

**Bộ thông tin và truyền thông**  
**Học viện Công nghệ Bưu Chính Viễn Thông**  
**Cơ sở tại thành phố Hồ Chí Minh**

**Khoa Công nghệ thông tin 2**



**Môn học** : Nhập môn khoa học dữ liệu  
**Đề tài** : Phân loại đa nhãn review phim  
**Giảng viên** : Thầy Nguyễn Ngọc Duy

Thành viên - nhóm 04 - D21CQCNHT01-N

N21DCCN038 : Hà Gia Huy

N21DCCN057 : Lê Trung Nguyên

N21DCCN059 : Trần Bình Phương Nhã

## NHẬN XÉT CỦA GIÁO VIÊN

\*\*\*

[illegible]

# I. Giới thiệu đề tài

## I.1. Tổng quan về đề tài

- Nhiệm vụ phân loại đa nhãn review, đánh giá của người xem về phim nhằm mục đích xác định những trải nghiệm, suy nghĩ khác nhau của người xem về các bộ phim mà họ đã thưởng thức. Trong thời đại kỹ thuật số hiện nay, việc xuất hiện những website, blog, hội nhóm,... cho phép người dùng đưa ra đánh giá về phim điện ảnh đã và đang rất phổ biến, xu hướng chia sẻ trải nghiệm tại các trang web chuyên về phim cũng tăng mạnh. Dẫn đến lượng dữ liệu về đánh giá của người dùng trên các trang này không ngừng tăng giúp cho việc thực hiện các nghiên cứu.

- Các tiến bộ trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural language processing – NLP) và công nghệ thông tin đã tạo điều kiện cho sự ra đời của các ứng dụng mới, đồng thời mở rộng không gian nghiên cứu dựa trên dữ liệu văn bản ngày càng phong phú. Phân loại văn bản là một trong những bài toán cơ bản của NLP. Phân loại văn bản là việc gán các nhãn đã có sẵn cho các đoạn văn bản tương ứng. Tùy thuộc vào từng trường hợp, mỗi văn bản có thể được gán một hoặc nhiều nhãn. Sự đa dạng các cách tiếp cận dẫn đến các kiểu phân loại văn bản khác nhau.

- Có nhiều cách phân loại văn bản trong NLP, nếu dựa trên số lượng nhãn được gán cho mỗi văn bản, có thể chia phân loại văn bản thành 2 loại là phân loại nhị phân (Binary) hay phân loại đa lớp (Multi-class) và phân loại đa nhãn (Multi-label). Khác với phân loại nhị phân hay phân loại đa lớp, mỗi văn bản chỉ được gán một nhãn, thì với phân loại đa nhãn, mỗi văn bản có thể được gán nhiều nhãn khác nhau.

Kiểu phân loại	Bài toán ví dụ	Nhãn
Phân loại nhị phân	Phân loại email spam	Spam, không spam
Phân loại đa lớp	Phân tích cảm xúc	Tích cực, tiêu cực, trung lập
Phân loại đa nhãn	Phân loại cảm xúc	Buồn, vui, giận, sợ hãi

*Bảng 1. Ví dụ về các bài toán phân loại văn bản.*

- Đối với ngữ cảnh cần thu thập ý kiến người dùng về một sản phẩm hay một vấn đề, bài toán phân loại nhị phân hay phân loại đa lớp sẽ tỏ ra kém hiệu quả vì trong những trường hợp này ngôn ngữ mà chúng ta sử dụng phức tạp hơn rất nhiều, gán nhãn đơn lẻ cho

vấn bản sẽ làm hạn chế khả năng trích xuất thông tin. Vì vậy sử dụng bài toán phân loại đa nhãn trở nên phù hợp.

## I.II. Mô tả bài toán

- Đề tài phân loại đa nhãn review, đánh giá của người xem về các bộ phim điện ảnh (phim chiếu rạp và cả phim lẻ, phim bộ) hướng đến việc, từ bình luận của người xem, có thể tìm ra được những thông tin cụ thể mà người dùng đề cập trong bình luận, từ đó có thể phân loại đánh giá của người dùng. Thay vì chỉ phân loại đánh giá tích cực hay tiêu cực, chúng tôi muốn đi sâu vào những thông tin mà người dùng đã đề cập để hiểu chi tiết hơn quan điểm của người dùng. Toàn bộ dữ liệu được chúng tôi thu thập từ trang review phim của Momo, nhiệm vụ của bài toán là phân loại đánh giá của khách hàng thành 5 nhãn (cụ thể sẽ được trình bình bên dưới), với mỗi đánh giá có thể thuộc một hay nhiều nhãn. Đề tài này có thể làm nền tảng cho việc xây dựng bộ lọc đánh giá của người dùng. Trong ngữ cảnh Machine Learning, đây là dạng bài toán Multi-Label Classification, với 5 nhãn khác nhau và 2 classes là 0 và 1.

## I.III. Các thuật toán máy học mà đề tài sử dụng

- Naïve Bayes (NB):

- NB là một thuật toán máy học được dựa trên định lý Bayes về giả định tính độc lập giữa các đặc trưng. Thuật toán này dựa trên xác suất để dự đoán hay phân lớp nhãn của một mẫu dựa trên xác suất tiên nghiệm và xác suất hậu nghiệm. NB có cơ chế hoạt động dựa trên định lý Bayes, một định lý cơ bản trong xác suất thống kê. Công thức Bayes cho phép tính xác suất hậu nghiệm (xác suất của một biến cố xảy ra sau khi có thông tin mới) dựa trên xác suất tiên nghiệm (xác suất của một biến cố diễn ra trước khi có thông tin mới) và thông tin mới đó. Xác suất hậu nghiệm được tính như sau:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

- Trong đó,  $P(y|X)$  là xác suất của lớp  $y$  khi biết  $X$ ,  $P(X|y)$  là xác suất của lớp  $X$  khi biết  $y$ ,  $P(y)$  là xác suất tiên nghiệm của lớp  $y$  và  $P(X)$  là xác suất đặc trưng  $X$ . Thông qua việc tính toán xác suất diễn ra của từng lớp, ta có thể tiến hành phân loại dựa vào phân lớp có xác suất cao nhất.

