Mục lục

[Mục lục 1](#_Toc120461209)

[Chương 1. Giới thiệu 3](#_Toc120461210)

[1.1. Động lực và đặt vấn đề 3](#_Toc120461211)

[1.2. Phân lớp đa nhãn 3](#_Toc120461212)

[1.3. Phương pháp xử lý mất cân bằng dữ liệu 4](#_Toc120461213)

[1.3.1. Resampling 4](#_Toc120461214)

[1.3.2. Class weighting 5](#_Toc120461215)

[1.3.3. Ensemble classifier 5](#_Toc120461216)

[1.4. Khó khăn và thách thức 5](#_Toc120461217)

[1.5. Các định nghĩa cơ bản 6](#_Toc120461218)

[Chương 2. Xây dựng dữ liệu 8](#_Toc120461219)

[2.1. Công nghệ sử dụng 8](#_Toc120461220)

[2.2. Quá trình xây dựng dữ liệu 8](#_Toc120461221)

[2.2.1. Thu thập dữ liệu 8](#_Toc120461222)

[2.2.2. Gán nhãn 9](#_Toc120461223)

[2.2.3. Kết quả thu được 11](#_Toc120461224)

[Chương 3. Các nghiên cứu liên quan 14](#_Toc120461225)

[3.1. Các mô hình học máy cơ bản 14](#_Toc120461226)

[3.1.1. Naïve Bayes 14](#_Toc120461227)

[3.1.2. Support Vector Machines 14](#_Toc120461228)

[3.1.3. Random Forest 15](#_Toc120461229)

[3.1.4. Logistic Regression 15](#_Toc120461230)

[3.2. Các nghiên cứu liên quan về các kỹ thuật xử lý mất cân bằng dữ liệu 17](#_Toc120461231)

[3.2.1. Bagging Ensemble 17](#_Toc120461232)

[3.2.2. Phương pháp boosting 17](#_Toc120461233)

[3.2.3. Resampling 18](#_Toc120461234)

[Chương 4. Mô hình đề xuất 19](#_Toc120461235)

[4.1. Mô hình phân lớp quan điểm 19](#_Toc120461236)

[4.1.1. Pha 1: Tiền xử lý dữ liệu 19](#_Toc120461237)

[4.1.2. Pha 2: Biểu diễn và trích chọn đặc trưng dữ liệu 20](#_Toc120461238)

[4.1.3. Pha 3: Mô hình phân lớp 21](#_Toc120461239)

[4.2. Kiến trúc THANOS 22](#_Toc120461240)

[4.2.1. Động lực 22](#_Toc120461241)

[4.2.2. Mô hình kiến trúc THANOS 24](#_Toc120461242)

[Chương 5. Thực nghiệm và kết quả 30](#_Toc120461243)

[5.1. Cài đặt và cấu hình 30](#_Toc120461244)

[5.1.1. Triển khai mô hình 30](#_Toc120461245)

[5.1.2. Môi trường thực nghiệm 30](#_Toc120461246)

[5.2. Phương pháp đánh giá 30](#_Toc120461247)

[5.3. Kết quả thực nghiệm 33](#_Toc120461248)

[Kết luận và định hướng phát triển 50](#_Toc120461249)

[Tài liệu tham khảo 51](#_Toc120461250)

# Giới thiệu

## Động lực và đặt vấn đề

Trong những năm gần đây, ngày càng có nhiều số lượng dữ liệu liên quan tới đánh giá các sản phẩm trên Internet, đặc biệt là trên các nền tảng thương mại điện tử. Không chỉ có vai trò quan trọng trong quá trình quyết định mua hàng của người tiêu dùng, những đánh giá này còn quan trọng đối với các doanh nghiệp trong việc theo dõi phản hồi của khách hàng cũng như hiểu nhu cầu của khách hàng. Mỗi đánh giá có thể đề cập đến nhiều hơn một khía cạnh, hơn nữa, các khía cạnh đó lại thường không được nhắc tới trong câu một cách trực tiếp (ví dụ, trong câu “Sữa ngon, hạn sử dụng xa”, hai khía cạnh “chất lượng” và “độ an toàn” được nhắc đến: “ngon”, “hạn sử dụng xa”, mà không có sự xuất hiện trực tiếp của các từ “chất lượng”, “độ an toàn” trong câu). Trong bài nghiên cứu này, chúng tôi tập trung giải quyết vấn đề đầu tiên với dữ liệu thương mại điện tử Việt Nam: phân lớp khía cạnh các đánh giá (xác định các khía cạnh được nhắc tới trong đánh giá và coi chúng là các lớp trong mô hình phân lớp, và phân lớp các dữ liệu về đánh giá vào các lớp đó). Những tập dữ liệu về đánh giá sản phẩm trước đây đã được phân lớp khía cạnh bằng mô hình phân lớp đa nhãn, nhưng thường bỏ qua vấn đề mất cân bằng dữ liệu. Phân phối lớp không cân bằng của tập dữ liệu đã gây ra hạn chế đáng kể về hiệu suất đối với hầu hết các thuật toán học phân lớp vốn giả định một phân phối lớp tương đối cân bằng. Vì vậy, chúng tôi còn đưa ra phân tích về vấn đề mất cân bằng đối với dữ liệu thương mại điện tử Việt Nam và các biện pháp khắc phục.

