các vector của từ khác. Mô hình CBOW thường sử dụng vector từ ngữ cảnh làm vector biểu diễn một từ.

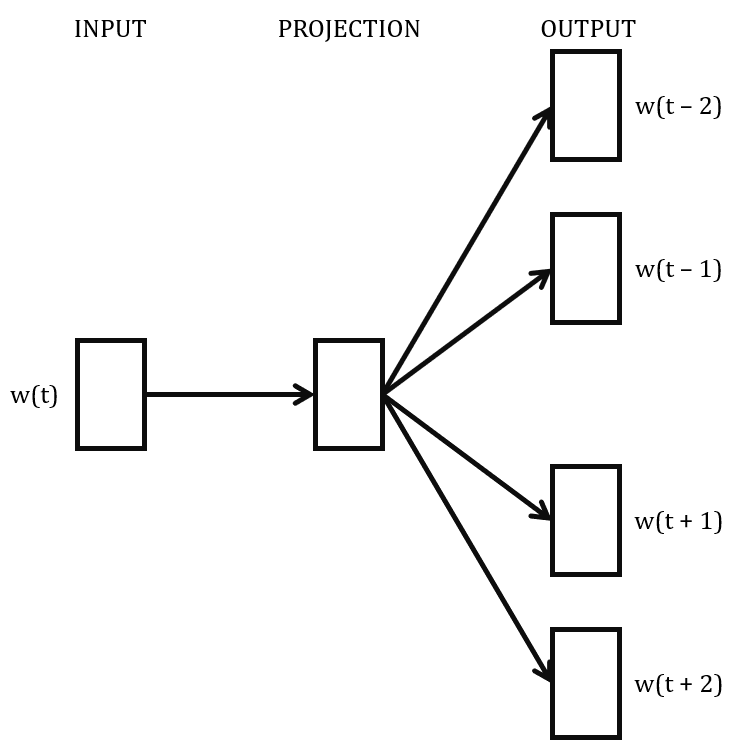
1.1.2 Mô hình Skip-gram

1.1.1.1 Ý nghĩa

Mô hình Skip-gram sử dụng một từ để dự đoán ngữ cảnh mục tiêu hay các từ xung quanh (Continuous skip-gram) xem xét những từ ngữ cảnh xung quanh sẽ được đánh giá tốt hơn so với những từ trong ngữ cảnh nhưng ở vị trí xa hơn. Mặc dù thứ tự từ vẫn không được xem xét, mỗi vector của từ bối cảnh được xem xét và cân nhắc.

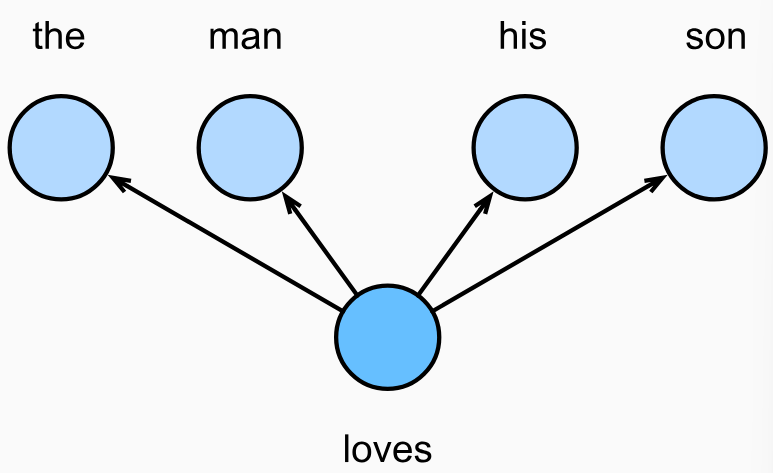
1.1.1.2 Mô hình

Mô hình Skip-gram hoạt động ngược lại với CBOW bằng cách đưa vào một từ vựng đích và dự đoán các từ vựng xung quanh nó. Cụ thể, từ vựng đích được biểu diễn bằng một vector one-hot và được đưa vào mạng nơ-ron để dự đoán các từ vựng xung quanh. Vector đầu ra được tạo ra bằng cách chuyển đổi từ vector đích thành một không gian vector có kích thước lớn hơn.



Hình 1.1.2.1 Mô hình Skip-gram dạng tổng quát

Ví dụ, giả sử chuỗi văn bản là “the”, “man”, “loves”, “his” và “son”. Ta sử dụng “loves” làm từ đích trung tâm và đặt kích thước cửa sổ ngữ cảnh bằng 2. Mô hình skip-gram quan tâm đến xác suất có điều kiện sinh ra các từ ngữ cảnh (“the”, “man”, “his” và “son”) nằm trong khoảng cách không quá 2 từ như sau:

****

Hình 1.1.2.2 Ví dụ minh họa theo mô hình Skip-gram

Giả định rằng, với từ đích trung tâm cho trước, các từ ngữ cảnh được sinh ra độc lập với nhau. Trong trường hợp này, công thức trên có thể được viết lại thành:

Trong mô hình skip-gam, mỗi từ được biểu diễn bằng hai vector d-chiều để tính xác suất có điều kiện.

Giả sử và là vector từ ngữ cảnh và vector từ đích trung tâm của từ có chỉ số trong từ điển. Gọi là chỉ số của từ đích trung tâm và là chỉ số các từ ngữ cảnh trong từ điển. Do đó, xác suất có điều kiện sinh ra từ ngữ cảnh cho một từ đích trung tâm cho trước bằng phép toán softmax trên tích vô hướng của vector là:

Trong đó, tập chỉ số trong bộ từ vựng là

Giả sử trong một chuỗi văn bản có độ dài , từ tại bước thời gian được ký hiệu là . Giả sử rằng các từ ngữ cảnh được sinh độc lập với từ trung tâm cho trước. Khi kích thước cửa sổ ngữ cảnh là , hàm hợp lý (likelihood) của mô hình skip-gam là xác suất kết hợp sinh ra tất cả các từ ngữ cảnh với bất kỳ từ trung tâm cho trước nào. Bất kỳ bước thời gian nào nhỏ hơn 1 hoặc lớn hơn đều có thể được bỏ qua.

1.1.1.3 Huấn luyện mô hình

Các tham số trong mô hình skip-gram là vector từ đích trung tâm và vector từ ngữ cảnh cho từng từ riêng lẻ. Trong quá trình huấn luyện, chúng ta sẽ học các tham số mô hình bằng cách cực đại hóa hàm hợp lý, còn gọi là ước lượng hợp lý cực đại. Việc này tương tự với việc giảm thiểu hàm mất mát sau đây:

Nếu ta dùng SGD, thì trong mỗi vòng lặp, ta chọn ra một chuỗi con nhỏ hơn bằng việc lấy mẫu ngẫu nhiên để tính toán mất mát cho chuỗi con đó, rồi sau đó tính gradient để cập nhật các tham số mô hình. Điểm then chốt của việc tính toán gradient là tính gradient của logarit xác suất có điều kiện cho vector từ trung tâm và vector từ ngữ cảnh. Đầu tiên, theo định nghĩa ta có:

Thông qua phép tính đạo hàm, ta nhận được giá trị gradient từ công thức trên.

Phép tính cho ra xác suất có điều kiện cho mọi từ có trong từ điển với từ đích trung tâm cho trước. Sau đó, ta lại sử dụng phương pháp đó để tìm gradient cho các vector từ khác.

Sau khi huấn luyện xong, với từ bất kỳ có chỉ số là trong từ điển, ta sẽ nhận được tập hai vector từ và . Trong các ứng dụng xử lý ngôn ngữ tự nhiên, vector từ đích trung tâm trong mô hình skip-gram thường được sử dụng để làm vector biểu diễn một từ.