- Random Forests (RF):

- RF là thuật toán học có giám sát có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Người ta nói rằng càng có nhiều cây thì rừng càng mạnh. Random Forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Random Forests giúp hạn chế việc overfitting so với Decision Tree.

- K-Nearest Neighbors (KNN):

- KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu. Hình dưới đây là một ví dụ về KNN trong classification với  $K = 1$ . Khi training, thuật toán này không học một điều gì từ dữ liệu training, mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. KNN có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression.

- Support Vector Machine (SVM):

- SVM là một mô hình phân loại hoạt động bằng việc xây dựng một siêu phẳng (hyperplane) có  $(n - 1)$  chiều trong không gian  $n$  chiều của dữ liệu sao cho siêu phẳng này phân loại các lớp một cách tối ưu nhất. Nói cách khác, cho một tập dữ liệu có nhãn (học có giám sát), thuật toán sẽ dựa trên dữ liệu học để xây dựng một siêu phẳng tối ưu được sử dụng để phân loại dữ liệu mới. Ở không gian 2 chiều thì siêu phẳng này là 1 đường thẳng phân cách chia mặt phẳng không gian thành 2 phần tương ứng 2 lớp với mỗi lớp nằm ở 1 phía của đường thẳng.

- Logistic Regression (LR):

- LR là một phương pháp phân tích quan hệ giữa biến phụ thuộc  $Y$  với một hay nhiều biến độc lập  $X$ . Mô hình hóa sử dụng hàm tuyến tính (bậc 1). Các tham số của mô hình (hay hàm số) được ước lượng từ dữ liệu.

## II. Xây dựng bộ dữ liệu

### II.1. Thu thập dữ liệu và chọn nhãn

- Chúng tôi thu thập dữ liệu từ API của trang đánh giá phim của Momo và xây dựng bộ dữ liệu từ 5530 đánh giá của người dùng đã xem các bộ phim chiếu rạp.

- Để phân tích đánh giá của người xem về chất lượng của 1 bộ phim chiếu rạp và thông tin liên quan, chúng tôi chia thành 5 nhãn để phân loại. Bao gồm:

- Acting:

- Liên quan đến yếu tố diễn xuất của các diễn viên.

- Ví dụ: “nhân vật A diễn rất hay và nhập tâm”, “nữ chính thể hiện còn bị đơ, chưa bộc lộ được nét tính cách ngây thơ của nhân vật”,...

- Plot:

- Liên quan đến yếu tố cốt truyện, kịch bản, nội dung phim.

- Ví dụ: “không ngờ một bộ phim Việt Nam lại có một kịch bản chất lượng đến vậy”, “nội dung phim cũng không quá xuất sắc”, “kịch bản có nhiều plot twist, làm người xem bất ngờ”,...

- Scene:

- Liên quan đến yếu tố bối cảnh, góc quay, kỹ xảo được sử dụng trong phim.

- Ví dụ: “hiếm thấy một bộ phim dám sử dụng góc nhìn thứ nhất như phim này”, “công nhận CGI đỉnh thật sự, quá choáng ngợp”, “phim có sử dụng một số địa danh của Việt Nam kìa”,...

- Comedy:

- Liên quan đến yếu tố cảm xúc “hài hước” mà bộ phim có thể mang đến cho người xem.

- Ví dụ: “phim hài thiệt sự, coi mà cười đau cả bụng”, “phim của Trấn Thành bao giờ cũng vừa hài hước vừa sâu sắc”,...

- Touching:

- Liên quan đến yếu tố cảm xúc “cảm động” mà bộ phim có thể mang đến cho người xem.
- Ví dụ: “xem xong khóc muốn trôi rập”, “phim thật sự chạm đến cảm xúc của khán giả”, “cảnh người cha cứu con gái mình khiến mình rất xúc động”,...

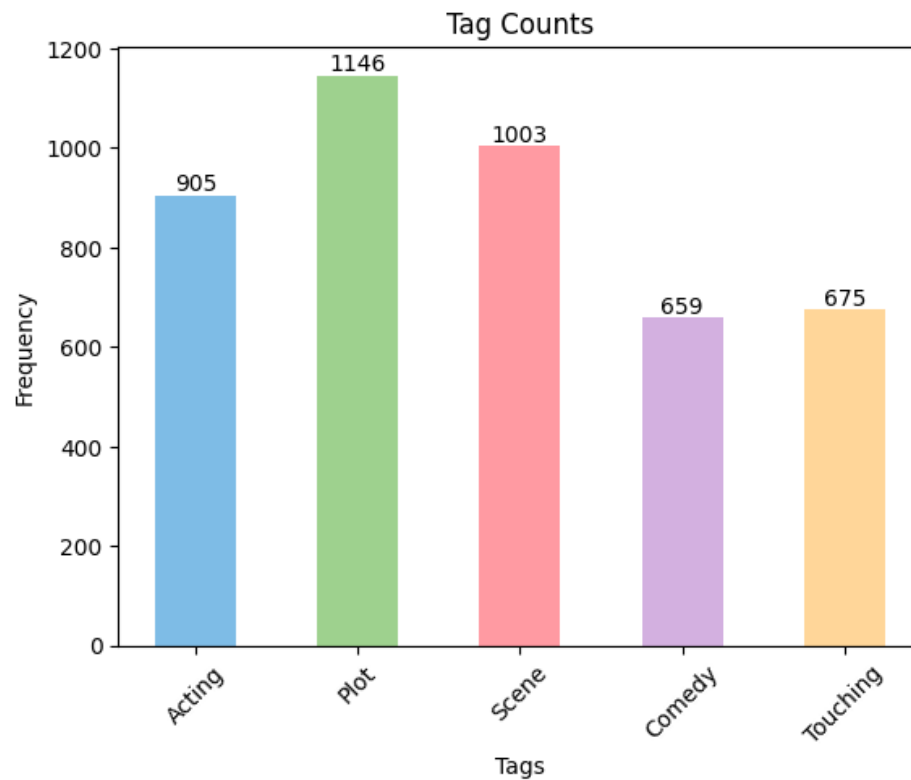
## **II.II. Quy trình đánh nhãn và tạo dataset**

