



**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ ĐIỆN TỬ**

**----🙣🕮🙡----**

**Học Máy**

**Đề tài:**

**Natural Language Processing with Disaster Tweets**

**NHÓM 14**

***Giảng viên hướng dẫn*: Phạm Việt Thành**

***Sinh viên thực hiện*: Nguyễn Duy Tùng 21085061**

**21001201**

**21136041**

**21135711**

**21071741**

***Lớp*: DHIOT17A**

*TP. Hồ Chí Minh, tháng 11, năm 2024*

**Tên đề tài: Natural Language Processing with Disaster Tweets**

***Lớp*: DHIOT17A Nhóm: 14**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và Tên** | **Nhiệm vụ Công việc trong nhóm** | **Điểm tự đánh giá  của nhóm** | **Điểm đánh giá  của giáo viên** |
| 1 | 21001211 | Nguyễn Huy Thái | Huấn luyện và đánh giá mô hình, mã hóa các cột danh mục. Thực hiện powerpoint, word | 10 |  |
| 2 | 21001201 | La Kỳ Nguyên | Xử lý dữ liệu bị thiếu, chuẩn hóa dữ liệu. Thực hiện powerpoint,word | 10 |  |
| 3 | 21136041 | Đinh Quốc Huy | Trực quan hóa dữ liệu,Thực hiện powerpoint,word | 8 |  |
| 4 | 21071741 | Lê Thị Hoàng Anh | Xác định tỉ lệ rỗng các cột dữ liệu. Thực hiện powerpoint,word | 8 |  |
| 5 | 21135711 | Đỗ Mạnh Dũng | Chuyển đổi các cột từ dạng số sang danh mục. Thực hiện powerpoint | 8 |  |

MỤC LỤC

[BÁO CÁO KẾT QUẢ BÀI TẬP LỚN 5](#_Toc181459560)

[1. Mục tiêu đề tài: 5](#_Toc181459561)

[2. Tổng quan về đề tài 5](#_Toc181459562)

[3. Các nội dung đã thực hiện 7](#_Toc181459563)

[3.1. Phân tích và khám phá dữ liệu 7](#_Toc181459564)

[3.2. Phân tích sự phân bố dữ liệu 8](#_Toc181459565)

[3.3. Xử lý tiền văn bản 11](#_Toc181459566)

[3.3.1. Tách văn bản từ tập dữ liệu – Tokenization 11](#_Toc181459567)

[3.3.2. Loại bỏ các tử thông dụng không có giá trị để lọc nhiễu – Stop Words Removal 12](#_Toc181459568)

[3.3.3. Rút gọn từ về dạng gốc - Stemming 13](#_Toc181459569)

[3.4. Tạo ra các đặc trưng dựa trên tần xuất từ để xác định mức quan trọng của từ - TF – IDF (Term frequency – Inverse Document Frequency) 14](#_Toc181459570)

[3.5. Xây dựng các đặc trưng từ văn bản để làm đầu vào cho mô hình – Feature Engineering 19](#_Toc181459571)

[3.6. Sử dụng mô hình học máy để phân loại văn bản - Classification 19](#_Toc181459572)

[3.7. Đánh giá: 19](#_Toc181459573)

[4. Các kết quả đã đạt được (chưa đạt được) 22](#_Toc181459574)

[4.1. Kết quả đạt được 22](#_Toc181459575)

[4.2. Các kết quả chưa đạt được 22](#_Toc181459576)

[5. Những khó khăn, vướng mắc gặp phải trong quá trình thực hiện 23](#_Toc181459577)

[6. Những kinh nghiệm thu được sau khi hoàn thành bài tập lớn 24](#_Toc181459578)

[7. Kết luận (và tự đánh giá theo thang điểm từ 0 đến 10) 25](#_Toc181459579)

[8. Tài liệu tham khảo 26](#_Toc181459580)

BÁO CÁO KẾT QUẢ BÀI TẬP LỚN

# Mục tiêu đề tài:

Phân loại tin tức về thảm họa tự động: xây dựng mô hình xử lý ngôn ngữ tự nhiên (Natural Language Processing hay NLP) nhằm phân loại các dòng Tweets (nền tảng mạng xã hội) thành hai nhóm chính là tin có nội dung liên quan đế thảm họa và tin không liên quan, từ đó giúp nhận diện sớm các thông tin hữu ích trong các tình huống khẩn cấp

Hiểu rõ hơn về cách sử dụng các kỹ thuật NLP: Tìm hiểu và ứng dụng các kỹ thuật tiên tiến trong NLP như stemming, loại bỏ từ dừng, xử lý các đặc trưng và trọng số của từ (như IDF – Inverse Documents Frequency), đồng thời tối ưu các tham số mô hình để cải thiện độ chính xác.

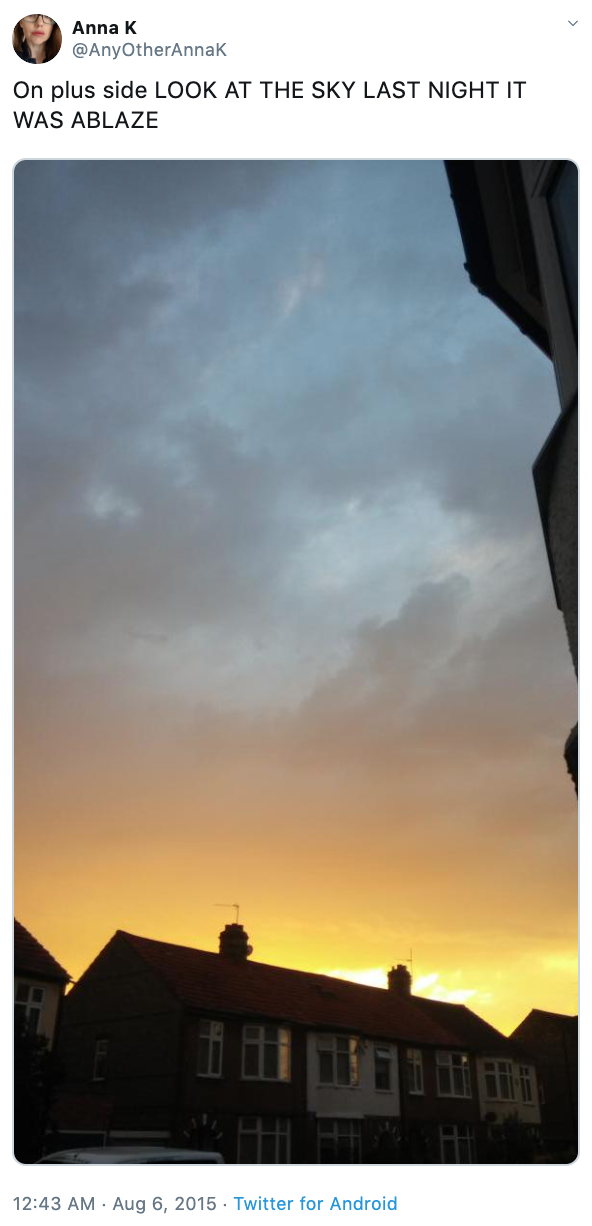
# Tổng quan về đề tài

Natural Language Processing – xử lý ngôn ngữ tự nhiên: là một phân ngành học của máy tính và đặc biệt là trí tuệ nhân tạo. Nó chủ yếu liên quan đến việc cung cấp cho máy tính khả năng xử lý dữ liệu được mã hóa bằng ngôn ngữ tự nhiên và do đó có liên quan chặt chẽ đến việc truy xuất dữ liệu, biễu diễn kiến thức và ngôn ngữ học tính toán, một phân ngành của ngôn ngữ học. Thông thường, dữ liệu được thu thập trong các tập hợp văn bản, sử dụng các phương pháp tiếp cận dựa trên quy tắc, thống kê hoặc dựa trên nơ-ron trong học máy và học sâu. Các nhiệm vụ chính trong xử lý ngôn ngữ tự nhiên là nhận dạng giọng nói, phân loại văn bản, hiểu ngôn ngữ tự nhiên và tạo ngôn ngữ tự nhiên.