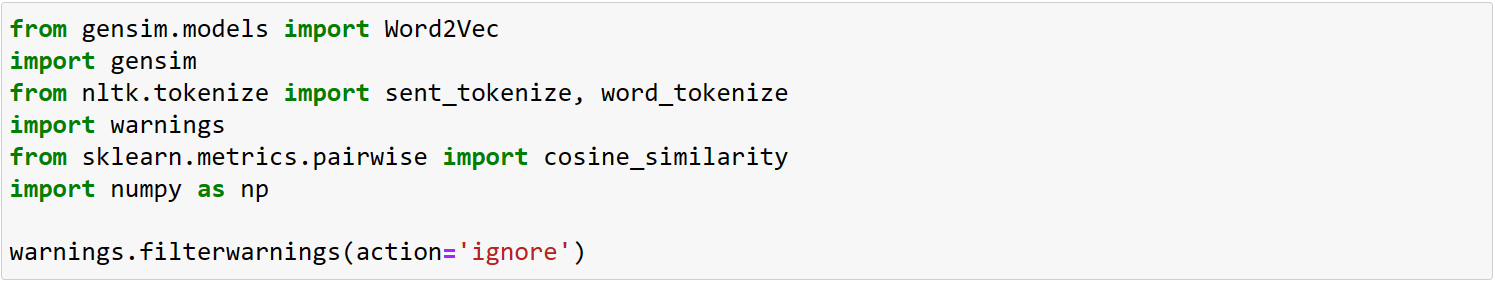
1.2 Ứng dụng vào bài toán

1.2.1 Yêu cầu

Cho môt tập các câu gọi là *.* Nhập vào 1 câu , hãy tìm câu nào đó trong tập mà tương tự với nhiều nhất.

1.2.2 Thực hiện

Bước 1: Cài đặt các thư viện cần thiết.



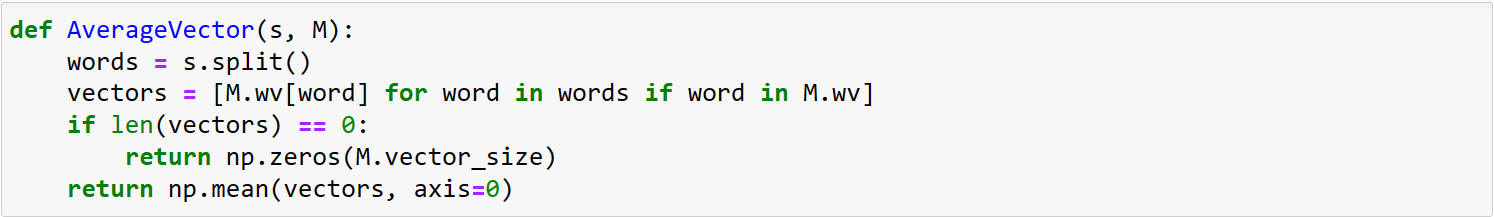
Hình 1.2.2.1 Code cài đặt các thư viện cần thiết

Ý nghĩa các thư viện:

* Thư viện gensim là thư viện chính cho mô hình Word2Vec, cung cấp khả năng huấn luyện và sử dụng các mô hình word embeddings.
* Thư viện nltk dùng cho việc tokenization.
* Thư viện sklearn dùng để tính toán độ tương tự cosine giữa các vector.
* Thư viện numpy dùng cho các phép tính toán học và ma trận, hỗ trợ tính toán nhanh chóng và hiệu quả với dữ liệu lớn.

Bước 2: Xây dựng hàm tính vector trung bình cho một câu.

Hàm tính vector đại diện cho mỗi câu trong tập dữ liệu và cho câu nhập vào . Cách làm này bao gồm việc lấy trung bình cộng của vectors của tất cả các từ trong một câu, cho ra một vector đại diện cho ngữ nghĩa tổng thể của câu đó.



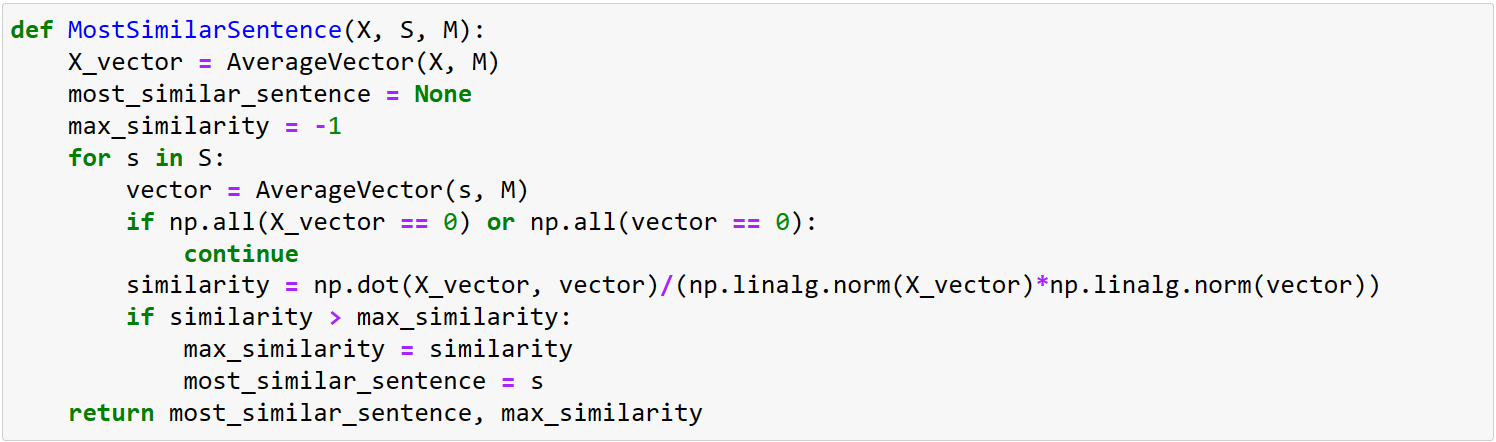
Hình 1.2.2.2 Code hàm tính vector trung bình cho một câu

Chi tiết các bước thực hiện:

* Hàm nhận 2 đối số đầu vào là câu cần tính vector trung bình (s) và mô hình Word2Vec đã được huấn luyện (M).
* Đầu tiên, phân tách câu thành các từ riêng biệt bằng cách sử dụng phương thức split().
* Sau đó, duyệt qua từng từ trong câu và kiểm tra xem từ đó có trong từ điển của mô hình Word2Vec hay không. Nếu từ đó không tồn tại trong từ điển, nó sẽ bị bỏ qua.
* Cuối cùng, lấy vector tương ứng với từng từ trong câu bằng cách sử dụng thuộc tính wv của mô hình Word2Vec. Nếu không có từ nào trong câu tồn tại trong từ điển của mô hình, hàm sẽ trả về vector zero có kích thước bằng kích thước của các vector trong mô hình. Nếu có ít nhất một từ trong câu tồn tại trong từ điển của mô hình, hàm sẽ tính toán vector trung bình của các vector từ và trả về vector đó.

Bước 3: Xây dựng hàm tìm câu tương tự nhất.

Hàm để so sánh câu nhập vào X với mỗi câu trong tập dữ liệu S dựa trên độ tương tự cosine giữa vector của câu X và vector của mỗi câu trong S. Câu có độ tương tự cosine cao nhất sẽ được chọn là câu tương tự nhất.

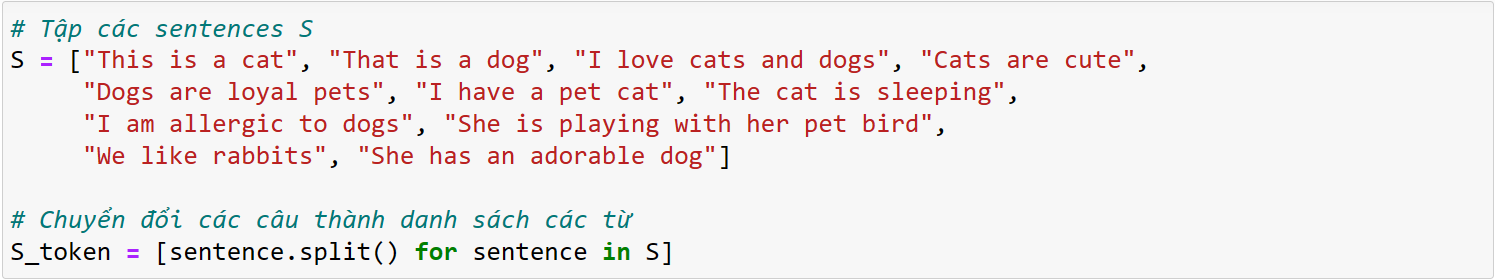


Hình 1.2.2.3 Code hàm tìm câu tương tự nhất

Chi tiết các bước thực hiện:

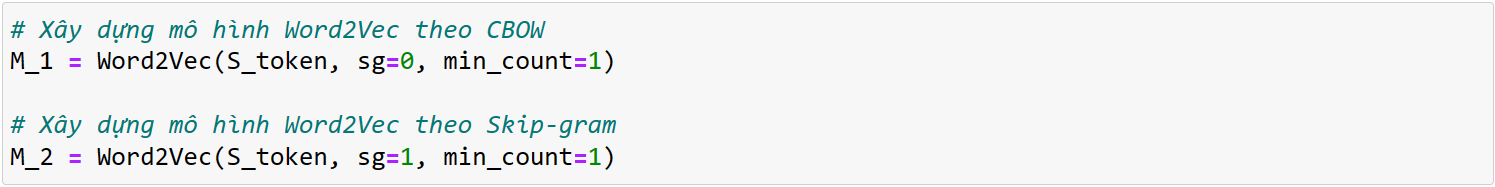
* Hàm nhận 3 đối số đầu vào là danh sách các câu trong tập dữ liệu (S), câu cần so sánh (X) và mô hình Word2Vec đã được huấn luyện (M).
* Đầu tiên, sử dụng hàm AverageVector để tính vector trung bình của các từ trong câu đầu vào X.
* Sau đó, khởi tạo 2 biến gồm 1 biến để lưu câu tương tự nhất được tìm thấy và 1 biến để lưu giá trị của câu tương tự nhất được tìm thấy.
* Tiếp theo, tạo một vòng lặp duyệt qua mỗi câu trong tập dữ liệu S để tính vector trung bình bằng cách sử dụng hàm AverageVector. Nếu vector của câu đầu vào hoặc vector của câu hiện tại rỗng thì bỏ qua và tiếp tục với câu tiếp theo. Sử dụng công thức độ tương tự cosine giữa hai vector để xác định mức độ tương tự giữa câu đầu vào và mỗi câu trong tập S. Nếu độ tương tự của câu hiện tại cao hơn giá trị của độ tương tự của câu tương tự nhất thì cập nhật biến lưu câu tương tự nhất với câu hiện tại và biến để lưu giá trị của câu tương tự nhất với giá trị của câu hiện tại.
* Cuối cùng, sau khi đã duyệt qua tất cả các câu, kết quả trả về là câu tương tự nhất và giá trị độ tương tự cao nhất tìm được.

Bước 4: Khởi tạo tập dữ liệu các câu S.



Hình 1.2.2.4 Code khởi tạo tập dữ liệu các câu S

Bước 5: Xây dựng mô hình Word2Vec theo CBOW và Skip-gram.

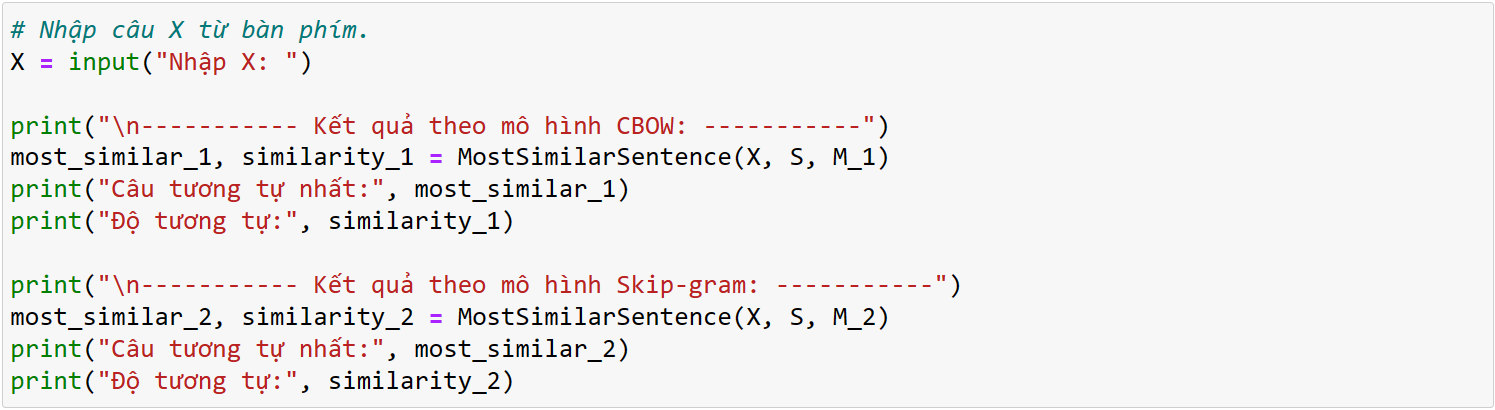


Hình 1.2.2.5 Code xây dựng mô hình theo CBOW và Skip-gram

Ý nghĩa các tham số:

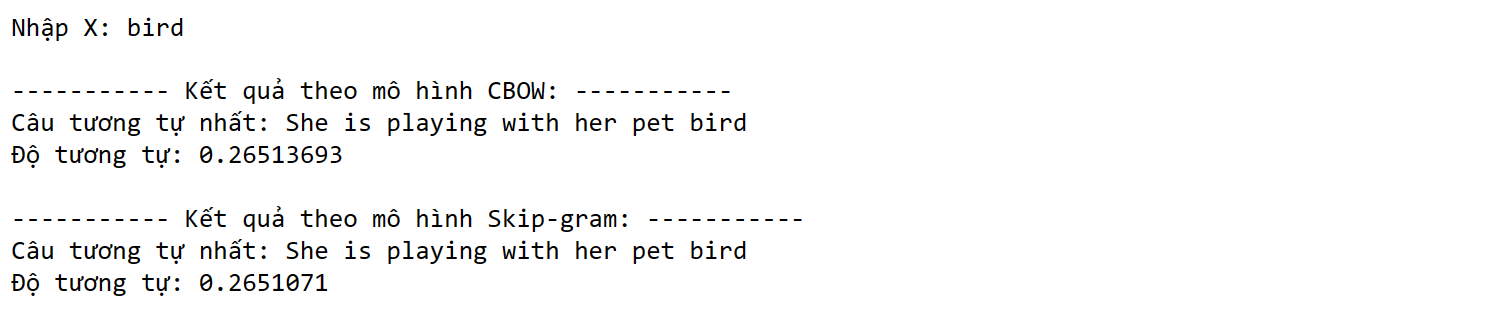
* S\_token: Danh sách các câu, mỗi câu được biểu diễn dưới dạng một danh sách các từ.
* sg: Kiến trúc của mô hình, 0 cho CBOW và 1 cho Skip-gram.
* min\_count: Số lần tối thiểu một từ xuất hiện để được xem xét. Trong trường hợp này là 1, có nghĩa là tất cả các từ đều được xem xét, kể cả những từ hiếm.

Bước 6: Nhập câu X từ người dùng và kiểm tra kết quả.



Hình 1.2.2.6 Code nhập câu X và kiểm tra kết quả

Khi X được nhập từ bàn phím là “bird” thì kết quả cho thấy rằng cả hai mô hình CBOW và Skip-gram đều cho ra câu tương tự nhất là “She is playing with her pet bird” với mức độ tương tự khoảng 0.265. Điều này cho thấy rằng cả hai mô hình đều nhận biết được mối liên hệ giữa từ “bird” và câu này, dù có thể có sự chênh lệch nhỏ về độ tương tự. Điều này cũng cho thấy rằng việc chọn kiến trúc mô hình (CBOW hoặc Skip-gram) không ảnh hưởng nhiều đến kết quả trong trường hợp này.