- Sau khi thu thập và tiến hành sàng lọc sơ bộ, loại bỏ bớt các dòng dữ liệu không liên quan hoặc bị thiếu nội dung trong quá trình thu thập, chúng tôi thu được hơn 5700 dòng dữ liệu thô.
- 2 thành viên trong nhóm sẽ tiến hành gán nhãn cho 2 bản copy của dữ liệu. Để đảm bảo tính khách quan, 2 thành viên này sẽ không trao đổi với nhau trong quá trình đánh nhãn, cũng như không được biết được kết quả của đối phương.
- Sau khi 2 bản copy đã được đánh nhãn xong, thành viên thứ 3 sẽ tổng hợp lại thành 1 bộ dữ liệu chung:
  - Các nhãn được đánh kết quả giống nhau giữa 2 bản copy sẽ được giữ lại.
  - Các nhãn được đánh kết quả khác nhau sẽ được cả 3 thành viên thảo luận và thống nhất lại.
- Sau khi thống nhất tất cả nhãn, bộ dữ liệu sẽ được sàng lọc lại, tiếp tục loại bỏ các bình luận có nội dung spam, không phù hợp,...
- Thành viên thứ 3 tiến hành xử lý mất cân bằng nhãn.
- Sau bước xử lý mất cân bằng, chúng tôi thu được file Excel gồm 5530 dòng, file sau đó được chuyển đổi sang định dạng CSV.

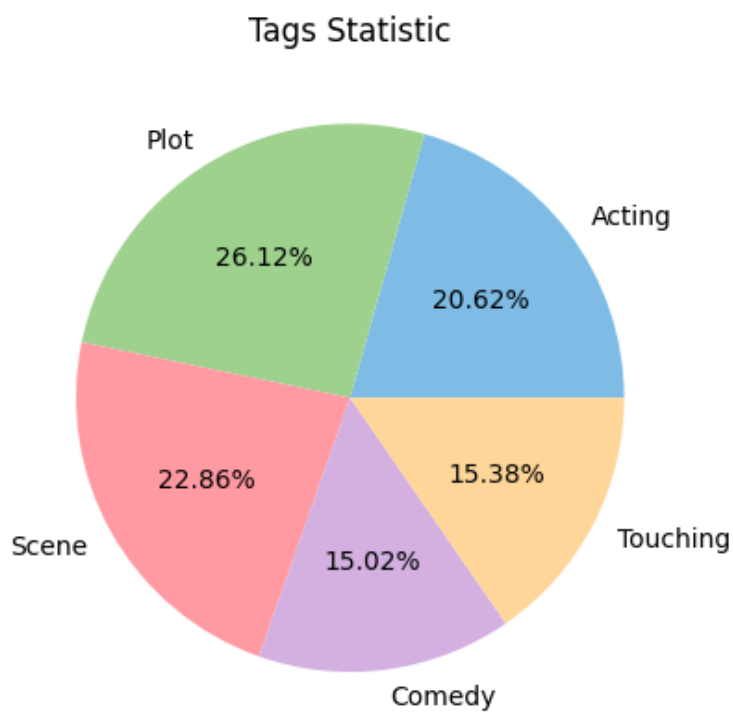
## **II.III. Trực quan hóa dữ liệu**

- Để có cái nhìn tổng quan và rõ ràng hơn về số lượng các nhãn, chúng tôi tiến hành trực quan hóa dữ liệu và thu được kết quả dưới đây:

### II.III.I. Thống kê

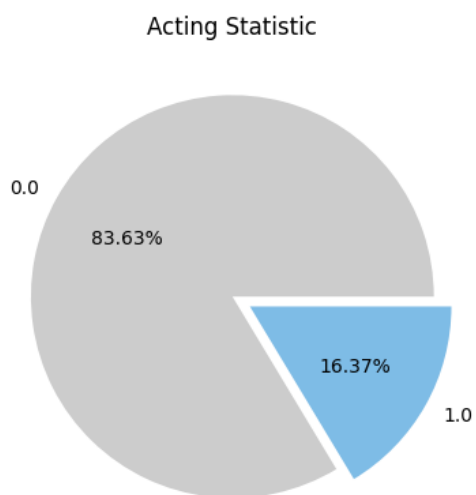


Hình 2.3.1. Kết quả thống kê tổng các nhãn của bài toán

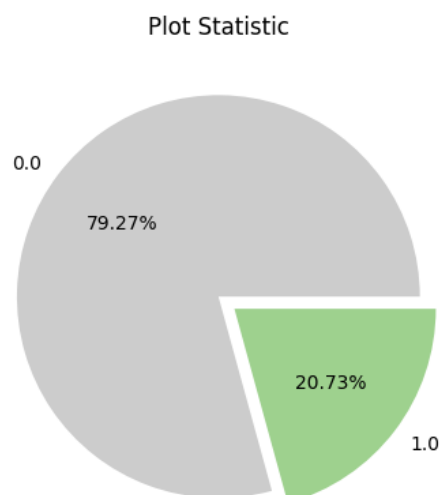


Hình 2.3.2. Kết quả thống kê phần trăm xuất hiện của từng nhãn

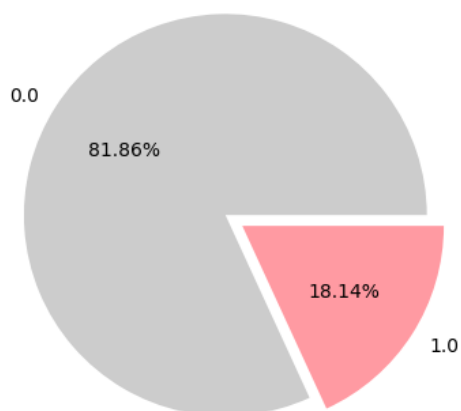




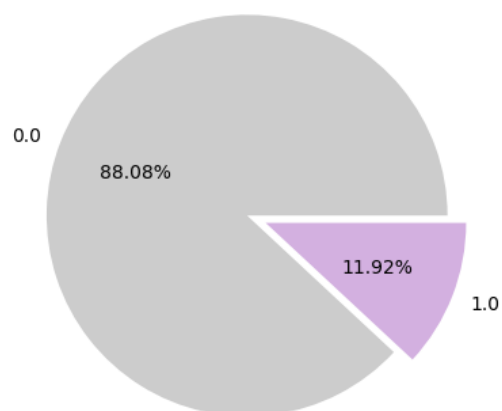
Hình 2.3.3. Thống kê nhãn Acting  
Scene Statistic