Disaster Tweets: tweets là các mẩu tin nhỏ trong chức năng đọc, nhắn và cập nhật trong mạng xã hội Twitter bây giờ được gọi là X, còn disaster dịch theo nghĩa tiếng anh là thảm họa.

Mô tả: Twitter (hiện tại là X) đã trở thành một kênh giao tiếp quan trọng trong các tình huống khẩn cấp. Với sự phổ biến rộng rãi của điện thoại thông minh, mọi người có thể thông báo về một tình huống khẩn cấp mà họ đang chứng kiến theo thời gian thực. Vì lý do này, ngày càng có nhiều cơ quan quan tâm đến việc giám sát Twitter một cách tự động (ví dụ như các tổ chức cứu trợ thảm họa và các cơ quan tin tức)

Tuy nhiên, không phải lúc nào cũng rõ ràng rằng lời nói của một người có thực sự là đang thông báo về một thảm họa hay không. Hãy xem xét ví dụ sau đây:



Tác giả sử dụng từ “ABLAZE” một cách rõ ràng, nhưng ý của họ lại là ẩn dụ. Điều này rất dễ nhận ra đối với con người, đặc biệt khi có hình ảnh minh họa. Nhưng đối với máy móc thì lại khó nhận biết hơn.

Trong đề tài này nhóm sẽ xây dựng mô hình máy học để dự đoán xem các Tweets có liên quan đến thảm họa không

Tập dữ liệu đã được cuộc thi thu thập nhóm sẽ tận dụng tập dữ liệu có sãn để xây dựng mô hình

Để thực hiện được mô hình máy học thì nhóm sẽ liệt kê quy trình thực hiện xử lý NLP của nhóm sử dụng cho đề tài

1. Tiền xử lý văn bảng:

* Tokenization: tách văn bản từ tập dữ liệu
* Stop Words Removal: loại bỏ các từ thông dụng không có giá trị để lọc nhiễu
* Stemming: rút gọn từ về dạng gốc

1. TF – IDF (term frequency – inverse document frequency): tạo ra các đặc trưng số dựa trên tần suất từ để xác định mức độ quan trọng của từ
2. Feature Engineering: xây dựng các đặc trưng từ văn bản để làm đầu vào cho mô hình
3. Classification: sử dụng mô hình học máy để phân loại văn bản, trong đề tài của nhóm sử dụng LinearRegression (hồi quy tuyến tính) và dùng F1-score đánh giá hiệu quả (dựa trên yêu cầu đề tài)

**Đánh giá** (Evaluation)

F1 được tính theo công thức sau

*F1 =*  với *precision =*  và recall =

Chú thích:

* True Positive [TP] = dự đoán của bạn là 1 và giá trị thực tế cũng là 1 - bạn dự đoán là tích cực và điều đó đúng
* False Positive [FP] = dự đoán của bạn là 1 và giá trị thực tế cũng là 0 - bạn dự đoán là tích cực và điều đó sai
* False Nagative [FN] = dự đoán của bạn là 0 và giá trị thực tế cũng là 1 - bạn dự đoán là tiêu cực và điều đó sai

# Các nội dung đã thực hiện

## Phân tích và khám phá dữ liệu

* **Xác định các đặc trưng có trong tập dữ liệu:**

Gồm id, keyword, location, text và target trong tệp dữ liệu train.csv

* id: duy nhất lỗi tweets, để định danh riêng từng dòng dữ liệu
* keyword: từ khóa đặc trưng trong nội dung tweets, có thể liên quan tới disaster (có thảm họa). Một số dòng có thể không có dự liệu cho keyword
* location: thông tin của người đang tweets, có thể là tên thành phố, quốc gia hoặc một khu vực cụ thể. Một số dòng có thể không có dữ liệu cho location
* text: nội dung văn bản của tweets, đây là trường thông tin quan trọng nhất để mô hình có thể phân loại được tweets thuộc loại target (nhãn của tweets) nào
* target: nhãn của tweets, để cho biết có liên qua tới thảm họa không
* 1: tweets liên quan tới thảm họa (disaster)
* 0: tweets không liên quan tới thảm họa (non-disaster)

## Phân tích sự phân bố dữ liệu

* **Sự phân bố của disaster và non-Disaster ở đặc trưng target trong tệp dữ liệu**

**Tổng số phần tử:** Có tổng cộng 7613 ở đặc trưng target trong tập dữ liệu.

**Biến target:** Biến này có 2 giá trị:

**0:** Thường được hiểu là không xảy ra thảm họa (non-disaster).

**1:** Thường được hiểu là xảy ra thảm họa (disaster).

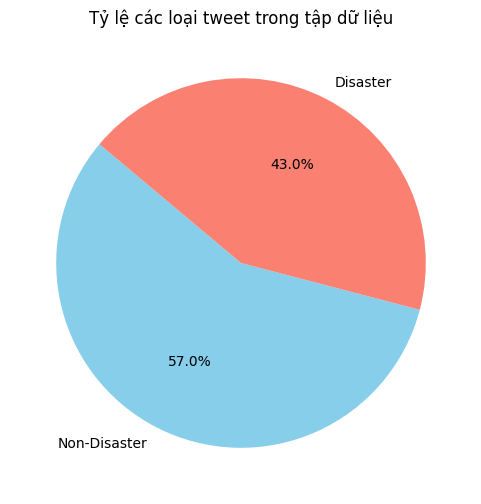
**Phân bố:**

**Giá trị 0 (không xảy ra sự kiện):** Chiếm 4342 quan sát.

**Giá trị 1 (xảy ra sự kiện):** Chiếm 3271 quan sát.

**Dữ liệu không cân bằng:** Số lượng quan sát thuộc lớp "không xảy ra sự kiện" nhiều hơn đáng kể so với lớp "xảy ra sự kiện". Tình trạng này thường gọi là dữ liệu không cân bằng (imbalanced data) trong học máy.

**Ý nghĩa:** Điều này có thể cho thấy rằng trong quá trình thu thập dữ liệu, các sự kiện không xảy ra thường phổ biến hơn so với các sự kiện xảy ra. Hoặc có thể do bản chất của vấn đề nghiên cứu, các sự kiện không xảy ra có xác suất xuất hiện cao hơn.

****

* **Sự phân bố độ dài của nội dung ở đặc trưng text trong tệp dữ liệu**

Hai biểu đồ trên cho thấy phân bố độ dài của các tweet được phân loại là "Disaster" (thảm họa) và "Non Disaster" (không phải thảm họa). Đường cong mượt mà chồng lên các cột biểu đồ cho ta thấy một ước tính về phân phối xác suất của độ dài tweet.

So sánh giữa hai loại nhãn disaster và non-disaster ở đặc trưng target phụ thuộc vào độ dài nội dung của đặc trưng text

**Độ dài trung bình:** Cả hai nhóm tweet đều có độ dài trung bình tương đối gần nhau, tập trung chủ yếu ở khoảng từ 80 đến 120 ký tự. Tuy nhiên, có vẻ như nhóm "Disaster" có độ dài trung bình hơi lớn hơn một chút so với nhóm "Non Disaster".