Nội dung của bài nghiên cứu gồm các phần sau: chương 2 mô tả quá trình xây dựng tập dữ liệu: quá trình thu thập và guideline gán nhãn; tiếp đó ở chương 3 là các mô hình học máy cơ bản, gồm những mô hình học máy được sử dụng để giải quyết bài toán; chương 4 nói về những nghiên cứu liên quan của chúng tôi về những phương pháp giải quyết mất cân bằng dữ liệu, chương 5 là mô hình chúng tôi đề xuất, chương 6 sẽ là kết quả thực nghiệm, và chương 7 là kết luận với những dự định trong tương lai của chúng tôi.

## Phân lớp đa nhãn

Phân lớp khía cạnh là một bài toán phân lớp đa nhãn, bởi một đánh giá có thể chứa một hoặc nhiều khía cạnh (ví dụ: “Sữa ngon, hạn sử dụng xa”, ở đây, có hai khía cạnh được đề cập tới là chất lượng và độ an toàn). Phân lớp đa nhãn nghĩa là nhiều nhãn có thể được gán cho một dữ liệu, hay một dữ liệu sẽ được phân vào ít nhất một lớp. Trong bài toán phân lớp đa nhãn, không có ràng buộc nào về số lượng nhãn có thể được gán cho một dữ liệu.

Chiến lược thông dụng nhất là One-vs-all (hay còn gọi là One-against-rest), là chuyển thành nhiều bài toán phân lớp nhị phân. Cụ thể, nếu có N lớp thì sẽ có N bộ phân lớp, mỗi bộ phân lớp tương ứng với một lớp. Chiến lược này hoạt động như sau: bộ phân lớp thứ nhất giúp xác định xem dữ liệu có thuộc vào lớp 1 là bao nhiêu, hoặc tính xác suất để dữ liệu rơi vào lớp 1 là bao nhiêu, và tương tự như thế đối với các bộ phân lớp còn lại. Tất cả những nhãn dương được đưa ra bởi các bộ phân lớp nhị phân được coi là nhãn cuối cùng của bộ phân lớp đa nhãn.

Một ý tưởng khác cho bài toán phân lớp đa nhãn là Round Robin; cụ thể là học một bộ phân loại cho mỗi cặp lớp, tương tự như các giải đấu vòng tròn trong thể thao, trong đó mỗi người tham gia là được ghép với những người tham gia khác.

Về cơ bản, cách tiếp cận One-vs-all chạy nhanh hơn, dễ cài đặt hơn, và thường cho kết quả cao hơn. Vì vậy, trong bài nghiên cứu này, chúng tôi sẽ chọn phương pháp này để phân lớp khía cạnh. Chúng tôi áp dụng chiến lược One-vs-all để giải bằng bài toán phân lớp khía cạnh như sau: sẽ có N bộ phân lớp, ứng với mỗi khía cạnh mà chúng tôi sẽ đề xuất ở chương 2; các bộ phân lớp sẽ xác định xem dữ liệu có nhãn đó không, và kết quả thu được sẽ là các dữ liệu đa nhãn.

## Phương pháp xử lý mất cân bằng dữ liệu

Một trong những thử thách chính trong rất nhiều chủ đề khi sử dụng học máy, chính là sự mất cân bằng dữ liệu. Các thuật toán học máy rất dễ trả về kết quả dự đoán không đạt yêu cầu khi được huấn luyện trên tập dữ liệu không cân bằng. Ta lấy một ví dụ điển hình về tập dữ liệu có hai lớp phân loại, trong đó lớp thiểu số chỉ chiếm 5% tổng dữ liệu. Khi huấn luyện, mô hình sẽ dự đoán gần như tất cả dữ liệu đều thuộc về lớp đa số, độ chính xác sẽ là 95%, một kết quả có vẻ tốt. Tuy nhiên, trên thực tế chúng ta đang tạo ra một bộ phân lớp không có ích. Trong bài nghiên cứu này, chúng ta sẽ nghiên cứu các phương pháp để xử lý mất cân bằng dữ liệu.

### Resampling

Resampling là phương pháp xử lý mất cân bằng dữ liệu phi kinh nghiệm, bằng cách xem xét tỷ lệ mẫu dữ liệu giữa các lớp. Phương pháp này được thực hiện trước khi dữ liệu được đưa vào mô hình huấn luyện. Phương pháp Resampling được cho là phương pháp tốn ít tài nguyên của máy và do đó mô hình sẽ chạy nhanh hơn, ngoài ra còn có hiệu quả tương đối tốt. Do đó chúng tôi sẽ sử dụng phương pháp này để áp dụng vào mô hình của chúng tôi.