Hình 1.2.2.7 Kết quả trả về

CHƯƠNG 2 – MÔ HÌNH PHÂN LOẠI VĂN BẢN

2.1 Tập dữ liệu

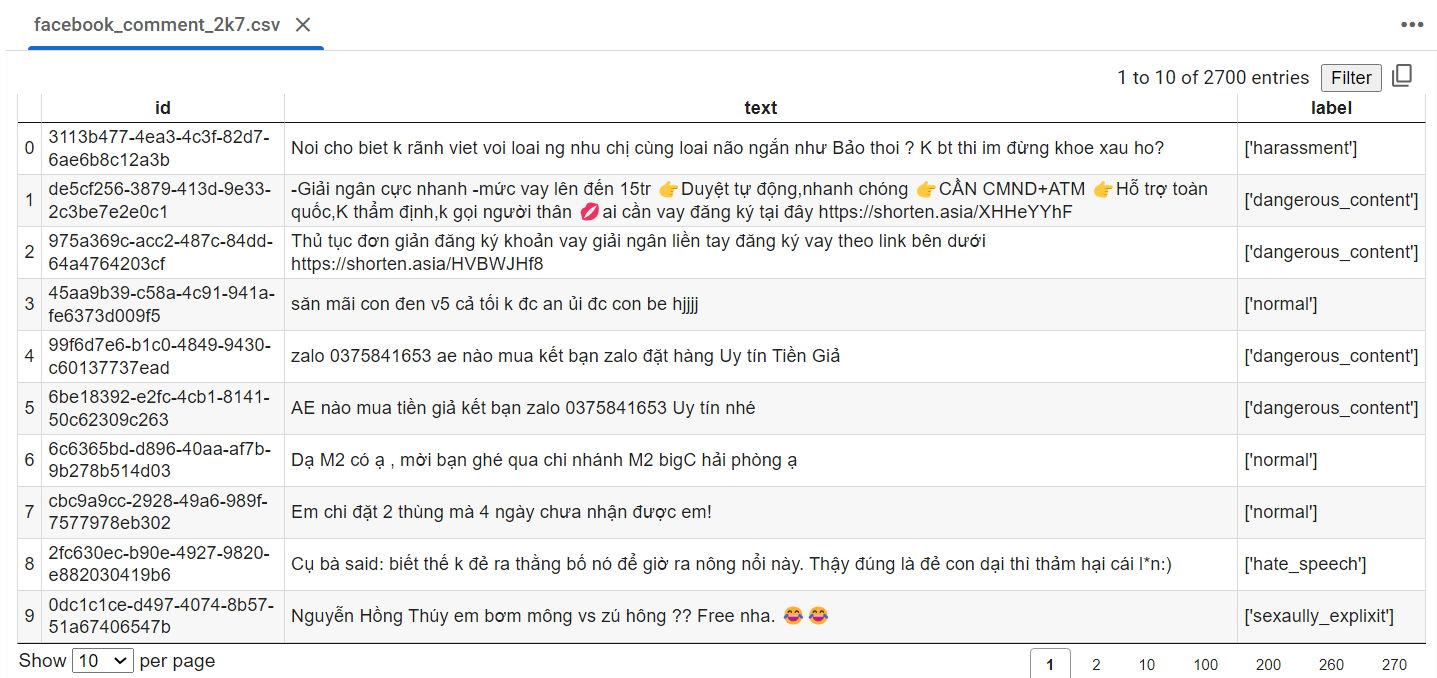
Tập dữ liệu sử dụng trong bài này là tập dữ liệu “facebook\_comment\_2k7” được cung cấp sẵn lấy từ link sau: [*facebook\_comment\_2k7*](https://drive.google.com/file/d/1pMwMc89Ff_TOAiEcJe8Hm0ZLtJxeHQiC/view)

Về nhãn, tập dữ liệu “facebook\_comment\_2k7” có 5 nhãn gồm harrasment, hate speech, sexually explicit, dangerous content và nomal. Ý nghĩa các nhãn được mô tả như sau:

* Harrasment: Đây là những ý kiến, nhận xét hoặc phê phán mang tính tiêu cực hoặc gây tổn hại đến người khác. Nó có thể bao gồm chỉ trích, phê phán hoặc tấn công trực tiếp danh tính của một người, bao gồm các khía cạnh như giới tính, dân tộc, tôn giáo, hoặc bất kỳ thuộc tính nào khác liên quan đến danh tính của họ.
* Hate speech: Đây là những từ ngữ hoặc hành động không phù hợp, thô tục hoặc thiếu văn minh thường liên quan đến các nội dung không thích hợp để sử dụng trong môi trường công cộng hoặc trao đổi, giao tiếp văn hóa lịch sự.
* Sexually explicit: Đây là nội dung đề cập hoặc mô tả về các hành động hoặc hoạt động liên quan đến tình dục, bao gồm những hành vi hay cử chỉ có tính chất tình dục. Hay những nội dung mang tính chất khiêu dâm hoặc tục tĩu, thường liên quan đến việc sử dụng ngôn từ hoặc hình ảnh mô tả một cách tục tĩu hoặc không phù hợp về tình dục.
* Dangerous content: Đây là hành động khuyến khích, tạo điều kiện hoặc thúc đẩy các hành vi có hại, bao gồm việc động viên hoặc hỗ trợ những hành động mà có thể gây ra hậu quả tiêu cực, nguy hiểm hoặc gây tổn thương cho cá nhân hoặc cộng đồng. Điều này có thể bao gồm khuyến khích vi phạm pháp luật, bạo lực, gây rối, lạm dụng, đố ky, hay bất kỳ hành động nào gây hại cho sức khỏe, an toàn và trí tuệ của mọi người.
* Nomal: Được sử dụng cho các bình luận hoặc nội dung không thuộc vào bất kỳ nhóm nào trong các nhãn trên, tức là không chứa nội dung quấy rối, kỳ thị, tình dục hoặc nguy hiểm.

Về số lượng, bộ dữ liệu có 2700 dòng gồm 4 cột:

* stt: Đây có thể là cột chỉ số hàng hoặc số thứ tự của mỗi bình luận trong tập dữ liệu. Cột này thường được sử dụng để xác định vị trí hoặc thứ tự của mỗi mẫu dữ liệu.
* id: Cột này có thể chứa các định danh duy nhất cho mỗi bình luận hoặc người dùng. Thông qua cột này, có thể xác định được nguồn gốc hoặc người tạo nội dung.
* text: Đây là cột chứa nội dung của mỗi bình luận trên Facebook. Nội dung của bình luận thường được sử dụng để phân loại hoặc đánh giá tính chất của bình luận.
* label: Đây là cột chứa nhãn hoặc loại của mỗi bình luận, chỉ ra tính chất của nội dung, như đã mô tả trong câu hỏi trước đó. Cột này chứa thông tin về những gì mà bình luận được phân loại là “harassment”, “hate speech”, “sexually explicit”, “dangerous content” hoặc “normal”.

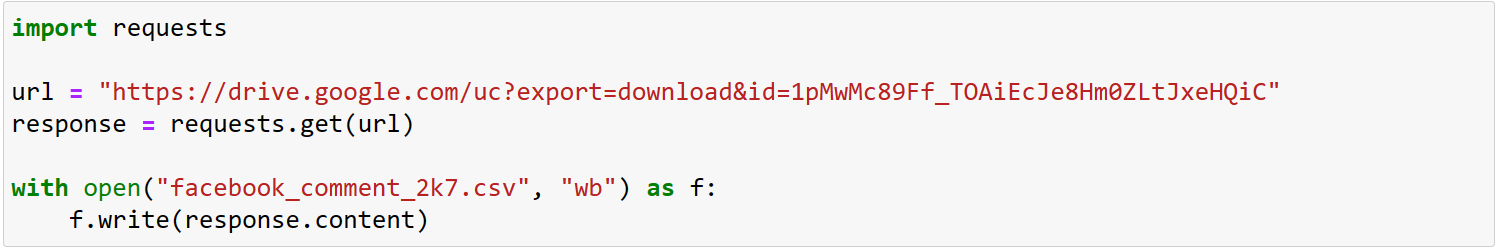


Hình 2.1 Tập dữ liệu facebook\_comment\_2k7

2.2 Tiền xử lí dữ liệu

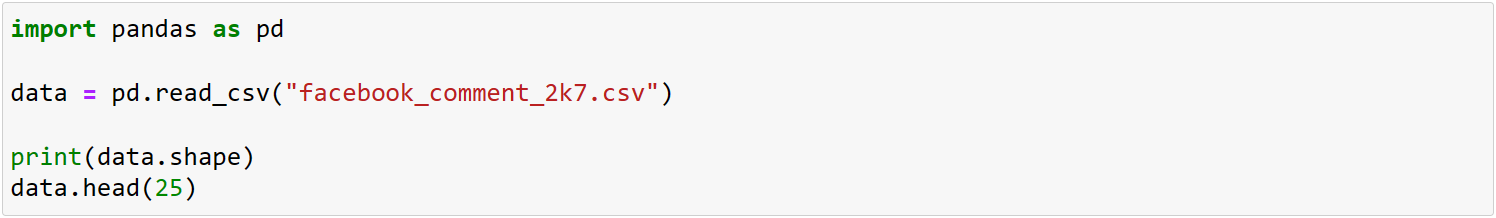
Để đưa dữ liệu trên vào các mô hình để huấn luyện và kiểm tra, trước tiên phải xem xét kiểu dữ liệu của các đặc trưng, xem xét dữ liệu ở các dòng, xem xét các vấn đề có thể gặp phải trên bộ dữ liệu để từ đó hiệu chỉnh tập dữ liệu thành 1 tập dữ liệu hoàn chỉnh để khi đưa vào mô hình huấn luyện và kiểm tra sẽ cho độ chính xác cao hơn.