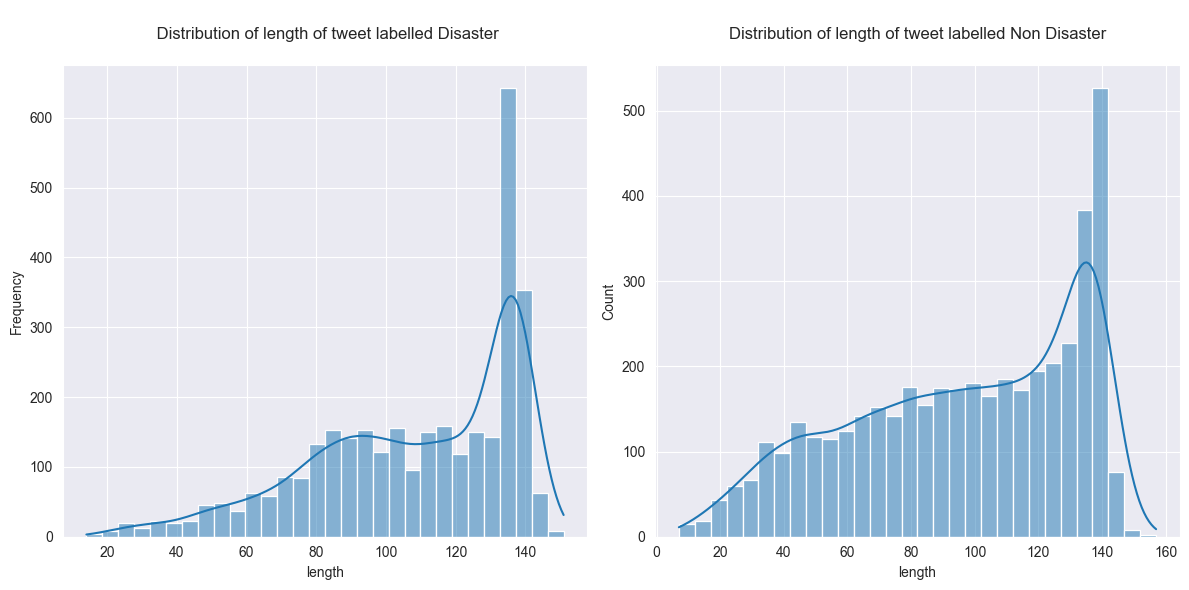
**Độ lệch chuẩn:** Cả hai nhóm đều có độ phân tán về độ dài khá lớn, nghĩa là độ dài của các tweet trong cùng một nhóm có thể khác nhau rất nhiều.

**Hình dạng phân phối:** Cả hai phân phối đều có dạng lệch phải (right-skewed), nghĩa là có một số ít tweet rất dài kéo dài phân phối về phía bên phải. Tuy nhiên, nhóm "Disaster" có vẻ như có phần đuôi phân phối dài hơn một chút so với nhóm "Non Disaster".

**Tweet về thảm họa thường dài hơn:** Việc nhóm "Disaster" có độ dài trung bình lớn hơn một chút và đuôi phân phối dài hơn có thể gợi ý rằng người dùng thường cần nhiều từ hơn để mô tả chi tiết một sự kiện thảm họa, bao gồm các chi tiết về địa điểm, thời gian, mức độ nghiêm trọng, v.v.

**Ảnh hưởng đến phân loại:** Sự khác biệt về độ dài này có thể được sử dụng như một đặc trưng để phân loại tweet. Các mô hình phân loại có thể học được rằng các tweet dài hơn có khả năng cao hơn thuộc về nhóm "Disaster".

**Giới hạn của phân tích:** Tuy nhiên, cần lưu ý rằng độ dài chỉ là một trong nhiều yếu tố ảnh hưởng đến việc phân loại tweet. Các từ khóa, ngữ cảnh, cảm xúc biểu hiện trong tweet cũng đóng vai trò quan trọng.



## Xử lý tiền văn bản

### Tách văn bản từ tập dữ liệu – Tokenization

Tokenization là bước phân tách văn bản thành các từ hoặc cụm từ nhỏ hơn gọi là “tokens.” Mỗi token thường đại diện cho một từ đơn lẻ trong câu, mặc dù có thể bao gồm cả các dấu hiệu đặc biệt hoặc biểu tượng nếu được cấu hình.

**Công cụ Tokenization**: Hàm sử dụng nltk.TweetTokenizer - bộ tokenizer đặc biệt dành cho dữ liệu mạng xã hội như Twitter. Bộ công cụ này có thể loại bỏ @mentions, giữ hoặc không giữ nguyên kiểu chữ và giảm độ dài các từ lặp lại.

**Giữ nguyên kiểu chữ** (preserve\_case) giúp giữ nguyên hoặc chuyển tất cả các từ về dạng chữ thường.

**Giảm độ dài** (reduce\_len) loại bỏ bớt các ký tự lặp lại để ngăn hiện tượng kéo dài ký tự như “heyyy” thành “hey.”

**Xóa bỏ @mentions** (strip\_handles) để bỏ qua các tên người dùng, giảm nhiễu cho tập dữ liệu.

**Quá trình Tokenization**:

* Đầu tiên, hàm lặp qua từng hàng trong DataFrame, sau đó thực hiện tokenization cho từng cột trong keys ('text', 'keyword', 'location').
* Với mỗi cột được xác định, hàm chuyển đổi nội dung văn bản thành một chuỗi tokens (danh sách các từ).
* Tất cả tokens từ các cột khác nhau sẽ được gộp lại thành một danh sách duy nhất cho mỗi hàng trong DataFrame, lưu vào danh sách output.

### Loại bỏ các tử thông dụng không có giá trị để lọc nhiễu – Stop Words Removal

Trong dữ liệu văn bản, một số từ thường xuất hiện nhiều nhưng không cung cấp nhiều thông tin ngữ nghĩa, gọi là “stopwords” (từ dừng). Chúng thường bao gồm các từ phổ biến như “the,” “is,” “at,” v.v.

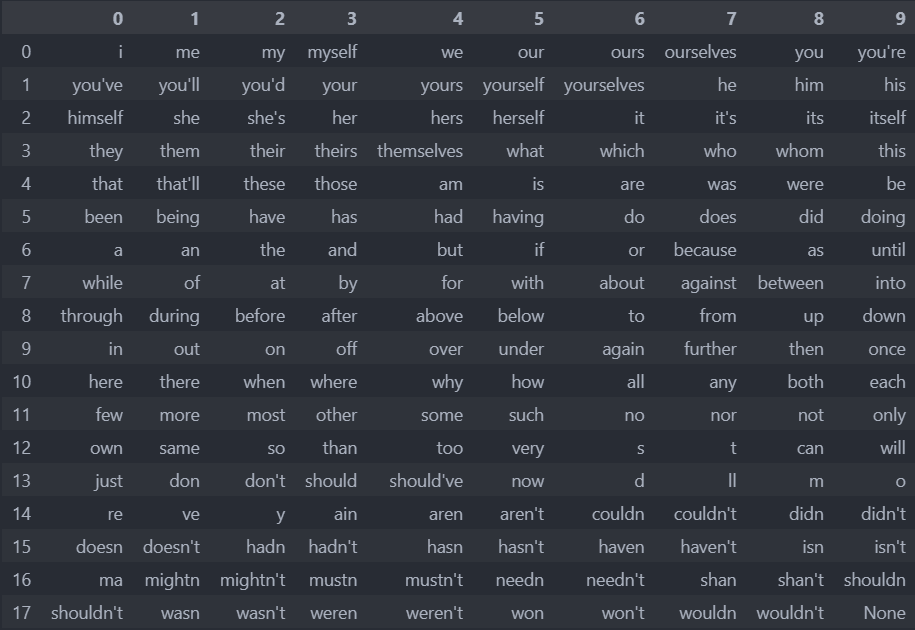
**Tạo danh sách stopwords**: Hàm sử dụng tập từ dừng chuẩn của NLTK với tiếng Anh, sau đó thêm 'nan' (chuỗi có thể xuất hiện khi dữ liệu thiếu) vào danh sách này.