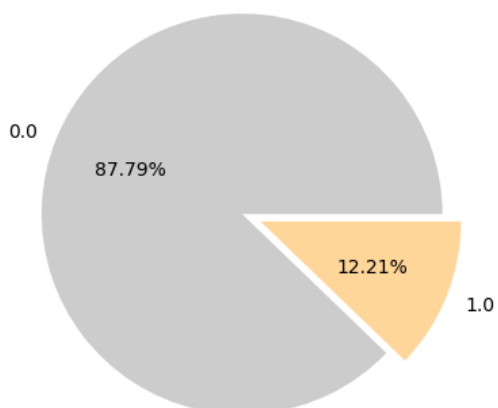
Hình 2.3.4. Thống kê nhãn Plot  
Comedy Statistic



Hình 2.3.5. Thống kê nhãn Scene  
Touching Statistic



Hình 2.3.6. Thống kê nhãn Comedy



Hình 2.3.7. Thống kê nhãn Touching

### **II.III.II. Nhận xét về sự phân bố các nhãn trong dataset**

- Nhãn cốt truyện (plot) có tần suất xuất hiện cao hơn, bởi cốt truyện, kịch bản là yếu tố then chốt quyết định chất lượng của hầu hết bộ phim.
- Các nhãn hài hước (comedy) và cảm động (touching) có tần suất xuất hiện thấp nhất, vì nó có liên quan mật thiết đến chủ đề, thông điệp và khả năng truyền tải cảm xúc của bộ phim nên không phải phim nào cũng được đề cập. Ngoài ra nó còn phụ thuộc vào cảm nhận cá nhân của người xem.
- Dựa vào dataset, các review có đề cập yếu tố diễn xuất (acting) hoặc yếu tố bối cảnh (scene) thường cũng nhắc đến yếu tố cốt truyện (plot). Điều đó cũng lý giải cho tần suất xuất hiện cao của nhãn cốt truyện (plot).

### **II.III.III. Phương pháp xử lý mất cân bằng nhãn**

- Giảm mất cân bằng nhãn là một yêu cầu quan trọng trong xử lý dữ liệu dạng multi label, bởi nó có thể ảnh hưởng đến độ chính xác của các thuật toán sử dụng xác suất và tần suất như Random Forests hay Naïve Bayes.
- Phương pháp được nhóm chúng tôi sử dụng là tiến hành lọc bớt các comment thiên về các nhãn đang có tần suất cao, đồng thời thu thập thêm các review về các bộ phim có chủ đề cảm động, hài hước,... để tăng tần suất xuất hiện cho các nhãn này.

## III. Xử lý dữ liệu và trích xuất đặc trưng

### III.I. Tiền xử lý dữ liệu

- Chúng tôi đã thực hiện các bước như sau để biến đổi dữ liệu thô:

- Xóa các ký tự đặc biệt, biểu tượng cảm xúc (emoji icon) và xóa các ký tự không cần thiết trong dataset (ví dụ: dấu chấm, phẩy, chấm than, ngoặc,...).
- Xóa khoảng trắng thừa.
- Đưa toàn bộ dữ liệu về chữ viết thường (lower case).
- Xóa các ký tự cố ý viết dài (ví dụ: “okeeeeeee” → “ok”, “hayyyyyyy” → “hay”).
- Chuẩn hóa Unicode (chuyển chuỗi từ dạng mã hóa Window-1252 sang Unicode utf-8).
- Chuẩn hóa dấu câu cho đúng vị trí (ví dụ: “tiêng Việt” → “tiếng Việt”, “mỗi ngày” → “mỗi ngày”).
- Thay các từ teen code thành dạng tiêu chuẩn (ví dụ: “ko đc” → “không được”, “vn” → “Việt Nam”).
- Tách từ (word tokenization) chia câu review thành các từ (ví dụ: “tác phẩm chủ đề công nghệ này hay xuất sắc” → “tác\_phẩm chủ\_đề công\_nghệ này hay xuất\_sắc”).
- Bỏ các từ stop word không mang nhiều ý nghĩa trong quá trình training (ví dụ: “dạ”, “lắm”, “nhe”,...). Các stop word này không được lựa chọn ngẫu nhiên mà được dựa trên các từ có số lần xuất hiện nhiều nhất trong bộ dataset.

### III.II. Trích xuất đặc trưng

- Chúng tôi sử dụng phương pháp TF-IDF có smooth.

- TF-IDF (Term frequency - inverse document frequency) là một trong những kỹ thuật cơ bản trong xử lý ngôn ngữ giúp đánh giá mức độ quan trọng của một từ trong văn bản.

- TF: Tính toán số lần xuất hiện của một từ trong văn bản.

- IDF: Ước lượng độ quan trọng của từ đó trong văn bản. Ví dụ như các từ “và, có, là, của, à,..” xuất hiện nhiều trong văn bản nhưng không đem lại nhiều ý nghĩa, lúc này trọng số của các từ này sẽ thấp.

$$\log \frac{1 + n_d}{1 + \text{df}(d, t)} + 1$$

- TF-IDF được tính như sau:

$$TF - IDF(t, d, D) = TF(t, d) * IDF(t, D)$$

- Như vậy, giá trị của TF-IDF sẽ tỷ lệ thuận đối với độ quan trọng của từ đó trong toàn bộ tập văn bản. Đối với các từ có điểm cao, đồng nghĩa với mức độ đặc biệt và hiếm gặp trên bộ văn bản. Trái lại, nếu có giá trị thấp thì các từ đó có thể là các từ xuất hiện đại trà trong toàn bộ tập dữ liệu.