### Class weighting

Phương pháp class weight là phương pháp đưa ra trọng số khác nhau cho lớp đa số và lớp thiểu số. Sự khác nhau về trọng số của mỗi lớp sẽ ảnh hưởng đến việc phân lớp trong quá trình huấn luyện mô hình. Mục đích của điều này là giảm sự phân lớp sai cho lớp thiểu số bằng cách tăng trọng số của lớp thiểu số và giảm trọng số của lớp đa số. Phương pháp này được cho là tốn tài nguyên và không quá hiệu quả với các lớp quá chênh lệch về số lượng dữ liệu.

### Ensemble classifier

Bộ phân lớp tổng hợp được mong đợi sẽ giúp tăng độ chính xác của các bộ phân lớp đơn lẻ bằng cách kết hợp chúng và đã được áp dụng thành công cho các tập dữ liệu không cân bằng. Phương pháp này đã được chứng minh là hiệu quả hơn các kỹ thuật lấy mẫu dữ liệu để tăng cường hiệu suất phân lớp của dữ liệu mất cân bằng. Tuy nhiên, kỹ thuật này vẫn gặp khó khăn trong việc phân lớp cho lớp thiểu số. Vì vậy, các thuật toán học tập hợp phải được thiết kế đặc biệt để xử lý hiệu quả vấn đề mất cân bằng dữ liệu. Sự kết hợp giữa bộ phân lớp tổng hợp với các kỹ thuật xử lý mất cân bằng dữ liệu như Resampling để giải quyết vấn đề mất cân bằng lớp đã được chứng minh có kết quả khả quan. Ngoài ra, ý tưởng kết hợp nhiều bộ phân lớp chính có thể làm giảm hiện tượng overfitting.

Lấy mẫu dưới (Undersampling) liên quan đến một nhóm các kỹ thuật được thiết kế để cân bằng phân phối các lớp cho tập dữ liệu phân loại có phân phối lớp lệch. Kỹ thuật lấy mẫu dưới loại bỏ các mẫu dữ liệu khỏi tập dữ liệu huấn luyện thuộc về lớp đa số để cân bằng việc phân phối lớp tốt hơn, ví dụ như giảm độ lệch từ 1:100 xuống 1:10, 1:2 hoặc thậm chí là lớp phân phối 1:1. Điều này khác với việc cố gắng lấy thêm các mẫu dữ liệu vào lớp thiểu số để giảm bớt sự sai lệch phân phối lớp của kỹ thuật lấy mẫu quá mức (oversampling). Phương pháp lấy mẫu dưới có thể được sử dụng trực tiếp trên tập dữ liệu huấn luyện mà sau đó có thể được sử dụng để phù hợp với mô hình học máy. Thông thường, các phương pháp lấy mẫu dưới được sử dụng cùng với kỹ thuật lấy mẫu quá mức cho lớp thiểu số và sự kết hợp này thường mang lại hiệu suất cao hơn so với việc chỉ sử dụng lấy mẫu quá mức hoặc lấy mẫu dưới trên tập dữ liệu huấn luyện.

## Khó khăn và thách thức

Trong lúc thực hiện dự án này, chúng tôi đã gặp phải một số khó khăn nhất định việc phân lớp dữ liệu mất cân bằng là một vấn đề đặc biệt khó. Khó khăn của vấn đề này được kết hợp bởi các thuộc tính như kích thước tập dữ liệu, nhiễu nhãn và phân bố dữ liệu. Về cơ bản, trở ngại lớn nhất của chúng tôi chính là độ khó của dữ liệu. Tiếng Việt là một ngôn ngữ đa dạng và phong phú cả về ngữ pháp và ngữ nghĩa. Một từ tùy vào từng ngữ cảnh có thể mang các nghĩa khác nhau. Tiếng Việt còn có dấu của các từ do đó chỉ cần thiếu một dấu thì nghĩa của từ cũng sẽ thay đổi. Đặc thù của dữ liệu là các bình luận của người dùng nên số lượng từ vựng phong phú, từ ngữ không được trau chuốt dẫn đến thiếu chuẩn mực về ngữ pháp và ngôn ngữ. Các bình luận của người dùng có thể chứa các từ viết tắt (ví dụ từ “không” có thể được viết tắt thành “k”, “ko”, “hok”,…), các từ mượn từ tiếng anh (“ok”, “good”,..) và có thể chứa các ẩn ý không dễ phát hiện bằng từ vựng. Có những bình luận mang tính trung lập như “chất lượng phù hợp với giá tiền”, “tiền nào của nấy” hoặc những bình luận không nhắc rõ đến các khía cạnh như “dùng ok”, “tốt”, “cũng được”,... đòi hỏi sự thống nhất về quan điểm để tạo ra nguyên tắc gán nhãn để dữ liệu được gán nhãn chính xác và đồng đều nhất.