**Loại bỏ stopwords**: Nếu tham số use\_stopwords được đặt là True, hàm lọc bỏ các tokens nằm trong tập stopwords này:

* Đầu tiên, mỗi token được chuyển sang chữ thường để có thể khớp chính xác với danh sách stopwords (danh sách này cũng ở dạng chữ thường).
* Điều kiện lọc bổ sung là các tokens phải có độ dài ít nhất 2 ký tự để loại bỏ các ký tự đơn lẻ và giảm thiểu nhiễu trong dữ liệu.

**Kết quả của bước loại bỏ stopwords**: Sau bước này, danh sách tokens đã được làm sạch, chỉ giữ lại những từ có ý nghĩa, giúp giảm kích thước dữ liệu đầu vào cho các bước xử lý tiếp theo.

**Bảng danh sách stopword của NLTK**

****

### Rút gọn từ về dạng gốc - Stemming

Stemming là bước xử lý văn bản giúp giảm các từ khác nhau về cùng một gốc chung. Ví dụ, “running” và “runs” sẽ được chuyển về “run.” Điều này giúp giảm số lượng từ trong tập dữ liệu, đồng thời giữ lại được ý nghĩa ngữ nghĩa của các từ liên quan.

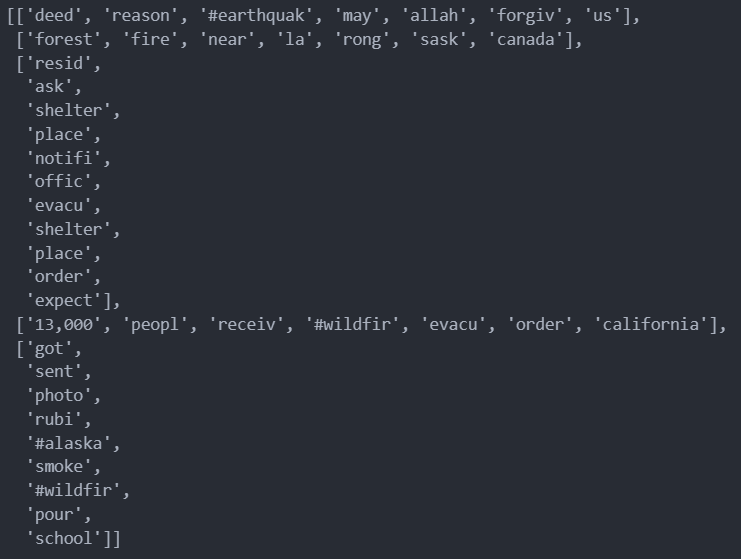
**Công cụ Stemming**: Hàm sử dụng PorterStemmer từ NLTK, một thuật toán stemming phổ biến và hiệu quả. PorterStemmer chuyển các từ về gốc bằng cách áp dụng các quy tắc cắt gọt chữ cái trong từ. Ví dụ, từ “running” sẽ thành “run,” và “happily” sẽ thành “happi.”

**Thực hiện Stemming**:

* Nếu tham số stemmer được đặt là True, hàm sẽ áp dụng stemming lên mỗi token đã qua bước loại bỏ stopwords.
* Sau khi stemming, danh sách tokens được thêm vào danh sách output, đại diện cho một hàng dữ liệu đã qua xử lý.

**Kết quả của bước Stemming**: Các từ có cùng gốc ngữ nghĩa được gom lại, giảm bớt kích thước từ vựng trong dữ liệu đầu vào. Điều này giúp cải thiện hiệu quả của các mô hình học máy trong việc nhận diện các mẫu trong dữ liệu văn bản.

**Kết quả 5 mẫu gần nhất sau khi Tiền xử lý văn bản**



## Tạo ra các đặc trưng dựa trên tần xuất từ để xác định mức quan trọng của từ - TF – IDF (Term frequency – Inverse Document Frequency)

**Giới thiệu TF – IDF (Term frequency – Inverse Document Frequency):**

* Là một chỉ số thống kê phản ánh tầm quan trọng của một từ trong một văn bản so với một tập hợp hoặc ngữ liệu văn bản. Chỉ số này thường được sử dụng dưới dạng trọng số trong các hệ thống tìm kiếm thông tin, khai thác văn bản và mô hình hóa người dùng.
* Giá trị của TF-IDF tăng lên tỷ lệ thuận với số lần xuất hiện của một từ trong tài liệu, đồng thời được điều chỉnh bởi số lượng tài liệu trong kho ngữ liệu chứa từ đó. Điều này giúp cân nhắc thực tế rằng một số từ thường xuất hiện nhiều hơn trong ngữ liệu tổng thể. TF-IDF hiện là một trong những phương pháp tính trọng số phổ biến nhất, và theo một cuộc khảo sát thực hiện vào năm 2015, 83% các hệ thống khuyến nghị dựa trên văn bản trong các thư viện số đã áp dụng TF-IDF.

**Trong dó**

* **TF – term frequency:** tần số xuất hiện của một từ trong một văn bản
  + Công thức:

**tf(t, d) =**

* + Chú ý
    - Thương của số lần xuất hiện một từ trong văn bản và số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản đó (giá trị 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 trong bất ỳ văn bản
* **IDF – inverse document frequency:** tần số nghịch của một từ trong tập văn bản (corpus)
  + Công thức: tính **IDF** để giảm giá trị của những từ phổ biến. Mỗi từ chỉ có một giá trị **IDF** duy nhất trong tập văn bản

**idf(t, D) = log**

* + |D|: tổng số văn bản cho tập D
  + : số văn bản chứa từ nhất định, với điều kiện t xuất hiện trong văn bản d(tf(t, d) 0). Nếu từ đó không xuất hiện ở bất cứ một văn bản nào trong tập mẫu thì mẫu số sẽ bằng 0 từ đó phép chia không hợp lẹ, vì thế người ta thay bằng biểu 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.

* **Đặc biết giá trị TF – IDF:**
  + Công thức

**tfidf(t, d, D) = tf(t, d) x idf(t, D)**

* 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).

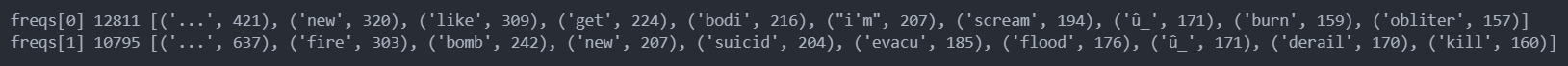
**Các bước thực hiện**

* Tính tần suất xuất hiện của cách từ trong từng đặc trưng target thông qua hàm word\_frequencies khi
  + disaster là target = 1
  + non-disaster là target = 0