## IV. Huấn luyện và đánh giá mô hình

### IV.1. Phương pháp huấn luyện

- Về cơ bản, có 3 phương pháp để giải quyết bài toán phân loại đa nhãn là:

- Problem Transformation
- Adapted Algorithm
- Ensemble approaches

- Ở phạm vi báo cáo lần này, chúng tôi trình bày là sử dụng phương pháp Problem Transformation.

- Ở phương pháp này, chúng tôi sẽ thực hiện bằng 3 cách khác nhau:


- Binary Relevance
- Classifier Chains
- Label Powerset

#### IV.1.1. Phương pháp Binary Relevance

- Ở cách này, mỗi nhãn sẽ được xem là một bài toán phân loại riêng biệt. Ví dụ trong trường hợp dưới đây:

- Với  $X$  là feature và  $Y$  là các labels. bài toán trên sẽ được chia làm 4 bài toán nhỏ riêng biệt (4 labels):

$X$	$Y_1$	$Y_2$	$Y_3$	$Y_4$
$X^{(1)}$	0	0	0	1
$X^{(2)}$	1	0	0	0
$X^{(3)}$	1	0	0	1
$X^{(4)}$	0	1	0	0
$X^{(5)}$	0	1	1	0



$X$	$Y_1$
$X^{(1)}$	0
$X^{(2)}$	1
$X^{(3)}$	1
$X^{(4)}$	0
$X^{(5)}$	0

$X$	$Y_2$
$X^{(1)}$	0
$X^{(2)}$	0
$X^{(3)}$	0
$X^{(4)}$	1
$X^{(5)}$	1

$X$	$Y_3$
$X^{(1)}$	0
$X^{(2)}$	0
$X^{(3)}$	0
$X^{(4)}$	0
$X^{(5)}$	1

Hình 4.1.1. Minh họa quá trình xử lý của phương pháp Binary Relevance

- Sau đó, dùng các thuật toán phân loại để huấn luyện mô hình. Chúng tôi sử dụng thuật toán Naive Bayes, Random Forest, SVC...

- Ưu điểm:

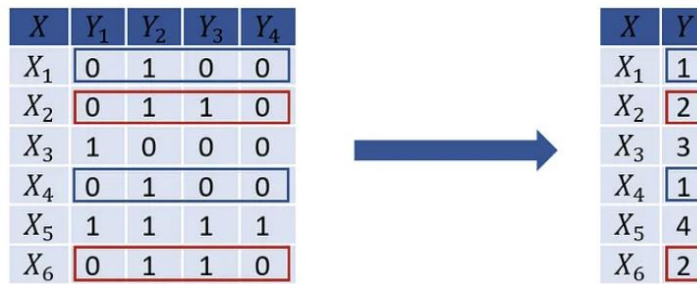
- Đơn giản và dễ triển khai: Binary Relevance chia bài toán phân loại đa nhãn thành nhiều bài toán phân loại nhị phân độc lập, mỗi nhãn (label) được xử lý như một bài toán riêng. Do đó, nó dễ dàng áp dụng với các thuật toán phân loại nhị phân thông thường.
- Linh hoạt: Phương pháp này có thể được sử dụng với bất kỳ mô hình phân loại nhị phân nào (như Logistic Regression, SVM, Random Forest, v.v.). Điều này mang lại tính linh hoạt cao trong việc lựa chọn mô hình phân loại.
- Khả năng mở rộng: Binary Relevance mở rộng tốt với nhiều nhãn vì nó chỉ cần giải quyết nhiều bài toán nhị phân thay vì một bài toán phức tạp hơn.
- Tính song song: Do các mô hình phân loại cho từng nhãn là độc lập với nhau, các mô hình này có thể được huấn luyện song song, giúp giảm thời gian huấn luyện nếu có tài nguyên tính toán mạnh.

- Nhược điểm:

- Không xem xét mối quan hệ giữa các nhãn: Phương pháp này bỏ qua mối quan hệ phụ thuộc giữa các nhãn. Trong nhiều trường hợp, các nhãn có thể liên quan hoặc phụ thuộc lẫn nhau. Vì vậy, kết quả có thể kém hiệu quả với dữ liệu có sự phụ thuộc giữa các nhãn.
- Tăng kích thước bài toán: Với số lượng nhãn lớn, số lượng mô hình nhị phân cần huấn luyện cũng sẽ tăng lên tương ứng, khiến cho việc xử lý trở nên tốn kém về thời gian và tài nguyên tính toán.
- Hiệu suất bị ảnh hưởng bởi phân phối nhãn không đều: Nếu tập dữ liệu có phân phối nhãn không đều, mô hình cho các nhãn hiếm có thể không hoạt động tốt do thiếu dữ liệu huấn luyện cho các nhãn này.

#### **IV.1.II. Phương pháp Label Powerset**

- Cách làm này sẽ xem xét các trường hợp có các nhãn là giống nhau mà gom chúng lại thành một lớp.



Hình 4.1.2. Minh họa quá trình xử lý của phương pháp Label Powerset

- Như ở ví dụ dưới đây, x1 và x4 có nhãn giống nhau nên sẽ được gom thành một nhãn mới, tương tự với x3 và x6.