## Các định nghĩa cơ bản

* Khía cạnh (aspect): Là các đặc trưng, thuộc tính của một sự vật hiện tượng nào đó.
* Đánh giá (review): Là các bình luận của người dùng đã mua sản phẩm trên các trang thương mại điện tử đánh giá về chất lượng sản phẩm, dịch vụ, trải nghiêm mua hàng.
* Phân lớp đa nhãn (Multi-label classification): là một loại biến thể khác của phân lớp văn bản, trong đó một mẫu dữ liệu có thể được gán với nhiều nhãn.
* Dữ liệu mất cân bằng (Imbalanced data): Là một vấn đề trong việc phân lớp trong đó số lượng quan sát trên mỗi lớp không được phân phối đồng đều tạo thành các lớp đa số và thiểu số.
* Lớp đa số (majority class): Là các lớp chiếm tỉ lệ lớn trong tập dữ liệu.
* Lớp thiểu sổ (minority class): Là các lớp chiếm tỉ lệ nhỏ hơn trong tập dữ liệu.
* Miền (Domain): Là một đối tượng mô hình độc lập mà ta có thể sử dụng để gán nhanh các thuộc tính cho các đặc tính hoặc cột.
* Dữ liệu huấn luyện (training data): Là một tập hợp dữ liệu ban đầu được sử dụng để giúp một chương trình hiểu cách áp dụng các công nghệ để học và tạo ra các kết quả. Nó có thể được bổ sung bởi các bộ dữ liệu tiếp theo được gọi là bộ xác nhận và kiểm thử. Dữ liệu huấn luyện còn được gọi là tập huấn luyện, tập dữ liệu đào tạo hoặc tập dữ liệu học.
* Dữ liệu kiểm thử (test data): Là một tập hợp các dữ liệu/quan sát đã được xác định cụ thể được sử dụng để đánh giá hiệu suất của mô hình bằng cách sử dụng một số thước đo hiệu suất.
* Dữ liệu văn bản (text data): Bao gồm các tài liệu có thể đại diện cho các từ, câu hoặc thậm chí đoạn văn bản tự do. Tính chất phi cấu trúc vốn có và tính lộn xộn của dữ liệu văn bản khiến các phương pháp học máy khó làm việc trực tiếp trên dữ liệu văn bản thô hơn.

# Xây dựng dữ liệu

Nhằm phục vụ cho bài nghiên cứu, chúng tôi đã thu thập dữ liệu từ hai trang thương mại điện tử lớn và nổi tiếng nhất tại Việt Nam là Tiki. Tiki và Shopee có số lượng truy cập/tháng nhiều nhất (tính trong đầu quý I/2020) vì sự thuận tiện, nhanh chóng và giá cả cạnh tranh so với thị trường. Người dùng ở hai sàn thương mại điện tử này thường để lại đánh giá về sản phẩm và trải nghiệm mua hàng sau khi nhận hàng. Hơn nữa, tất cả các đánh giá của khách hàng đều công khai và người bán không thể sửa hoặc xóa các đánh giá đó, vì vậy đây sẽ là nguồn dữ liệu lớn và đa dạng để thu thập. Trong bài nghiên cứu này, chúng tôi tập trung vào hai miền dữ liệu là Công nghệ (cụ thể là máy tính bảng và điện thoại) và Mẹ & Bé (gồm 2 loại sản phẩm chính là bỉm và sữa). Đây là các sản phẩm được quan tâm nhất trong hai miền dữ liệu, có lượt mua, lượt đánh giá rất cao.

## Công nghệ sử dụng

Chúng tôi đã chọn Scrapy để thu thập dữ liệu. Scrapy là một framework để thu thập dữ liệu các trang web và trích xuất cấu trúc dữ liệu có thể được sử dụng cho nhiều ứng dụng, như khai phá dữ liệu, xử lý thông tin hoặc lưu trữ lịch sử. Mặc dù Scrapy ban đầu được thiết kế để tìm kiếm web, nó cũng có thể được sử dụng để trích xuất dữ liệu bằng cách sử dụng API, hoặc như một trình thu thập dữ liệu web thông thường.

## Quá trình xây dựng dữ liệu

### Thu thập dữ liệu

Đầu tiên, chọn XHR trong Developer Tool để lọc các request AJAX, và xem xét hết tất cả các response để tìm xem cái nào sẽ trả những dữ liệu liên quan tới đánh giá sản phẩm: tên người dùng viết đánh giá, nội dung đánh giá, thời gian viết đánh giá,… Đối với Shopee, địa chỉ API đó có dạng như sau: <https://shopee.vn/api/reviews>

Sau khi đã lấy hết được đánh giá của một sản phẩm, ta sẽ tìm itemid và shopid của các sản phẩm khác để thay vào giá trị của itemid và shopid trong link API trên. Một cách để tìm được nhanh các itemid và shopid là thu thập dữ liệu đó qua link API chứa thông tin về các sản phẩm liên quan: [https://shopee.vn/api/values](https://shopee.vn/api/v4/recommend/recommend?catid=84&item_card=3&itemid=9512629620&limit=24&offset=0&shopid=65589552&bundle=product_detail_page)

Tiếp theo, chúng tôi sẽ lọc ra những đánh giá có thể đem lại thông tin bằng cách loại bỏ những đánh giá không chứa nội dung, những từ ít có giá trị trong việc huấn luyện mô hình. Sau đó, khi đã có được tập dữ liệu hoàn chỉnh, chúng tôi chuyển sang công việc tiếp theo là gán nhãn cho tập dữ liệu đó.