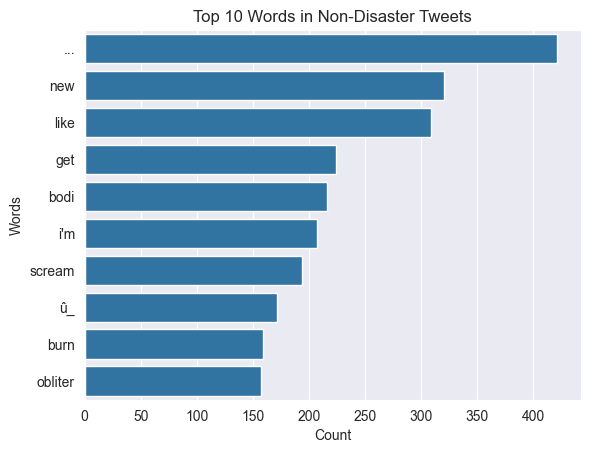
Sau dó thực hiện

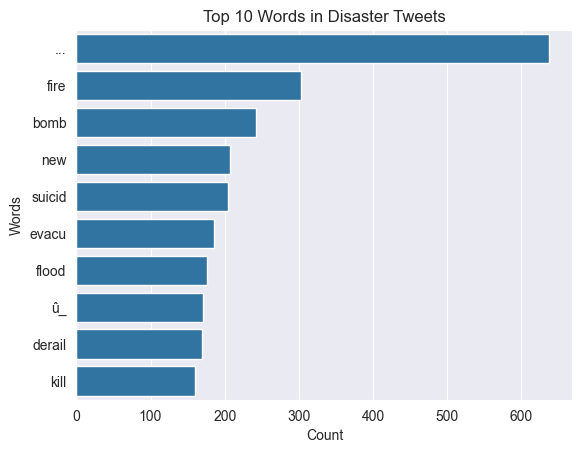
* + Xử lý tiền văn bản: thông qua hàm tokenize\_df mã chuyển đổi văn bản của từng nhóm target thành danh sách các từ. Kết quả là một cấu trúc dữ liệu dạng từ điển, trong đó các từ được lưu trữ dưới dạng danh sách.
  + Tính tần suất từ: Sử dụng Counter để đếm tần suất của từng từ trong cả hai nhóm target. Kết quả là một Counter chứa số lần xuất hiện của các từ trong từng nhóm tweets.
  + Trả giá trị về tần suất từ: trả về tần suất từ của hai nhóm target là 0 và 1, lưu trữ trong biến freqs. Khi in ra, ta có thể xem các từ phổ biến nhất trong mỗi nhóm, giúp nắm bắt nhanh những từ nào xuất hiện thường xuyên nhất trong từng nhóm disaster và non-disaster.

**Các giá trị trả về tần suất từ**

****

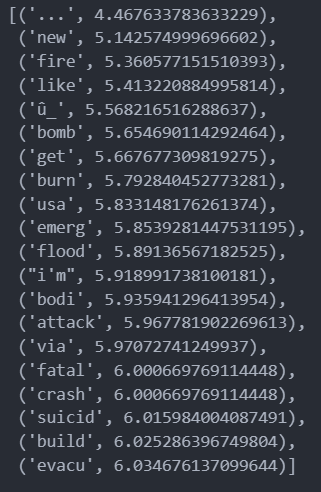
**Biểu đồ biểu thị tần suất từ của disaster và non-disaster**

****

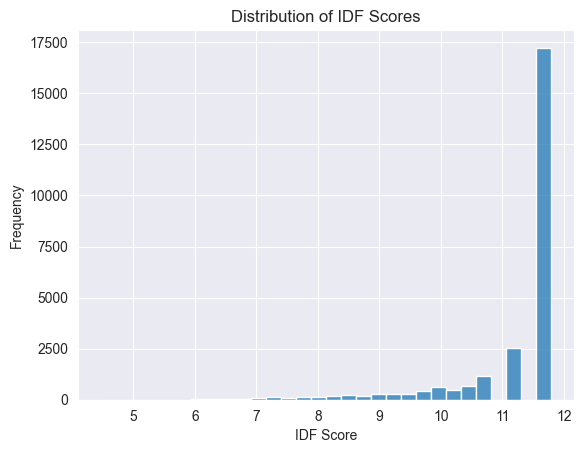
****

* Thực hiện tính TF – IDF cho tập dữ liệu
  + Áp dụng các công thức tử TF và IDF ở mục giới thiệu vào chương trình
    - **Tokenize và tính IDF**: Sử dụng Counter để tính tần suất xuất hiện của từ trên tất cả tài liệu. Sau đó, áp dụng công thức trên để tính IDF cho từng từ.
    - **Tokenize và tổng hợp IDF**: Kết hợp các từ từ df\_train và df\_test, sau đó gọi hàm inverse\_document\_frequency để tính IDF cho các từ này.
  + Bằng cách kết hợp cả tần suất từ (TF) và độ độc đáo của từ (IDF), TF-IDF giúp giảm thiểu tác động của các từ phổ biến, chỉ giữ lại các từ quan trọng cho phân loại nội dung. Trong bài toán phân loại tweet, TF-IDF sẽ giúp mô hình học được các từ có tính phân biệt cao giữa hai nhóm disaster và non-disaster

**Kết quả tính giá trị TF-IDF**



**Biểu đồ hiển thị giá trị TF-IDF**

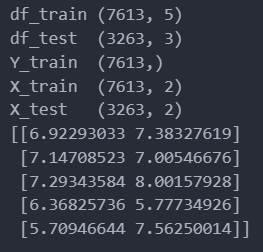
****

## Xây dựng các đặc trưng từ văn bản để làm đầu vào cho mô hình – Feature Engineering

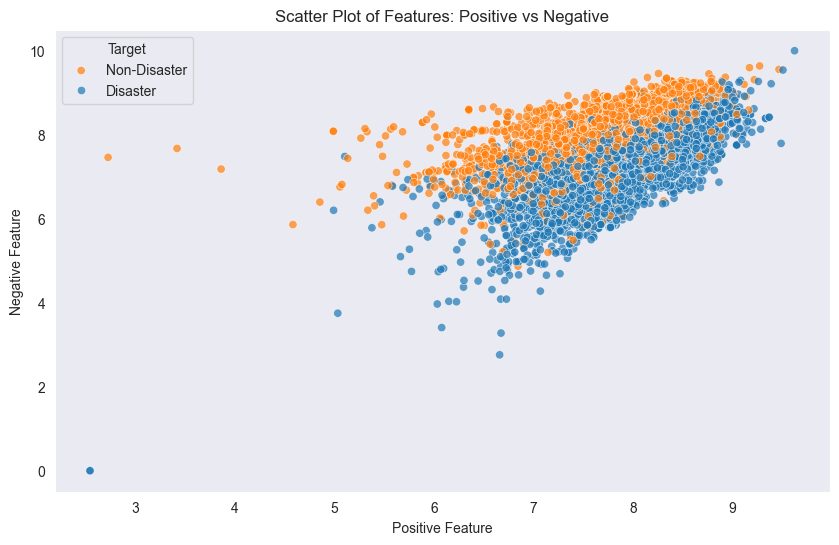
Feature Engineering là một bước quan trọng trong quy trình xây dựng mô hình học máy, đặc biệt khi làm việc với dữ liệu văn bản. Mục tiêu của bước này là trích xuất các đặc trưng có thể giúp cải thiện khả năng dự đoán của mô hình. Trong đoạn mã dưới đây, chúng tôi đã xây dựng các đặc trưng dựa trên tần suất từ khóa kết hợp với trọng số TF-IDF (Term Frequency-Inverse Document Frequency) để phân loại các tweet thành hai nhóm: có thảm họa (disaster) và không có thảm họa (no disaster).

* **Tokenization**: Sử dụng hàm tokenize\_df để tách các văn bản trong DataFrame thành các token (từ).
* **Tính tần suất từ và IDF**: Với mỗi token trong danh sách các token, hàm sẽ tính toán tần suất từ cho cả hai nhóm mục tiêu (có thảm họa và không có thảm họa) và kết hợp với trọng số IDF nếu tham số use\_idf được bật.
* **Tính toán đặc trưng**: Đối với mỗi văn bản, hàm sẽ tính toán một vectơ đặc trưng gồm hai giá trị:
  + positive: Tổng tần suất từ của các từ trong nhóm có thảm họa, tính theo IDF.
  + negative: Tổng tần suất từ của các từ trong nhóm không có thảm họa, cũng tính theo IDF.