- Ưu điểm:

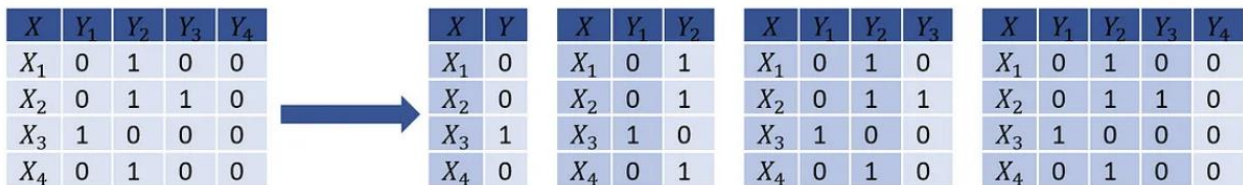
- Xử lý mối quan hệ giữa các nhãn: Label Powerset không bỏ qua sự phụ thuộc giữa các nhãn. Phương pháp này ghi nhận và xem xét tất cả các kết hợp của các nhãn, từ đó có thể cải thiện độ chính xác của mô hình trong những bài toán mà các nhãn có mối quan hệ phụ thuộc.
- Giữ lại thông tin về phân phối nhãn: Ghi nhận toàn bộ phân phối nhãn theo các tổ hợp khác nhau, điều này giúp mô hình học được các tổ hợp nhãn cụ thể mà dữ liệu biểu diễn.

- Nhược điểm:

- Số lượng nhãn mới có thể rất lớn: Số lượng nhãn mới có thể bùng nổ nếu số lượng nhãn gốc lớn. Điều này dẫn đến việc mô hình phải phân loại trên không gian nhãn rất lớn, có thể làm giảm hiệu quả huấn luyện và yêu cầu nhiều tài nguyên hơn.
- Xử lý dữ liệu thưa: Khi số lượng nhãn tăng lên, không phải tất cả các tổ hợp nhãn đều xuất hiện trong dữ liệu huấn luyện. Điều này có thể dẫn đến vấn đề dữ liệu thưa thớt, tức là nhiều tổ hợp nhãn có thể không xuất hiện đủ để mô hình học tốt, dẫn đến độ chính xác giảm trên các tổ hợp hiếm.
- Khó dự đoán các nhãn chưa từng thấy: Mô hình LP chỉ học được các tổ hợp nhãn có trong dữ liệu huấn luyện. Nếu gặp phải các tổ hợp nhãn mới chưa xuất hiện trong dữ liệu, nó không thể dự đoán chính xác tổ hợp đó.

### IV.I.III. Phương pháp Classifier Chains

- Ban đầu mô hình được train trên input đầu vào và một nhãn. Sau khi huấn luyện xong, nhãn đã được huấn luyện sẽ trở thành input và tiếp tục huấn luyện để dự đoán ra nhãn tiếp theo.



X	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>
X <sub>1</sub>	0	1	0	0
X <sub>2</sub>	0	1	1	0
X <sub>3</sub>	1	0	0	0
X <sub>4</sub>	0	1	0	0

X	Y
X <sub>1</sub>	0
X <sub>2</sub>	0
X <sub>3</sub>	1
X <sub>4</sub>	0

X	Y <sub>1</sub>	Y <sub>2</sub>
X <sub>1</sub>	0	1
X <sub>2</sub>	0	1
X <sub>3</sub>	1	0
X <sub>4</sub>	0	1

X	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
X <sub>1</sub>	0	1	0
X <sub>2</sub>	0	1	1
X <sub>3</sub>	1	0	0
X <sub>4</sub>	0	1	0

X	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>
X <sub>1</sub>	0	1	0	0
X <sub>2</sub>	0	1	1	0
X <sub>3</sub>	1	0	0	0
X <sub>4</sub>	0	1	0	0

Hình 4.1.3. Minh họa quá trình xử lý của phương pháp Classifier Chains

- Ưu điểm:

- Xem xét mối quan hệ giữa các nhãn: Classifier Chains xem xét sự phụ thuộc giữa các nhãn bằng cách sử dụng nhãn đã dự đoán ở các bước trước làm đặc trưng bổ sung cho các bộ phân loại sau trong chuỗi. Điều này giúp mô hình khai thác được mối quan hệ giữa các nhãn để dự đoán chính xác hơn.
- Linh hoạt: Có thể được áp dụng với hầu hết các thuật toán phân loại nhị phân, mang lại sự linh hoạt trong việc lựa chọn mô hình phân loại cơ bản. Ví dụ: Logistic Regression, Random Forest, SVM, v.v.
- Có thể điều chỉnh: Chuỗi của các bộ phân loại có thể được điều chỉnh bằng cách thử nghiệm nhiều chuỗi khác nhau hoặc sử dụng các biến thể như Random Classifier Chains, nơi nhiều chuỗi ngẫu nhiên được thử để cải thiện kết quả.

- Nhược điểm:

- Phụ thuộc vào thứ tự chuỗi: Hiệu suất của mô hình phụ thuộc rất nhiều vào thứ tự của các nhãn trong chuỗi. Một thứ tự chuỗi không tối ưu có thể dẫn đến dự đoán không chính xác.
- Hiệu ứng truyền lỗi: Nếu một nhãn được dự đoán sai ở đầu chuỗi, lỗi này có thể lan truyền và ảnh hưởng đến các nhãn được dự đoán sau đó. Đây là hiệu ứng "truyền lỗi" trong chuỗi, làm giảm độ chính xác của toàn bộ mô hình.
- Tốn kém tài nguyên: Vì mỗi nhãn cần một bộ phân loại riêng trong chuỗi, số lượng mô hình cần huấn luyện sẽ bằng số lượng nhãn. Điều này có thể làm tăng yêu cầu



về bộ nhớ và thời gian huấn luyện, đặc biệt là khi chuỗi được điều chỉnh nhiều lần hoặc dữ liệu lớn.

## IV.II. Các chỉ số đánh giá (Evaluation Metrics)

- Phân loại đa nhãn đòi hỏi các kỹ thuật đánh giá khác so với các kỹ thuật đánh giá phân loại đơn nhãn truyền thống. Trong bài báo cáo này, chúng tôi sử dụng các độ đo Hamming Loss (HL), Micro Averaged Precision (MicroP), Macro Averaged Precision (MacroP), Micro Averaged Recall (MicroR), Macro Averaged Recall (MacroR), Micro F1 Score (Micro F1) và Macro F1 Score (Macro F1).