Đầu tiên, sử dụng thư viện requests để tải xuống dữ liệu từ URL đã được cung cấp. Sau khi dữ liệu được tải về, nó được lưu vào một tệp cục bộ với tên “facebook\_comment\_2k7.csv”.



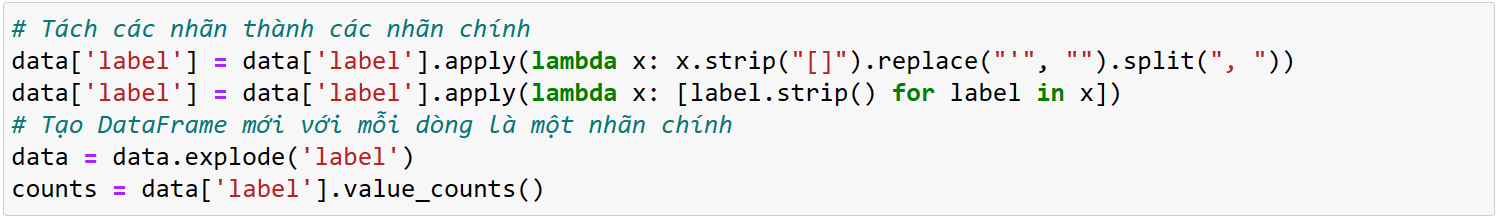
Hình 2.2.1 Code tải dữ liệu

Sau đó, thư viện Pandas được sử dụng để đọc dữ liệu từ tệp CSV này vào một DataFrame để thực hiện các xử lý tiếp theo.



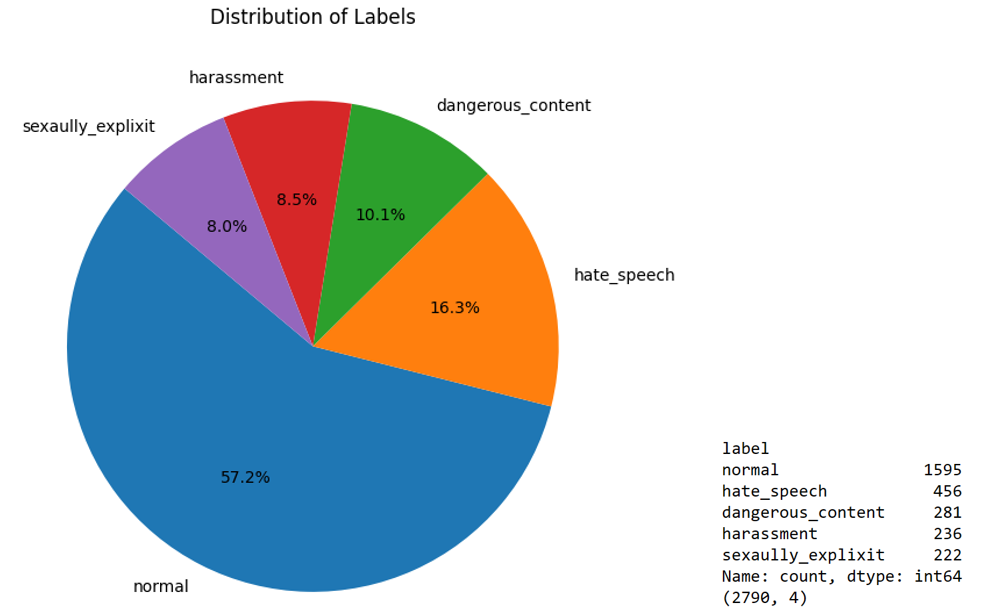
Hình 2.2.2 Code đọc dữ liệu

Tiếp theo, tách dữ liệu có nhiều nhãn thành mỗi hàng để mỗi dòng dữ liệu là một nhãn chính. Khi tách thì loại bỏ các ký tự không cần thiết và tạo thành một danh sách các nhãn. Dữ liệu sau khi xử lý được trực quan hóa thông qua biểu đồ để hiển thị phân phối của các nhãn.



Hình 2.2.3 Code tách nhãn chính cho từng dòng dữ liệu

Cuối cùng, in ra màn hình và trực quan hóa số lượng các nhãn. Kết quả thu được ta thấy rằng, dữ liệu ban đầu là 2700 và số lượng dữ liệu hiện tại là 2790. Có nghĩa là có 90 dữ liệu là có 2 nhãn trở lên.



Hình 2.2.4 Kết quả so sánh tỉ lệ số lượng các nhãn

2.3 Mô hình phân loại văn bản và độ đo

2.3.1 Mô hình phân loại văn bản

2.3.1.1 Logistic Regression

* Định nghĩa: Logistic Regression là mô hình học máy được sử dụng cho bài toán phân loại. Nó dựa trên hàm logistic để ước lượng xác suất rơi vào một lớp nhất định.
* Công thức của hàm logistic:

Trong đó:

* với là các biến đầu vào.
* là các tham số cần tìm.

2.3.1.2 Naive Bayes

* Định nghĩa: Naive Bayes là một thuật toán phân loại được mô hình hóa, nó hoạt động dựa trên tính toán xác suất áp dụng định lý Bayes, giả định các biến đầu vào là độc lập với nhau.
* Công thức:

Trong đó:

* + là xác suất của y với điều kiện X.
  + là xác suất của X khi đã biết y.
  + là xác suất của X.
  + là xác suất của y.
  + là vector đặc trưng như sau:
  + đạt được cực đại khi

2.3.1.3 Random Forest

* Định nghĩa: Random Forest là một thuật toán supervised learning, có thể giải quyết cả hai bài toán regression và classification. Random Forest kết hợp nhiều cây quyết định (Decision Trees) để tạo ra kết quả dự đoán.
* Mô hình này không có công thức cụ thể. Nó hoạt động bằng cách kết hợp cây quyết định và áp dụng voting hoặc averaging để đưa ra kết quả dự đoán.
  + - 1. Support Vector Machine
* Định nghĩa: Support Vector Machine là một thuật toán giám sát, nó dùng được cho hai bài toán phân loại (classification) và hồi quy (regression). Support Vector Machine tạo ra một ranh giới phân chia tối ưu giữa các lớp.
* Công thức được mô tả như sau:

Cho một tập dữ liệu huấn luyện gồm mẫu dữ liệu:

Trong đó:

* là mẫu dữ liệu đầu vào
* là nhãn tương ứng (có thể là -1 hoặc 1).

Mục tiêu của SVM là tìm ra siêu phẳng (hyperplane) tốt nhất để phân chia không gian đa chiều sao cho các điểm dữ liệu được phân loại đúng và có độ lớn lề (margin) lớn nhất. Siêu phẳng này có thể được biểu diễn bởi phương trình:

Trong đó:

* là vector trọng số của siêu phẳng,
* là vector đầu vào của mẫu dữ liệu,
* là hệ số điều chỉnh.

Mục tiêu của SVM là tối ưu hóa và sao cho:

* Mỗi mẫu dữ liệu được phân loại đúng, tức là
* Margin là lớn nhất, tức là độ lớn nhỏ nhất của

Để tối ưu hóa bài toán này, chúng ta thường sử dụng phương pháp tối ưu hoá convex. Trong SVM, có hai dạng phổ biến là:

* Hard-margin SVM: Một mô hình SVM mà yêu cầu tất cả các mẫu dữ liệu đều phải nằm đúng về một phía của siêu phẳng.
* Soft-margin SVM: Một mô hình SVM mà cho phép một số mẫu dữ liệu bị phân loại sai, nhằm tối ưu hóa margin và giảm overfitting.

Công thức tối ưu hoá mục tiêu trong bài toán SVM thường dùng là Lagrange dual formulation, được giải bằng các phương pháp tối ưu hoá như Sequential Minimal Optimization (SMO) hoặc gradient descent.