### Gán nhãn

1. **Định nghĩa tập nhãn**

Sau khi khảo sát dữ tập liệu thu được, chúng tôi đã chọn ra được những khía cạnh đặc trưng nhất có thể bao quát được đa số dữ liệu, và các đặc trưng đó sẽ ứng với tên các nhãn:

* Miền dữ liệu Tech:
* Ship
* Giá
* Mẫu mã
* Cấu hình
* Hiệu năng
* Chính hãng
* Phụ kiện
* Dịch vụ
* Miền dữ liệu Mẹ & Bé:
* Ship
* Giá
* Chính hãng
* Chất lương
* Dịch vụ
* An toàn

Tuy nhiên, để phục vụ thêm cho nhiệm vụ phân cực quan điểm, mỗi nhãn sẽ được chia ra thành 2 nhãn nhỏ khác là nhãn + (chỉ quan điểm, cảm xúc tích cực) và – (chỉ quan điểm, cảm xúc tiêu cực).

#### **Guideline gán nhãn**

Để có thể thống nhất và không gây ra mâu thuẫn cho các dữ liệu, chúng tôi đã tạo ra những nguyên tắc gán nhãn như sau:

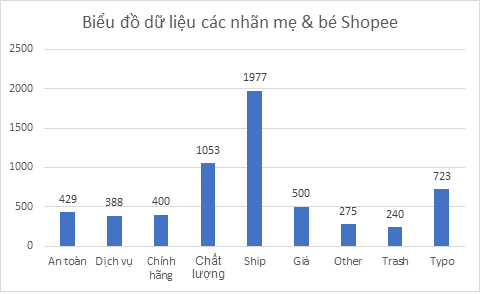
| Miền dữ liệu | Loại nhãn | | Keywords | Độ ưu tiên |
| --- | --- | --- | --- | --- |
| Bỉm/sữa | ship | + | giao hàng nhanh, đóng gói cẩn thận, freeship | giao hàng(4), đóng gói (3), phí ship (2), còn lại(1) |
| - | ship chậm, trễ, không giao, đóng gói không cẩn thận, ... |
| giá | + | giá hợp lí, phải chăng, rẻ, vừa tiền,… |  |
| - | giá cao, mắc,... |
| chất lượng | + | thấm tốt, .... | bỉm: độ thấm(5), độ mềm(2), dày (2), còn lại(1) |
| - | có chữ bị, chữ quá đi vs tính từ trung lập |
| dịch vụ | + | phản hồi nhanh. nhiệt tình, ... |  |
| - | thiếu quà tặng, trả lời lâu,… |
| chính hãng | + | chính hãng,… |  |
| - | dởm, lởm, fake,... |
| an toàn | + | mới, còn date, an toàn, đảm bảo,... |  |
| - | cũ, dị ứng, ngứa, ngộ độc, mẩn, hăm... |
| Điện thoại / Máy tính bảng | ship | + | giao hàng nhanh, đóng gói cẩn thận, freeship | giao hàng(4), đóng gói (3), phí ship (2), còn lại(1) |
| - | ship chậm, trễ, không giao, đóng gói không cẩn thận, ... |
| giá | + | giá hợp lí, phải chăng, rẻ, vừa tiền,… |  |
| - | giá cao, mắc,... |
| mẫu mã | + | đẹp, sang, mới, ... |  |
| - | xấu, cũ, xước, vỡ, nứt, ... |
| hiệu năng | + | khỏe, mượt, nhanh, |  |
| - | lag, giật, chậm, đơ,… |
| chính hãng | + | chính hãng,... |  |
| - | rởm, lởm, fake,... |
| phụ kiện | + | đầy đủ, đi kèm, được tặng, ... |  |
| - | thiếu, không có,… |
| dịch vụ | + | phản hồi nhanh. nhiệt tình, ... |  |
| - | thiếu quà tặng, trả lời lâu,..... |
| cấu hình | + | pin trâu, khỏe,..; cam sắc nét, chụp ảnh đẹp,....; màn hình sáng, nét,...; cảm ứng mượt,...; | Nếu có có từ 2 đặc điểm mang cảm xúc tiêu cực thì gán nhãn - |
| - | pin chai, có chữ “bị”, ... |

Bảng ‑. Các nguyên tắc gán nhãn

### Kết quả thu được

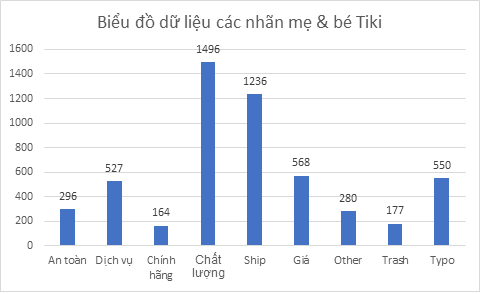
Thống kê tập dữ liệu thu được:

Dữ liệu sau khi thu thập và được gán nhãn trên 4 tập dữ liệu được chúng tôi thể hiện qua các biểu đồ dưới đây:



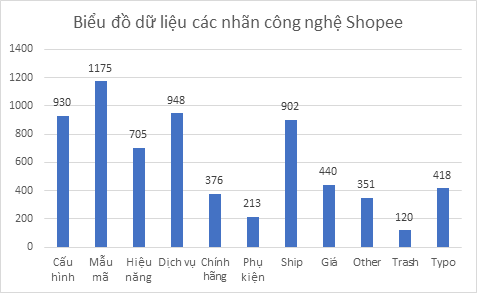
**Hình 2‑1. Biểu đồ phân bố các nhãn mẹ & bé shopee**

Đối với dữ liệu mẹ & bé Shopee, sự phân bố giữa các nhãn tương đối lệch. Trong đó nhãn Ship chiếm tỉ lệ cao nhất, chứng tỏ người dùng Shopee rất quan tâm đến thời gian giao hàng và chất lượng đóng gói hàng; nhãn Trash và Other chiếm tỉ lệ thấp nhất một phần do một số người dùng spam cho đủ kí tự để nhận xu của Shopee.



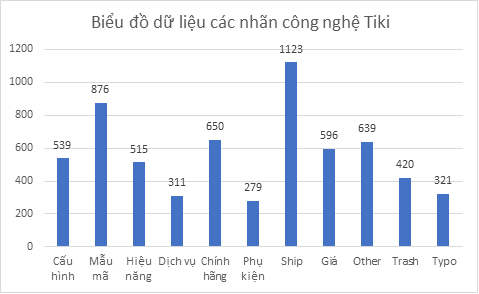
**Hình 2‑2. Biểu đồ phân bố các nhãn mẹ & bé tiki**

Dữ liệu các nhãn mẹ & bé Tiki cũng tương đối lệch so với mẹ & bé Shopee. Trong đó hai nhãn Chất lượng và Ship có số lượng nhiều nhất và nhãn Chính hãng có số lượng ít nhất do người dùng bên Tiki quan tâm về vấn đề chất lượng hơn trong khi vấn đề chính hãng được Tiki đảm bảo nên ít được nhắc đến.



**Hình 2‑3. Biểu đồ phân bố các nhãn công nghệ shopee**

Dữ liệu phân bố các nhãn công nghệ Shopee có sự phân bố khá đồng đều chia ra làm hầu hết cao và một phần thấp. Trong đó, Mẫu mã, Cấu hình, Dịch vụ và Ship là những nhãn chiếm tỉ lệ cao nhất, cho thấy người dùng quan tâm đến các vấn đề chất lượng sản phẩm và dịch vụ CSKH của Shopee.



**Hình 2‑4. Biểu đồ phân bố các nhãn công nghệ tiki**

Đây là dữ liệu được phân bố đều nhất trong các tập dữ liệu. Tuy nhiên, nhãn Ship có số lượng cao nhất và giảm dần đều về các nhãn còn lại. Người dùng Tiki quan tâm rất lớn đến vấn đề giao hàng. Nhãn Phụ kiện được nhắc đến ít nhất.

# Các nghiên cứu liên quan

## Các mô hình học máy cơ bản

### Naïve Bayes

Phân lớp Naïve Bayes là một phân lớp theo xác suất, dựa trên lý thuyết Bayes, giả định rằng mỗi đối tượng đều đóng góp độc lập và bình đẳng cho lớp đích. Phân lớp NB giả định rằng mỗi đặc trưng độc lập và không liên quan đến các đặc trưng khác, sao cho mỗi đặc trưng góp phần vào xác suất củ một mẫu thuộc một lớp cụ thể một cách độc lập và bình đẳng.

Việc phân lớp được tiến hành bằng cách suy ra hậu nghiệm cực đại là *P(Ci | X)* với giả thiết trên áp dụng cho định lý Bayes. Giả định này làm giảm đáng kể chi phí tính toán bằng cách chỉ đếm phân phối lớp. Mặc dù giả định không hợp lệ trong hầu hết các trường hợp vì các thuộc tính phụ thuộc, nhưng đáng ngạc nhiên là Naïve Bayes đã thực hiện một cách ấn tượng.

Naïve Bayes là một thuật toán rất đơn giản để thực hiện và hầu hết các trường hợp đã thu được kết quả tốt. Nó có thể dễ dàng mở rộng cho các bộ dữ liệu lớn hơn vì nó cần thời gian tuyến tính, thay vì bằng phép xấp xỉ lặp lại tốn kém được sử dụng cho nhiều bộ phân lớp khác.

Naïve Bayes có thể mắc phải vấn đề được gọi là xác suất bằng không. Khi xác suất có điều kiện bằng 0 đối với một thuộc tính cụ thể, nó không thể đưa ra dự đoán hợp lệ. Điều này cần được cố định một cách rõ ràng bằng cách sử dụng công cụ ước tính Laplacian.