### IV.II.I. Hamming Loss (HL)

$$HL = \frac{1}{|N| \cdot |L|} \sum_{l=1}^L \sum_{i=1}^N Y_{i,l} \oplus X_{i,l}.$$

- Hamming Loss là một thước đo chủ yếu được sử dụng trong các bài toán phân loại đa nhãn. Nó đo tỷ lệ nhãn được dự đoán không chính xác.

- Hamming Loss được dùng trong các tình huống trong đó mỗi instance có thể có nhiều nhãn và ta cần nắm bắt mức độ chính xác của các dự đoán của mô hình trên tất cả các nhãn trong instance đó. Không giống như Precision hay Recall, Hamming Loss sẽ ghi nhận cả FP và FN ở cấp độ nhãn, do đó, nó hữu ích cho các ứng dụng mà mọi nhãn được dự đoán ra đều quan trọng.

### IV.II.II. Micro-average Precision và Macro-average Precision

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

- Precision đo lường có bao nhiêu dự đoán positive do mô hình đưa ra thực sự đúng. Nó cho chúng ta biết mức độ chính xác của mô hình trong việc dự đoán lớp positive.

- Precision thường được dùng trong trường hợp hậu quả của FN là lớn hơn FP, chẳng hạn như trong các trường hợp như phát hiện thư rác qua email, khi chúng tôi muốn giảm thiểu số lượng email thông thường được phân loại là thư rác.

$$MicroP = \frac{\sum_{c_i \in C} TPs(c_i)}{\sum_{c_i \in C} TPs(c_i) + FPs(c_i)}.$$

- Micro-average Precision tổng hợp sự đóng góp của tất cả các class để tính toán precision tổng thể. Micro-average Precision tổng hợp sự đóng góp của tất cả các class để tính toán các số liệu. Về cơ bản, nó xem xét từng instance (bất kể class) theo cách như nhau.

- Cách tiếp cận này hữu ích khi chúng ta quan tâm nhiều hơn đến hiệu suất tổng thể của tất cả các lớp hơn là cách mô hình hoạt động trên từng lớp riêng lẻ. Micro-average Precision được ưu tiên hơn khi các lớp mất cân bằng vì nó mang lại nhiều trọng số hơn cho các class có tần suất xuất hiện cao hơn.

$$MacroP = \frac{\sum_{c_i \in C} P(D, c_i)}{|C|}.$$

- Macro-average Precision là trung bình cộng của các precision theo từng class. Macro-average Precision tính toán precision một cách độc lập cho từng class và sau đó lấy giá trị trung bình của các giá trị đó. Cách tiếp cận này đối xử bình đẳng với tất cả các lớp, bất kể tần suất của chúng.

- Macro-average Precision thường được dùng khi chúng ta muốn đánh giá mô hình hoạt động như thế nào trên từng class một cách độc lập, điều này rất hữu ích trong việc xác định các vấn đề với các lớp có ít thể hiện. Tuy nhiên, nó có thể bị sai lệch nếu có sự mất cân bằng đáng kể trong việc phân bố các class.

#### IV.II.III. Micro-average Recall và Macro-average Recall

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

- Recall đo lường số lượng mẫu positive thực tế mà mô hình đã xác định chính xác. Nó nắm bắt khả năng của mô hình trong việc phát hiện tất cả các trường hợp positive.

- Recall thường hữu dụng trong các trường hợp việc bỏ lỡ các trường hợp positive gây hậu quả to lớn hơn so với dự đoán FP, chẳng hạn như trong chẩn đoán y tế, trong đó điều quan trọng là phải bắt được càng nhiều trường hợp càng tốt.

$$Micro\ R = \frac{\sum_{c_i \in C} TP_s(c_i)}{\sum_{c_i \in C} TP_s(c_i) + FN_s(c_i)}.$$

- Micro-average Recall tổng hợp sự đóng góp của tất cả các class để tính toán recall tổng thể. Micro-average Recall tổng hợp sự đóng góp của tất cả các class để tính toán các số liệu. Về cơ bản, nó xem xét từng instance (bất kể class) theo cách như nhau.

- Cách tiếp cận này hữu ích khi chúng ta quan tâm nhiều hơn đến hiệu suất tổng thể của tất cả các lớp hơn là cách mô hình hoạt động trên từng lớp riêng lẻ. Micro-average Recall được ưu tiên hơn khi các lớp mất cân bằng vì nó mang lại nhiều trọng số hơn cho các class có tần suất xuất hiện cao hơn.

$$MacroR = \frac{\sum_{c_i \in C} R(D, c_i)}{|C|}.$$

- Macro-average Recall là trung bình cộng của các recall theo từng class. Macro-average Recall tính toán recall một cách độc lập cho từng class và sau đó lấy giá trị trung bình của các giá trị đó. Cách tiếp cận này đối xử bình đẳng với tất cả các lớp, bất kể tần suất của chúng.

- Macro-average Recall thường được dùng khi chúng ta muốn đánh giá mô hình hoạt động như thế nào trên từng class một cách độc lập, điều này rất hữu ích trong việc xác định các vấn đề với các lớp có ít thể hiện. Tuy nhiên, nó có thể bị sai lệch nếu có sự mất cân bằng đáng kể trong việc phân bố các class.

#### IV.II.IV. Micro-average F1 Score và Macro-average F1 Score

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN}$$

- F1 Score là giá trị trung bình điều hòa giữa Precision và Recall, mang lại sự cân bằng giữa hai chỉ số trên. Thước đo này hữu ích khi bạn cần một số liệu duy nhất xem xét cả Precision và Recall, đặc biệt là với các tập dữ liệu không cân bằng và cả kết quả dương tính giả (FP) và âm tính giả (FN) cần được xem xét như nhau.

$$MicroF1 = 2 * \frac{MicroP * MicroR}{MicroP + MicroR}.$$

- Micro-average F1 Score là giá trị trung bình điều hòa của micro precision và micro recall. Nó tính toán ra 1 điểm F1 score duy nhất dựa trên các kết quả TP, TN, FP, FN tổng hợp từ toàn bộ các nhãn trên phạm vi toàn cục. Micro-average F1 Score tổng hợp sự đóng góp của tất cả các class để tính toán các số liệu. Về cơ bản, nó xem xét từng instance (bất kể class) theo cách như nhau.