**Kết quả sau khi xây dựng**

****

**Biểu đồ biểu hiện giữa hai nhóm đặc trưng**

****

## Sử dụng mô hình học máy để phân loại văn bản – Classification

Dữ liệu đầu vào cho bài toán này thường bao gồm các tweet (văn bản) và nhãn tương ứng. Mô hình sẽ học từ dữ liệu huấn luyện để xác định xem một tweet mới có thuộc về nhóm disaster hay non-disaster hay không.

Các bước thực hiện

* Tính tần suất từ vựng
  + Mục đích: Xác định tần suất xuất hiện của các từ trong tập dữ liệu huấn luyện
  + Giúp tạo ra một từ điển các từ và số lần xuất hiện của chúng. Từ này sẽ được sử dụng để rút trích các đặc trưng cho mô hình
* Trích xuất nhãn mục tiêu
  + Nhãn mục tiêu được lấy từ cột target trong DataFrame, trong đó giá trị 1 thể hiện tweet là diaster và 0 thể hiện tweet là non-disaster.
* Trích xuất đặc trưng
  + Các đặc trưng cho tập dữ liệu huấn luyện được trích xuất dựa trên tần suất từ vựng. Điều này có thể bao gồm các đặc trưng như tần suất xuất hiện của từng từ trong mỗi tweet.
  + Tương tự, các đặc trưng cho tập dữ liệu kiểm tra cũng được trích xuất
* Huấn luyện mô hình:
  + Sử dụng thuât toán hồi quy tuyến tính (LinearRegression)
  + Lý do chọn dùng thuật toán hồi quy tuyến tính cho đề tài
    - Mô hình đơn giản và dễ hiểu
* **Cấu trúc đơn giản**: Hồi quy tuyến tính là một trong những thuật toán học máy đơn giản nhất. Điều này giúp cho việc hiểu, giải thích và triển khai mô hình dễ dàng hơn, đặc biệt là cho những người mới bắt đầu trong lĩnh vực học máy và xử lý ngôn ngữ tự nhiên.
* **Giải thích được kết quả**: Một trong những ưu điểm lớn nhất của hồi quy tuyến tính là khả năng giải thích các kết quả dự đoán. Mỗi hệ số trong mô hình tương ứng với tầm ảnh hưởng của một từ cụ thể đến nhãn mục tiêu. Điều này rất quan trọng trong các ứng dụng như phân tích văn bản, nơi việc hiểu rõ nguyên nhân phân loại là cần thiết.
  + - Đặc điểm của bài toán
* **Dữ liệu nhãn nhị phân (binary classification)**: Các tweet cần được phân loại thành hai nhóm rõ ràng: disaster (1) và **non-disaster** (0). Hồi quy tuyến tính có thể được áp dụng cho bài toán phân loại nhị phân thông qua việc dự đoán xác suất và sau đó phân loại dựa trên ngưỡng (threshold).
* **Số lượng đặc trưng lớn**: Tweet có thể chứa nhiều từ khác nhau, dẫn đến việc số lượng đặc trưng rất lớn. Hồi quy tuyến tính có thể xử lý tốt các đặc trưng này, miễn là có đủ dữ liệu huấn luyện.
  + - Hiệu suất tốt trong cách bài toán cơ bản
* **Mô hình cơ bản nhưng hiệu quả**: Trong nhiều trường hợp, hồi quy tuyến tính có thể cung cấp một điểm khởi đầu tốt cho các bài toán phân loại văn bản. Dù không phải là lựa chọn tốt nhất trong mọi tình huống, nó vẫn có thể cho ra những dự đoán khá chính xác nếu dữ liệu được chuẩn bị tốt.
* **Chống lại overfitting**: Hồi quy tuyến tính ít có khả năng overfitting hơn so với các mô hình phức tạp hơn, đặc biệt khi dữ liệu huấn luyện không đủ lớn. Điều này giúp mô hình hoạt động tốt hơn trên dữ liệu chưa thấy.
  + - Khả năng tính toán nhanh chóng: Hồi quy tuyến tính thường yêu cầu ít thời gian tính toán hơn so với các mô hình phức tạp hơn. Điều này rất quan trọng khi xử lý lượng lớn dữ liệu như tweet, nơi mà thời gian và tài nguyên là yếu tố cần xem xét.
    - Tích hợp dễ dàng với nhiều thuật toán khác: Hồi quy tuyến tính có thể được sử dụng như một bước đầu tiên trong quy trình xây dựng mô hình. Nếu hồi quy tuyến tính không đạt hiệu suất mong muốn, người dùng có thể dễ dàng chuyển sang các thuật toán phức tạp hơn như hồi quy logistic, cây quyết định, hoặc mạng nơ-ron.
* Phân tích ưu điểm và nhược điểm
  + - Ưa điểm:
* **Dễ hiểu và triển khai**: Đơn giản trong việc lập trình và hiệu chỉnh.
* **Giải thích rõ ràng**: Cung cấp insight về cách mà các từ ảnh hưởng đến dự đoán.
* **Thích hợp cho dữ liệu lớn**: Có thể xử lý hiệu quả với nhiều đặc trưng mà không gặp nhiều vấn đề về tính toán.
* Nhược điểm:
* **Giới hạn trong mối quan hệ phi tuyến**: Hồi quy tuyến tính không thể nắm bắt được các mối quan hệ phi tuyến giữa các đặc trưng và nhãn. Điều này có thể dẫn đến kết quả kém khi dữ liệu không tuân theo giả định tuyến tính.
* **Nhạy cảm với outlier**: Các điểm dữ liệu cực đoan có thể ảnh hưởng lớn đến kết quả dự đoán của hồi quy tuyến tính.
* Dự đoán
  + Mô hình đã huấn luyện được sử dụng để dự đoán nhãn cho các tweet trong tập kiểm tra.
  + Kết quả dự đoán được làm tròn và chuyển đổi thành kiểu số nguyên, phản ánh nhãn mục tiêu rời rạc.

## Đánh giá

Theo trang chủ cuộc thi Kaggle thì F1 được tính theo công thức sau

*F1 =*  với *precision =*  và recall =

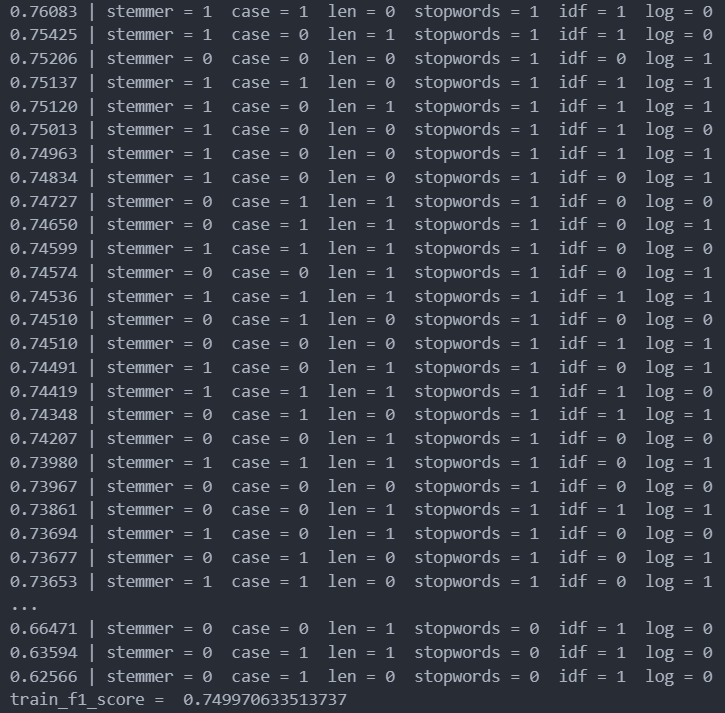
Chú thích:

* True Positive [TP] = dự đoán của bạn là 1 và giá trị thực tế cũng là 1 - bạn dự đoán là tích cực và điều đó đúng
* False Positive [FP] = dự đoán của bạn là 1 và giá trị thực tế cũng là 0 - bạn dự đoán là tích cực và điều đó sai
* False Nagative [FN] = dự đoán của bạn là 0 và giá trị thực tế cũng là 1 - bạn dự đoán là tiêu cực và điều đó sai

Các thực hiện

* Tính toán điểm F1 cho mô hình phân loại. Sử dụng phương pháp k-fold cross-validation để đánh giá độ chính xác của mô hình trên tập dữ liệu huấn luyện.
  + **Chia dữ liệu**: Dữ liệu được chia thành các tập con huấn luyện và kiểm tra
  + **Dự đoán**: sử dụng thuật toán hồi quy tuyến tính để dự đoán nhãn cho tập kiểm tra.
  + **Tính điểm F1**: Điểm F1 được tính bằng cách so sánh nhãn dự đoán với nhãn thực tế. Tổng điểm F1 từ mỗi lần lặp được tính và chia cho số lần chia để có điểm trung bình
* Tìm kiếm và đánh giá các siêu tham số tốt nhất cho mô hình phân loại để tối ưu hóa điểm F1.
  + **Kết hợp siêu tham số**: Sử dụng hàm product để tạo ra tất cả các kết hợp có thể của các siêu tham số (như stemmer, preserve\_case, v.v.).
  + **Tính điểm F1 cho từng tổ hợp**: Mỗi tổ hợp siêu tham số được đánh giá bằng cách gọi hàm tính toán điểm F1 cho mô hình phân loại, và kết quả được lưu vào biến để xuất ra.
  + **Song song hóa**: Sử dụng Parallel để tính toán song song, giúp tiết kiệm thời gian trong quá trình tìm kiếm.

**Kết quả sau khi tính F1**



# Các kết quả đã đạt được (chưa đạt được)

## Kết quả đạt được

**Phân chia công việc hợp lý**

Nhóm đã tổ chức phân công công việc rõ ràng, mỗi thành viên phụ trách một phần nhất định như xử lý dữ liệu, xây dựng mô hình, đánh giá kết quả, và viết báo cáo. Điều này giúp tận dụng được điểm mạnh của từng thành viên và đảm bảo tiến độ công việc.

**Nâng cao kiến thức và kỹ năng**

Qua dự án này, cả nhóm đã có cơ hội thực hành các kỹ thuật hồi quy, xử lý dữ liệu nâng cao như loại bỏ outliers và xử lý dữ liệu thiếu, cũng như học cách sử dụng công cụ như Python, các thư viện machine learning, và công cụ phân tích dữ liệu.

**Kết quả mô hình khả quan**

Nhờ sự hợp tác và đóng góp của các thành viên, nhóm đã xây dựng được một mô hình dự đoán giá nhà đạt độ chính xác tương đối cao. Ridge Regression là mô hình nổi bật nhất trong số các mô hình thử nghiệm, giúp nhóm đưa ra dự đoán khá sát với giá trị thực tế.

**Tăng cường kỹ năng giao tiếp và báo cáo**

Nhóm đã học cách trình bày và báo cáo kết quả khoa học rõ ràng, từ cách giải thích các chỉ số đánh giá đến việc so sánh hiệu suất của các mô hình khác nhau. Điều này giúp đảm bảo rằng kết quả nghiên cứu dễ hiểu và có thể áp dụng vào thực tiễn.

## Các kết quả chưa đạt được

**Khó khăn trong xử lý và tích hợp ý kiến**

Trong quá trình làm việc nhóm, đã có những lúc ý kiến của các thành viên khác nhau về cách xử lý dữ liệu, cách chọn mô hình hoặc đánh giá kết quả. Điều này đôi khi gây mất thời gian và ảnh hưởng đến tiến độ công việc.

**Thiếu kinh nghiệm trong tối ưu hóa mô hình**

Việc xử lý dữ liệu là một công việc tương đối mới mẻ với các thành viên trong nhóm nên các thành viên còn thiếu kinh nghiệm trong việc tinh chỉnh các tham số của mô hình (hyperparameter tuning), dẫn đến việc một số mô hình chưa đạt hiệu quả cao nhất. Điều này cũng có thể là nguyên nhân làm tăng sai số dự đoán.

**Chưa tối ưu hóa hiệu quả xử lý dữ liệu lớn**

Một số công đoạn xử lý dữ liệu tốn nhiều thời gian do lượng dữ liệu lớn, và nhóm chưa có kinh nghiệm tối ưu hóa hiệu quả tính toán trong những trường hợp này. Điều này dẫn đến việc mô hình đôi khi chạy chậm và gây trở ngại trong việc kiểm tra kết quả nhanh chóng.

# Những khó khăn, vướng mắc gặp phải trong quá trình thực hiện

**Khó khăn về xử lý dữ liệu**

Dữ liệu thiếu và nhiễu: Trong dữ liệu về giá nhà, thường có nhiều giá trị bị thiếu (missing values) và dữ liệu không hợp lệ (outliers). Việc xác định và xử lý những giá trị này là một thách thức lớn, vì có thể ảnh hưởng trực tiếp đến độ chính xác của mô hình.

Biến đổi dữ liệu: Một số biến số có độ lệch lớn (skewed) hoặc không tuân theo phân phối chuẩn, gây khó khăn khi áp dụng các mô hình hồi quy. Nhóm cần thực hiện các phép biến đổi như log transformation, Box-Cox để cải thiện sự phân phối của các biến, nhưng điều này yêu cầu hiểu biết sâu về phương pháp và thời gian thử nghiệm.

**Khó khăn trong việc phối hợp làm việc nhóm**

Khác biệt về kỹ năng và kinh nghiệm: Mỗi thành viên trong nhóm có mức độ hiểu biết và kinh nghiệm khác nhau về machine learning và lập trình, điều này đôi khi gây khó khăn trong việc phân chia công việc và hoàn thành đúng tiến độ.

Giao tiếp và thống nhất ý kiến: Việc thống nhất ý kiến khi gặp những lựa chọn quan trọng như chọn mô hình, cách xử lý dữ liệu hoặc cách đánh giá kết quả là một thách thức. Đôi khi, việc bất đồng ý kiến có thể làm chậm tiến độ công việc và ảnh hưởng đến tinh thần làm việc của nhóm.

**Hạn chế về kiến thức chuyên môn**

Thiếu kiến thức về hồi quy và phân tích thống kê: Một số thành viên có thể chưa nắm vững các khái niệm về hồi quy, phân phối chuẩn, outliers, hoặc kỹ thuật biến đổi dữ liệu, dẫn đến khó khăn trong việc lựa chọn phương pháp phù hợp và phân tích kết quả.

Hiểu biết về đặc thù của dữ liệu giá nhà: Để xây dựng mô hình dự đoán giá nhà tốt, nhóm cần có kiến thức về thị trường bất động sản, như các yếu tố ảnh hưởng đến giá nhà, nhằm chọn ra các đặc trưng (features) quan trọng. Việc thiếu hiểu biết này có thể làm giảm tính chính xác của mô hình.

# Những kinh nghiệm thu được sau khi hoàn thành bài tập lớn

**Kinh nghiệm về xử lý và phân tích dữ liệu**

Kỹ năng xử lý dữ liệu thiếu và bất thường: Nhóm đã học được cách xác định và xử lý các giá trị thiếu (missing values) và outliers, chẳng hạn như sử dụng trung vị (median) để điền dữ liệu thiếu khi dữ liệu có sự phân phối lệch. Kỹ năng này rất quan trọng để đảm bảo dữ liệu sạch, giúp cải thiện độ chính xác của mô hình.