2.3.1.5 Multilayer Perceptron

* Định nghĩa: MLP là một mô hình mạng nơ-ron sâu, có nhiều lớp ẩn (Hidden layer) giữa lớp đầu vào (Input layer) và lớp đầu ra (Output Layer).
* Công thức được mô tả như sau:
* Đầu vào cho nơ-ron trong lớp ẩn:

Trong đó:

* là tổng trọng số đầu vào.
* là trọng số kết nối giữa nơ-ron đầu vào thứ .
* là giá trị đầu vào từ nơ-ron đầu vào thứ .
* là ngưỡng (bias) của nơ-ron ẩn thứ .
* là số lượng nơ-ron trong lớp đầu vào.
* Đầu ra của nơ-ron trong lớp ẩn sau khi áp dụng hàm kích hoạt :

Trong đó thường là một hàm phi tuyến tính như hàm sigmoid, hàm tanh, hoặc hàm ReLU.

* Đầu vào cho nơ-ron trong lớp đầu ra:

Trong đó:

* là tổng trọng số đầu vào.
* là trọng số kết nối giữa nơ-ron ẩn thứ và nơ-ron đầu vào thứ .
* là đầu ra của nơ-ron ẩn thứ .
* là ngưỡng (bias) của nơ-ron đầu ra thứ .
* là số lượng nơ-ron trong lớp ẩn.
* Đầu ra của nơ-ron trong lớp đầu ra sau khi áp dụng hàm kích hoạt :

Trong đó:

* là giá trị đầu ra dự đoán của nơ-ron đầu ra thứ
* là một hàm kích hoạt của lớp đầu ra.

Trong quá trình huấn luyện, các trọng số và ngưỡng được điều chỉnh để giảm thiểu sai số giữa giá trị dự đoán và giá trị thực tế bằng cách sử dụng các thuật toán tối ưu hóa như gradient descent và backpropagation.

2.3.2 Độ đo

* Accuracy: Độ chính xác của mô hình được tính toán 1 cách đơn giản thông qua tỉ lệ giữa số điểm dữ liệu được dự đoán đúng trên tổng số lượng điểm dữ liệu. Tuy nhiên, một mô hình có độ chính xác cao chưa hẳn đã tốt. Accuracy lộ rõ hạn chế khi được sử dụng trên bộ dữ liệu không cân bằng (imbalanced dataset).
* Precision: Là tỉ lệ số điểm TP của 1 class trong tất cả các điểm được mô hình phân loại là TP. Precision sẽ được tính bằng công thức:

Khi Precision = 1, mọi điểm tìm được đều thực sự là positive, tức không có điểm negative nào lẫn vào kết quả. Nhưng vẫn chưa đủ để đánh giá là mô hình tốt nhất vì vẫn không biết rằng mô hình đã tìm ra được tất cả điểm positive hay chưa.

* Recall: Đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive. Công thức của recall như sau:

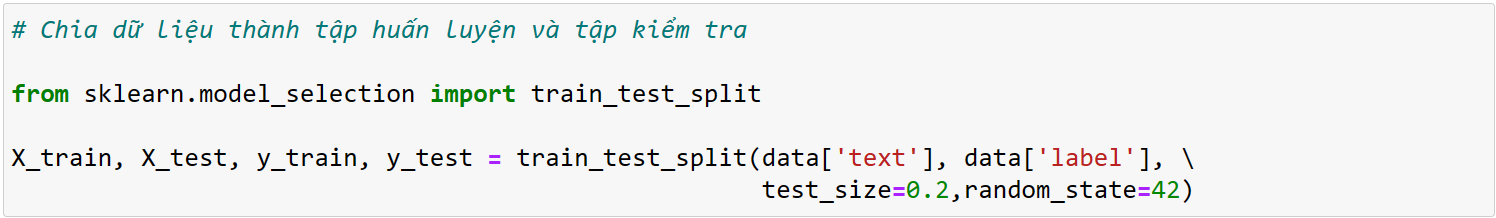
Khi Recall = 1, mọi điểm positive đều được tìm thấy. Tuy nhiên, đại lượng này lại không đo liệu có bao nhiêu điểm negative bị lẫn trong đó.

* F1-score: Đánh giá chính xác độ hiệu quả của mô hình dựa trên 2 đại lượng là Recall và Precision. Như ta đã thấy ở trên, chỉ duy nhất Precision hoặc Recall thì không thể đánh giá độ hiệu quả của một mô hình, vì thế người ta kết hợp 2 đại lượng thành đại lượng đo mới là F1-score. F1 có giá trị càng cao thì mô hình cho hiệu quả càng tốt.

2.3.3 Áp dụng mô hình và so sánh kết quả

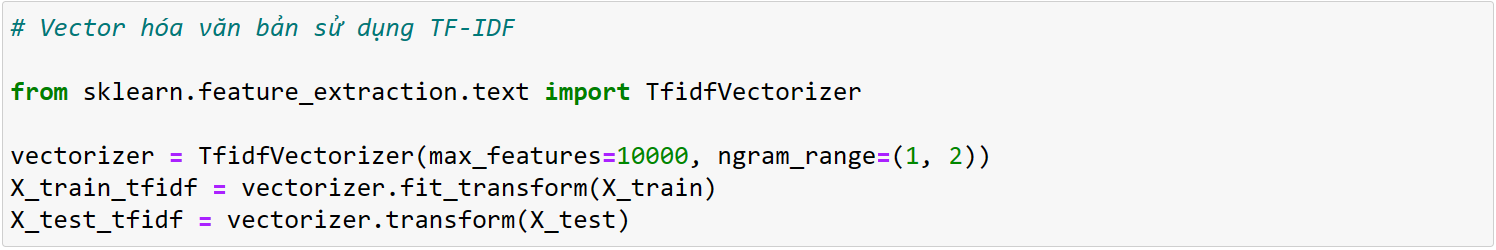
Việc áp dụng các mô hình sẽ thông qua thư viện sklearn của python. Các bước thực hiện bao gồm:

Bước 1: Chia tập dữ liệu đã qua xử lí để đưa vào mô hình dùng cho việc huấn luyện và kiểm tra, ta sử dụng train\_test\_split.



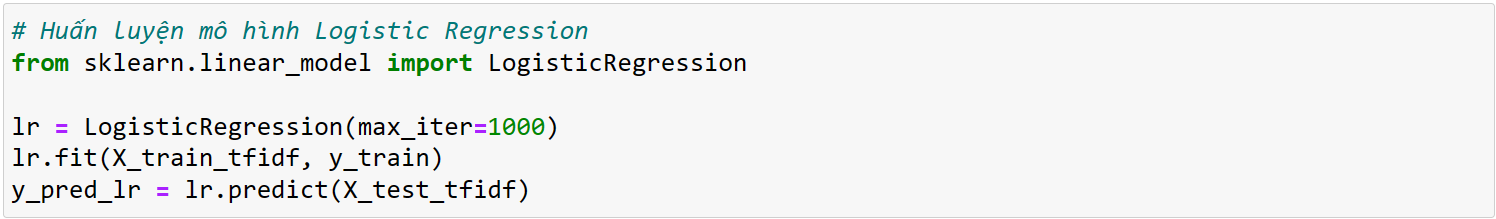
Hình 2.3.3.1 Code chia dữ liệu thành tập huấn luyện

Bước 2: Sử dụng phương pháp TF-IDF để vector hóa văn bản biểu diễn các bình luận dưới dạng các vectơ số. Điều này giúp mô hình có thể làm việc với dữ liệu dưới dạng số.