- Cách tiếp cận này hữu ích khi chúng ta quan tâm nhiều hơn đến hiệu suất tổng thể của tất cả các lớp hơn là cách mô hình hoạt động trên từng lớp riêng lẻ. Micro-average F1 Score được ưu tiên hơn khi các lớp mất cân bằng vì nó mang lại nhiều trọng số hơn cho các class có tần suất xuất hiện cao hơn.

$$MacroF1 = \frac{1}{N} \sum_{i=0}^N F1.$$

- Macro-average F1 score là giá trị trung bình điều hòa của macro precision và macro recall, được tính bằng cách tính F1 score cho từng class riêng lẻ sau đó tính giá trị trung bình giữa các F1 score này. Macro-average F1 Score tính toán f1 score một cách độc lập cho từng class và sau đó lấy giá trị trung bình của các giá trị đó. Cách tiếp cận này đối xử bình đẳng với tất cả các lớp, bất kể tần suất của chúng.

- Macro-average F1 Score thường được dùng khi chúng ta muốn đánh giá mô hình hoạt động như thế nào trên từng class một cách độc lập, điều này rất hữu ích trong việc xác định các vấn đề với các lớp có ít thể hiện. Tuy nhiên, nó có thể bị sai lệch nếu có sự mất cân bằng đáng kể trong việc phân bố các class.

### IV.III. Các bước huấn luyện mô hình

- Với 1 trong 3 kỹ thuật kể trên (Binary Relevance, Label Powerset và Classifier Chains), ta tiến hành cross validate với GridSearchCV và các thuật toán học máy và các thông số mô hình khác nhau.

- Đánh giá các mô hình thu được dựa theo các tiêu chí thước đo như Micro-average Precision, Macro-average Precision, Micro-average Recall, Macro-average Recall, Micro-average F1-Score, Macro-average F1-Score.

- Tìm ra phương pháp, thuật toán học máy tốt nhất, và bộ thông số tối ưu nhất trong tất cả mô hình đã huấn luyện.

- Tiến hành tái tạo lại best model với các thông số tối ưu ấy, và training cho mô hình này bằng tập dataset X\_train đầy đủ.

## IV.IV. So sánh kết quả của các phương pháp huấn luyện

### IV.IV.I. Kết quả

Method	Classifier	Micro F1	Macro F1	Micro P	Macro P	Micro R	Macro R
Binary Relevance	NB	0.47084	0.45532	0.86291	0.88371	0.32420	0.31383
	<b>RF</b>	<b>0.74712</b>	<b>0.75481</b>	<b>0.86619</b>	<b>0.86918</b>	<b>0.65707</b>	<b>0.66927</b>
	KNN	0.27737	0.29294	0.60750	0.61529	0.17972	0.19586
	SVC	0.73163	0.73565	0.87898	0.88799	0.62671	0.62903
	LR	0.59853	0.58945	0.90414	0.91333	0.44759	0.43862
Label Powerset	NB	0.36228	0.35249	0.85716	0.87853	0.22809	0.22355
	<b>RF</b>	<b>0.61889</b>	<b>0.62140</b>	<b>0.87332</b>	<b>0.87742</b>	<b>0.47933</b>	<b>0.48277</b>
	KNN	0.27820	0.29374	0.60629	0.61558	0.18054	0.19645
	SVC	0.59445	0.59478	0.86718	0.88195	0.45250	0.45111
	LR	0.49325	0.49061	0.89388	0.90780	0.34094	0.33835
Classifier Chains	NB	0.47688	0.46384	0.86783	0.88586	0.32928	0.32050
	<b>RF</b>	<b>0.74509</b>	<b>0.75257</b>	<b>0.86956</b>	<b>0.87253</b>	<b>0.65195</b>	<b>0.66384</b>
	KNN	0.27253	0.29290	0.60642	0.61465	0.17972	0.19586
	SVC	0.73081	0.73481	0.88006	0.88873	0.62501	0.62725
	LR	0.57750	0.56709	0.90007	0.90834	0.42537	0.44645

### IV.IV.II. Kết luận

- Phương pháp cho kết quả tốt nhất (dựa theo Micro F1 score) là: Binary Relevance
- Thuật toán học máy tương ứng cho kết quả tốt nhất là: Random Forest
- Giá trị Accuracy Score thu được từ mô hình này là: 0.7242314647377939
- Giá trị Hamming Loss thu được từ mô hình này là: 0.06672694394213381

### IV.IV.III. Nhận xét

- Hai độ đo mà chúng tôi quan tâm nhất là F1-score và Hamming Loss. Đầu tiên có thể thấy, Binary Relevance và Classifier Chains là hai phương pháp có điểm cao nhất và gần ngang bằng nhau, Binary Relevance có điểm cao hơn một chút. Kết quả này cho thấy bộ dữ liệu của chúng tôi vừa có sự tương quan và vừa không tương quan giữa nhãn. Có thể hiểu rằng một nhãn có sự tương quan với nhãn này nhưng lại không có sự tương quan với nhãn khác. Ví dụ, trong quá trình gắn nhãn dữ liệu, chúng tôi nhận thấy rằng, khi người dùng đánh giá một vấn đề liên quan đến nhãn diễn xuất (Acting) thì khả năng cao sẽ nhắc đến vấn đề liên quan đến nhãn bối cảnh (Scene), nhưng khi nhắc đến nhãn hài

hước (Comedy) lại có khả năng rất thấp nhắc đến nhãn cảm động (Touching). Thứ hai, trong cả 3 phương pháp thì mô hình Random Forest cho kết quả tốt nhất.