### Support Vector Machines

Máy véc-tơ hỗ trợ (SVM) là một tập hợp các phương pháp học có giám sát được sử dụng để phân loại, hồi quy và phát hiện ngoại lệ. Tất cả những điều này đều là những tác vụ phổ biến trong học máy.

Có những loại SVM cụ thể mà có thể sử dụng cho các vấn đề học máy cụ thể, như hồi quy véc-tơ hỗ trợ (SVR) là một phần mở rộng của phân loại véc-tơ hỗ trợ (SVC).

Bộ phân loại SVM tuyến tính đơn giản hoạt động bằng các tạo ra một đường thẳng giữa hai lớp. Điều đó có nghĩa là tất cả điểm dữ liệu ở một bên của đường thẳng sẽ đại diện cho một lớp và các điểm dữ liệu ở phía bên kia của đường thẳng sẽ được đưa vào một lớp khác, tức là có thể có vô số đường thẳng để lựa chọn.

SVM khác với các thuật toán phân lớp khác vì cách chọn ranh giới quyết định tối đa hoá khoảng cacgs từ các điểm dữ liệu gần nhất của tất cả các lớp. Ranh giới quyết định được tạo bởi SVM được gọi là bộ phân lớp lề tối đa hoặc siêu phẳng lề tối đa.

Điều làm cho thuật toán SVM tuyến tính tốt hơn một số thuật toán khác, như K-láng giềng gần nhất, là nó chọn đường thẳng tốt nhất để phân loại các điểm dữ liệu. Nó chọn đường thẳng phân tách dữ liệu và càng xa các điểm dữ liệu từ các điểm dữ liệu gần nhất càng tốt.

### Random Forest

Rừng ngẫu nhiên là một thuật toán học có giám sát. “Forset” mà nó xây dựng là một tập hợp các cây quyết định, thường được huấn luyện bằng phương pháp bagging. Ý tưởng chung của phương pháp bagging là sự kết hợp của các mô hình học sẽ làm tăng kết quả chung. Nói một cách đơn giản, rừng ngẫu nhiên xây dựng nhiều câu quyết định và kết hợp chúng lại với nhau để có được dự đoán chính xác và ổn định hơn.

Một lợi thế lớn của rừng nhẫu nhiên là nó có thể sử dụng cho cả bài toán phân lớp và hồi quy, vốn tạo nên phần lớn các hệ thống học máy hiện tại.

Rừng ngẫu nhiên có các siêu tham số gần giống như cây quyết định hoặc bộ phân lớp bagging. May mắn thay, không cần phải kết hợp cây quyết định với bộ phân lớp bagging vì có thể dễ dàng sử dụng lớp phân lớp của rừng ngẫu nhiên.Với rừng ngẫu nhiên, có thể giải quyết các tác vụ hồi quy bằng cách sử dụng bộ hồi quy của thuật toán.

Rừng ngẫu nhiên bổ sung thêm tính ngẫu nhiên cho mô hình, trong khi trồng cây. Thay vì tìm kiếm đặc trưng quan trọng nhất trong khi tách một nút, nó tìm kiếm đặc trưng tốt nhất trong số một tập hợp con ngẫu nhiên của các đặc trưng. Điều này dẫn đến sự đa dạng rỗng rãi và thường dẫn đến một mô hình tốt hơn.

Trong rừng ngẫu nhiên, chỉ một tập hơn con ngẫu nhiên của các đặc trưng được xem xét bởi thuật toán tách một nút. Thậm chí có thể làm cho các cây trở nên ngẫu nhiên hơn bằng cách sử dụng thêm các ngưỡng ngẫu nhiên cho từng tính năng thay vì tìm kiếm các ngưỡng tốt nhất có thể (giống như cây quyết định thông thường).

### Logistic Regression

Hồi quy Logistic là một thuật toán học máy được sử dụng cho các bài toán phân lớp, nó là một thuật toán phân tích dự đoán và dựa trên khái niệm xác suất.

Có thể gọi Hồi quy Logistic là mô hình Hồi quy tuyến tính nhưng Hồi quy Logistic sử dụng một hàm chi phí phức tạp hơn, hàm chi phí này có thể được định nghĩa là “hàm sigmoid” hoặc còn được gọi là “hàm logistic” thay vì hàm tuyến tính.

Giả thuyết của Hồi quy Logistic có xu hướng giới hạn hàm chi phí trong khoảng từ 0 đến 1. Do đó, các hàm tuyến tính không thể hiện được nó vì nó có thể có giá trị lớn hơn 1 hoặc nhỏ hơn 0, điều này không khả thi theo giả thuyết của Hồi quy Logistic.

Hồi quy logistic là một kỹ thuật thống kê được sử dụng để dự đoán xác suất phản hồi nhị phân dựa trên một hoặc nhiều biến độc lập. Nó có nghĩa là, với một số yếu tố nhất định, hồi quy logistic được sử dụng để dự đoán một kết quả có hai giá trị như 0 hoặc 1, đạt hoặc không đạt, có hoặc không.

Xác suất được ước tính bằng cách sử dụng hàm logistic / sigmoid. Đồ thị của hàm sigmoid là một đường cong chữ "S".