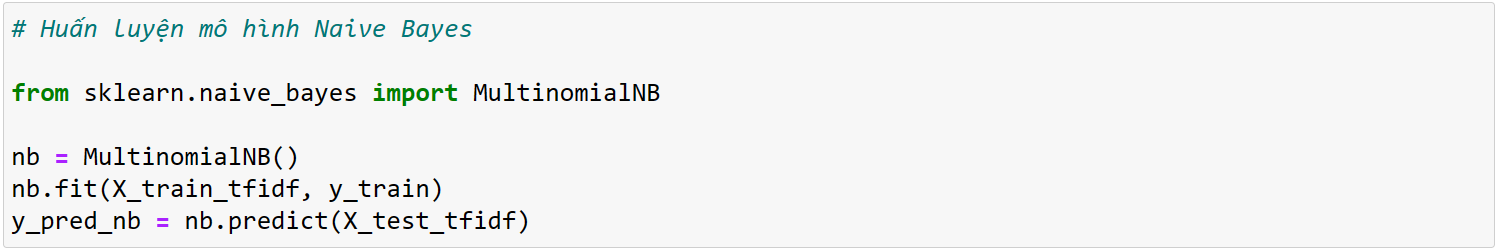


Hình 2.3.3.2 Code vector hóa văn bản

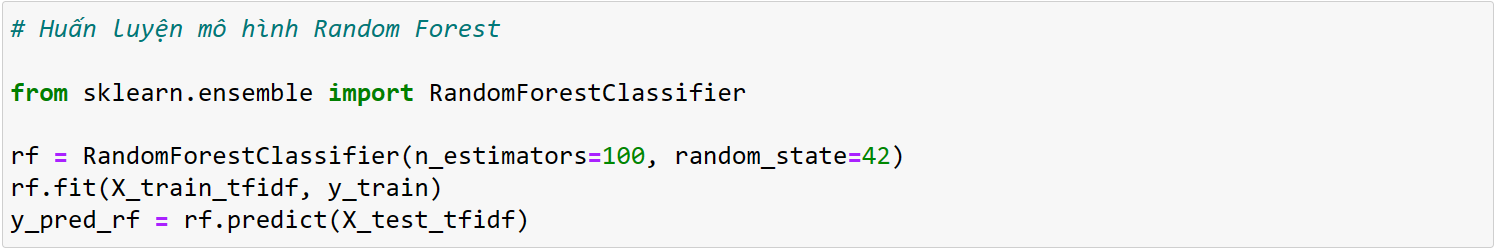
Bước 3: Huấn luyện các mô hình phân loại văn bản.



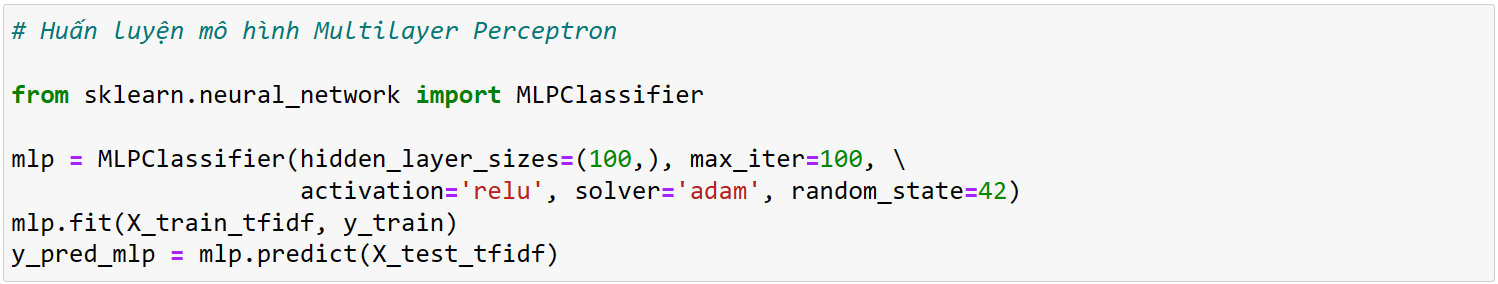
Hình 2.3.3.3 Code huấn luyện mô hình LR



Hình 2.3.3.4 Code huấn luyện mô hình NB



Hình 2.3.3.5 Code huấn luyện mô hình RF



Hình 2.3.3.6 Code huấn luyện mô hình MLP

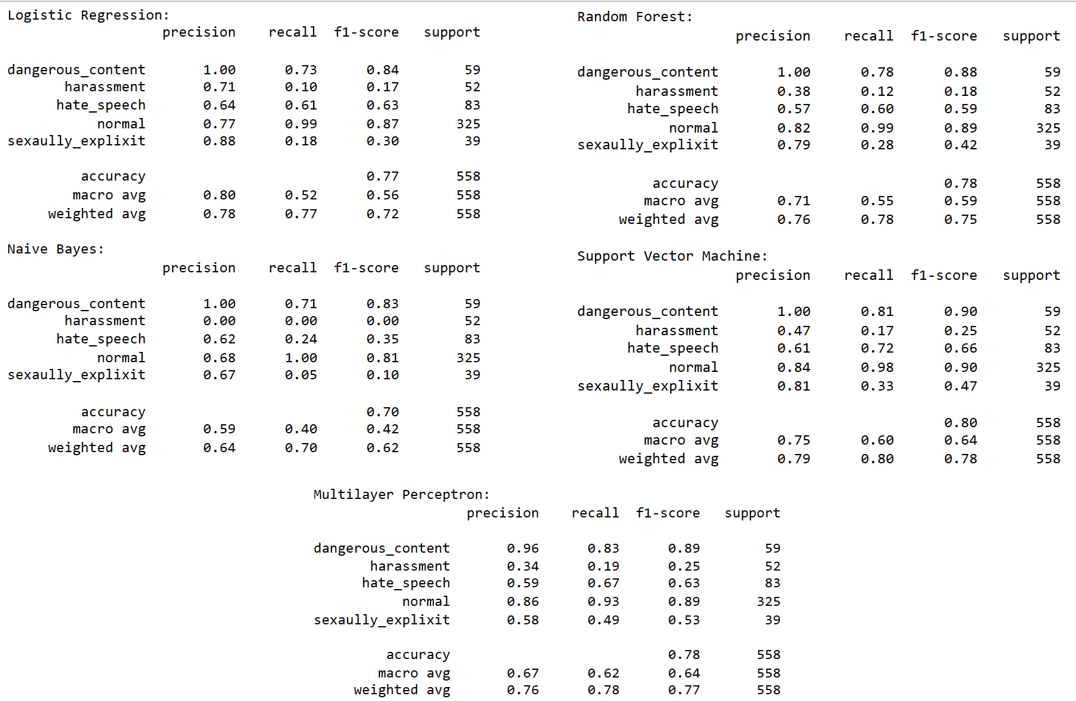


Hình 2.3.3.7 Code huấn luyện mô hình SVM

Bước 4: Đánh giá các mô hình phân loại văn bản và sử dụng các metric như Accuracy, Precision, Recall và F1-score để đánh giá hiệu suất của mỗi mô hình. Cụ thể, sử dụng classification\_report để xuất các metric này.