Biến đổi và chuẩn hóa dữ liệu: Kinh nghiệm trong việc áp dụng các phép biến đổi như log hoặc Box-Cox để làm cho dữ liệu tuân theo phân phối chuẩn. Điều này giúp nhóm hiểu hơn về sự ảnh hưởng của phân phối dữ liệu đến mô hình và hiệu suất dự đoán.

**Kinh nghiệm về xây dựng và tối ưu hóa mô hình**

Hiểu rõ ưu và nhược điểm của các mô hình hồi quy: Nhóm đã hiểu sâu hơn về các mô hình hồi quy như Linear Regression, Polynomial Regression, Lasso, và Ridge Regression. Qua việc thử nghiệm và đánh giá các mô hình này, nhóm nắm bắt được khi nào mô hình bị overfitting hoặc underfitting, và biết cách điều chỉnh tham số để tối ưu hóa mô hình.

**3. Kinh nghiệm làm việc nhóm và quản lý dự án**

Phân chia công việc và tận dụng kỹ năng từng thành viên: Qua đề tài này, nhóm học được cách phân chia công việc hiệu quả dựa trên điểm mạnh và kỹ năng của từng thành viên. Điều này không chỉ giúp hoàn thành công việc đúng tiến độ mà còn cải thiện chất lượng của từng phần công việc.

Giao tiếp và hợp tác: Nhóm đã rút ra bài học về tầm quan trọng của giao tiếp liên tục và rõ ràng trong việc thống nhất ý kiến và giải quyết xung đột. Việc học cách lắng nghe và tôn trọng ý kiến của từng thành viên giúp cải thiện không khí làm việc và tăng cường tinh thần làm việc nhóm.

**Kinh nghiệm sử dụng công cụ và thư viện**

Sử dụng thư viện và công cụ lập trình: Nhóm đã làm quen và thành thạo hơn trong việc sử dụng các thư viện như Pandas, Numpy, Scikit-learn để xử lý dữ liệu và xây dựng mô hình. Nhóm cũng học cách sử dụng Matplotlib và Seaborn để trực quan hóa dữ liệu, giúp hiểu rõ hơn về các đặc tính của dữ liệu.

Làm quen với môi trường làm việc và tài nguyên hạn chế: Khi phải làm việc với tài nguyên phần cứng giới hạn hoặc không gian lưu trữ dữ liệu hạn chế, nhóm đã học cách tối ưu hóa mã và sử dụng tài nguyên hợp lý, đồng thời chuẩn bị sẵn sàng cho các hạn chế khi làm việc với dữ liệu lớn.

**Hiểu biết sâu hơn về dữ liệu bất động sản**

Xác định yếu tố ảnh hưởng đến giá nhà: Nhóm đã hiểu rõ hơn về các yếu tố quan trọng ảnh hưởng đến giá nhà, như diện tích, chất lượng xây dựng, và năm xây dựng. Kiến thức này giúp nhóm xác định và chọn lọc các đặc trưng (features) có giá trị để cải thiện độ chính xác của mô hình.

Hiểu về tính phi tuyến và sự phức tạp của dữ liệu giá nhà: Qua việc phân tích dữ liệu giá nhà, nhóm nhận thấy tính phi tuyến của một số biến số, và hiểu rằng không phải mọi quan hệ đều có thể được giải thích bằng hồi quy tuyến tính. Điều này giúp nhóm có cách nhìn sâu sắc hơn khi chọn mô hình và thực hiện phân tích.

# Kết luận (và tự đánh giá theo thang điểm từ 0 đến 10)

Qua quá trình nghiên cứu và thực hiện đề tài dự đoán giá nhà bằng các phương pháp hồi quy, nhóm đã có được những hiểu biết sâu sắc hơn về quy trình phân tích và xử lý dữ liệu, cũng như áp dụng thành công các mô hình học máy để giải quyết bài toán thực tế. Việc so sánh các phương pháp hồi quy tuyến tính, hồi quy đa biến, hồi quy bậc hai, Lasso và Ridge giúp nhóm hiểu rõ hơn về ưu, nhược điểm của từng mô hình, đặc biệt trong việc tránh overfitting và underfitting.

Kết quả thu được cho thấy một số mô hình có khả năng dự đoán tốt với độ chính xác cao, nhưng cũng có những mô hình chưa đạt kết quả như mong đợi do một số yếu tố như outliers, đa cộng tuyến và phân phối dữ liệu không chuẩn. Nhóm đã tìm ra các phương pháp xử lý hiệu quả, bao gồm điều chỉnh đặc trưng, biến đổi dữ liệu và áp dụng kỹ thuật Regularization, giúp nâng cao chất lượng mô hình.

Nhìn chung, nhóm đã hoàn thành tốt các mục tiêu đề ra, tích lũy được nhiều kinh nghiệm bổ ích và có khả năng tự tin hơn khi thực hiện các dự án phân tích dữ liệu trong tương lai.

**Nhóm tự đánh giá**

Hiểu biết về dữ liệu và xử lý dữ liệu: 9/10

Nhóm đã có sự chuẩn bị tốt trong việc phân tích và làm sạch dữ liệu, xử lý các giá trị ngoại lai, lấp đầy dữ liệu thiếu và thực hiện các biến đổi cần thiết, giúp mô hình hoạt động ổn định hơn.

Kỹ năng xây dựng và tối ưu hóa mô hình: 9/10

Các mô hình đã được tối ưu và so sánh kỹ lưỡng, đặc biệt là việc áp dụng Box-Cox để điều chỉnh phân phối và sử dụng Regularization giúp giảm thiểu overfitting, cho thấy nhóm có sự hiểu biết sâu sắc trong tối ưu hóa mô hình.

Hiểu biết về lý thuyết hồi quy và học máy: 10/10

Nhóm nắm vững lý thuyết và các chỉ số đánh giá hiệu quả mô hình như RMSE và R-squared, giúp giải thích kết quả dự đoán rõ ràng và thuyết phục hơn.

Kỹ năng làm việc nhóm và phân chia công việc: 10/10

• Phân chia công việc hiệu quả, đảm bảo tiến độ và phối hợp nhịp nhàng là một trong những điểm mạnh của nhóm. Mỗi thành viên đều đóng góp tích cực vào từng phần của đề tài.

Khả năng trình bày và báo cáo: 9/10

Báo cáo và trình bày được chuẩn bị kỹ càng, có hệ thống, giúp người xem dễ hiểu và theo dõi quá trình thực hiện. Các biểu đồ, hình ảnh minh họa được sử dụng phù hợp, tạo ra báo cáo hấp dẫn.

**Tổng điểm tự đánh giá: 9.5/10**

Nhóm tự tin rằng những nỗ lực và kết quả đạt được xứng đáng với thang điểm 9.5/10. Đề tài này không chỉ giúp nhóm nắm vững kiến thức về hồi quy mà còn cải thiện kỹ năng làm việc nhóm và khả năng giải quyết các bài toán thực tế. Nhóm tin tưởng rằng những kinh nghiệm này sẽ là nền tảng vững chắc cho các dự án trong tương lai.

# Tài liệu tham khảo

Papiu, A. (2023). *Regularized Linear Models*. Retrieved from https://www.kaggle.com/code/apapiu/regularized-linear-models

Gusthema;Kin. (2023). *House Prices Prediction using TFDF*. Retrieved from https://www.kaggle.com/code/gusthema/house-prices-prediction-using-tfdf