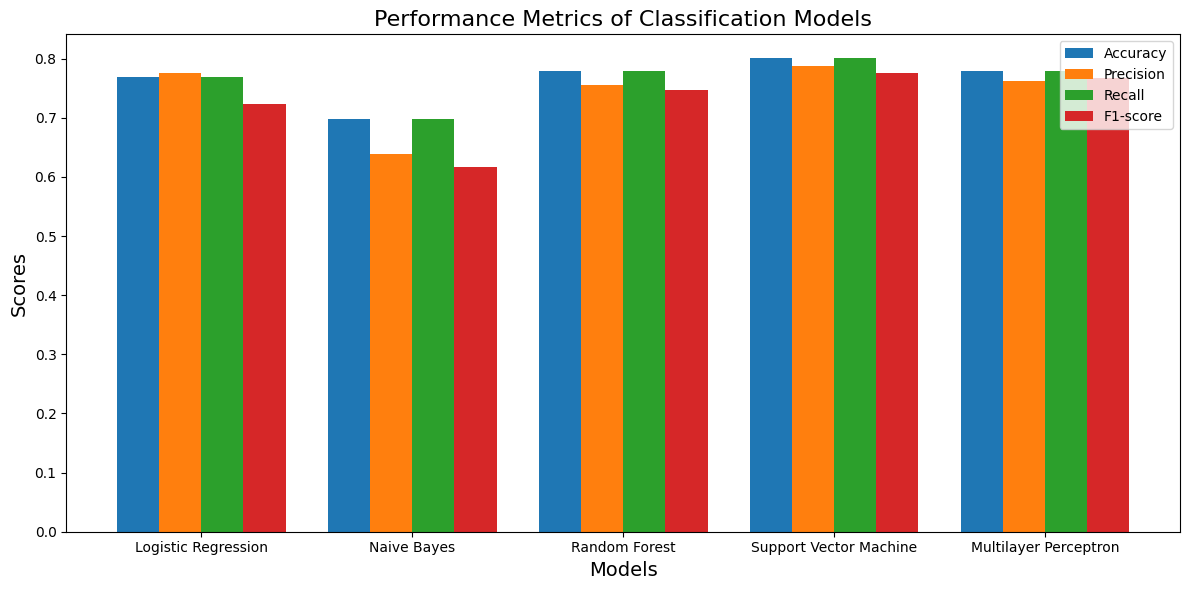


Hình 2.3.3.8 Code đánh giá 5 loại mô hình



Hình 2.3.3.9 Kết quả đánh giá 5 loại mô hình

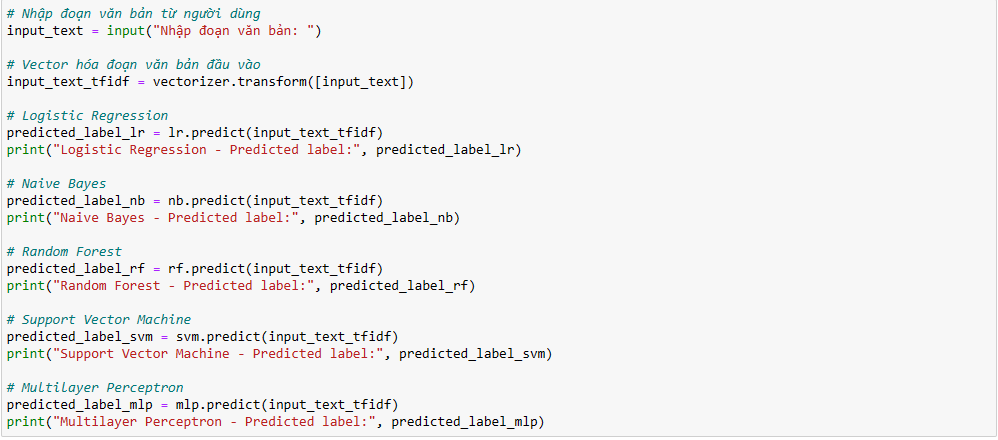
Bước 5: Trực quan hóa các metric trên tập kiểm tra mỗi mô hình thông qua biểu đồ cột.



Hình 2.3.3.10 Kết quả so sánh các metric 5 loại mô hình

2.4 Phân tích dự đoán nhãn

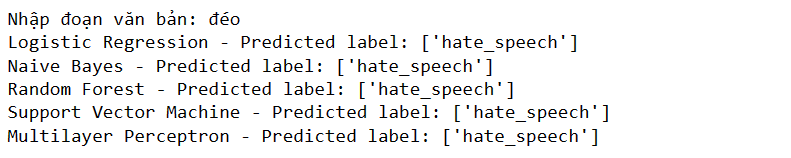
Quá trình này cho phép người dùng nhập một đoạn văn bản và dự đoán nhãn của nó sử dụng các mô hình đã được huấn luyện trước. Kết quả dự đoán từ mỗi mô hình có thể giúp người dùng hiểu hơn về cách mỗi mô hình đưa ra dự đoán và cung cấp sự đa dạng trong việc đánh giá.



Hình 2.4.1 Code phân tích dự đoán nhãn

Chi tiết các bước thực hiện:

* Bước 1: Người dùng được yêu cầu nhập một đoạn văn bản.
* Bước 2: Vector hóa đoạn văn bản đầu vào cũng sử dụng cùng phương pháp TF-IDF đã được sử dụng trong quá trình huấn luyện mô hình.
* Bước 3: Dự đoán nhãn cho các văn bản mới dựa trên mô hình phân loại đã được huấn luyện trước đó và dự đoán được in ra màn hình cho mỗi mô hình.



Hình 2.4.2 Kết quả dự đoán nhãn

2.5 Phân tích dữ liệu văn bản

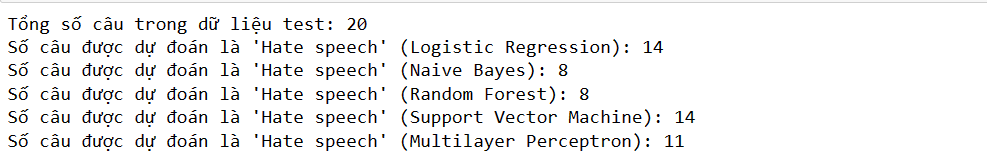
Quá trình này cung cấp cái nhìn về khả năng của mỗi mô hình trong việc dự đoán các câu chứa từ ngữ khiêu khích hoặc tiêu cực. Việc so sánh các kết quả này có thể giúp đánh giá hiệu suất và tính linh hoạt của các mô hình phân loại trong các tình huống thực tế.



Hình 2.5.1 Code phân tích dữ liệu văn bản

Chi tiết các bước thực hiện:

* Bước 1: Cung cấp một dữ liệu test gồm một danh sách các câu được chọn mẫu để kiểm tra hiệu suất của các mô hình phân loại.
* Bước 2: Dự đoán nhãn “Hate Speech” cho các câu trong dữ tiệu test:
* Bước 3: Tính toán và in ra màn hình số lượng câu được dự đoán là “Hate speech” bởi mỗi mô hình.



Hình 2.5.1 Kết quả phân tích dữ liệu văn bản

CHƯƠNG 3 – KẾT LUẬN

3.1 Kết quả đạt được

Trong quá trình thực hiện, nhóm 10 đã đạt được các kết quả sau:

* Hiểu rõ khái niệm và đặc điểm của Word2Vec, nắm được lợi ích của việc sử dụng Word2Vec trong lĩnh vực xử lý ngôn ngữ tự nhiên.
* Đã tìm hiểu về ý nghĩ, mô hình, cách huấn luyện mô hình của phương pháp CBOW và Skip-gram.
* Sử dụng hiệu quả tài nguyên từ Python thông qua các thư viện MLPClassifier, SVC, MultinomialNB, nltk, LogisticRegression, …

3.2 Thuận lời và khó khăn

3.2.1 Thuận lợi

* Dữ liệu sẵn có: Có sẵn dữ liệu từ tập tin facebook\_comment\_2k7.csv, giúp tiết kiệm thời gian và công sức thu thập dữ liệu.
* Thư viện hỗ trợ: Được sử dụng nhiều thư viện có sẵn để triển khai trên nhiều mô hình.
* Tích hợp với Python: một ngôn ngữ lập trình phổ biến và dễ học, giúp việc phát triển và triển khai dễ dàng hơn.

3.2.2 Khó khăn

* Phải học hỏi và hiểu biết về các khái niệm và cách thức hoạt động của các mô hình để sử dụng các mô hình hiệu quả,
* Triển khai nhiều mô hình cần phải sử dụng nhiều thư viện đòi hỏi kiến thức đủ tốt về Python.

3.3 Đánh giá

|  |  |  |
| --- | --- | --- |
| **Câu** | **Tự đánh giá** | **Độ hoàn thành** |
| 1 – Mô hình Word2Vec | Tốt | 100% |
| 2 – Mô hình phân loại văn bản | Tốt | 100% |

Bảng 4.3 Đánh giá độ hoàn thành đồ án

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Tiệp Vũ (2021), “[*Word2vec*](https://machinelearningcoban.com/tabml_book/ch_embedding/word2vec.html)”, Machine Learning cơ bản [Online].
2. Đ.V.D.Thanh, N.V.Quang, N.V.Cường, P.Đ.Khoa và L.K.H.Phúc (2020), *“*[*Xử lý Ngôn ngữ Tự nhiên: Tiền Huấn luyện*](https://d2l.aivivn.com/chapter_natural-language-processing-pretraining/word2vec_vn.html)*”*, Đắm mình vào học sâu [Online].
3. Nguyễn Đình Mạnh (12/2020), *“Nghiên cứu các phương pháp tính toán độ tương tự của văn bản luật tiếng việt”*, Đại học Quốc gia Hà Nội, Trường Đại học Công Nghệ.
4. Bùi Văn Hợp (July 4, 2021), Giới thiệu về mô hình SVM – Phần 1.
5. Huỳnh Chí Trung (2020), Giới thiệu về Support Vector Machine (SVM).
6. Huỳnh Chí Trung (2020), Phân lớp với Naive Bayes Classification – Mô hình và ứng dụng.
7. Phạm Đình Khánh (2021), Giới thiệu về mô hình rừng cây (Random Forest).
8. Nguyễn Duy Sim (2018), Phân lớp bằng Random Forests trong Python.

**Tiếng Anh**

1. Mimi Dutta (Nov 9, 2023), *“Word2Vec For Word Embeddings - A Beginner’s Guide”*, Analytics Vidhya.
2. S Bishop, Christopher M. (2006), *“Pattern recognition and Machine Learning.”*, Springer.