Hình ‑. Đồ thị hàm Sigmoid

Biểu thức toán học được đưa ra bởi

Trong đó , ở đây là hệ số hồi quy của mô hình và được tính bằng Maximum Likelihood Estimation và là các đặc trưng hoặc các biến độc lập. tính toán xác suất của kết quả nhị phân và sử dụng xác suất để phân loại điểm dữ liệu đã cho thành một trong hai loại.

## Các nghiên cứu liên quan về các kỹ thuật xử lý mất cân bằng dữ liệu

### Bagging Ensemble

Thuật toán Bagging (viết tắt của “Bootstrap Aggregating”) được Breiman (1996) giới thiệu như một phương pháp bỏ phiếu. Trên thực tế, chúng tôi không thể xây dựng các mô hình độc lập hoàn toàn cho bagging bởi vì nó đòi hỏi quá nhiều dữ liệu. Vì vậy, như tên đầy đủ của nó, bootstrap aggregating, bagging dựa trên các thuộc tính gần đúng của các mẫu dữ liệu bootstrap (đại diện và độc lập) để xây dựng các mô hình gần như độc lập.

Bootstrapping là một kĩ thuật lấy mẫu, trong đó chúng ta tạo ra các tập con của các quan sát từ tập dữ liệu ban đầu bằng cách vẽ ngẫu nhiên từ nó. Do đó, mỗi tập dữ liệu bootstrap mới hoạt động như một dữ liệu độc lập được rút ra từ phân phối thực sự. Do đó, chúng tôi có được sự đa dạng kỳ vọng bằng cách sử dụng các bộ dữ liệu mới khác nhau. Bagging sau đó sử dụng bản sao khởi động của dữ liệu huấn luyện ban đầu để đào tạo một modem cơ sở. Quá trình này được lặp lại để tạo ra nhiều mô hình cơ sở. Nói ngắn gọn, dữ liệu bootstrap gần như độc lập và được phân phối giống nhau và các mô hình cơ sở học được cũng như vậy. Đầu ra của ensemble không thay đổi kết quả kỳ vọng nhưng giảm độ sai của nó.

Chiến lược bootstrapping ban đầu cho phép chúng ta vẽ các thực thể ngẫu nhiên từ dữ liệu huấn luyện ban đầu với sự thay thế, nghĩa là một số dữ liệu có thể được lặp lại trong dữ liệu được sao chép mới nhưng một số khác có thể bị bỏ qua. Trong trường hợp bagging, bất kỳ thực thể nào có cùng xác suất hiện trong một tập dữ liệu mới.

Bagging ensemble đã được sử dụng để cải thiện hiệu suất và sự ổn định của bộ phân lớp đa nhãn.

### Phương pháp boosting

Boosting là một chiến lược nhận được tương đối ít sự chú ý trong các nghiên cứu. Các nghiên cứu về phương pháp này đề cập đến việc giới thiệu các kỹ thuật boosting mới được thiết kế đặc biệt cho vấn đề mất cân bằng lớp. DataBoost-IM, do Gou và Viktor đề xuất (Guo & Viktor 2004) kết hợp phương pháp boosting và lấy mẫu dữ liệu để cải thiện hiệu suất phân lớp. SMOTEBoost (Chawla et al. 2003) là một kỹ thuật khác kết hợp lấy mẫu dữ liệu (SMOTE) và boosting để bù đắp cho dữ liệu bị mất cân bằng.

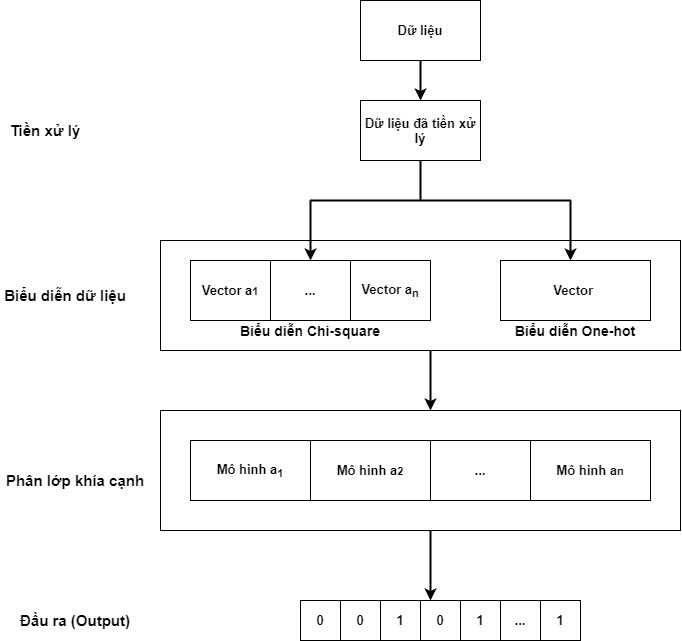
RareBoost (Joshi, Kumar và Agarwal 2001) đã sửa đổi quá trình boosting bằng cách điều chỉnh trọng số khác nhau trên các lớp của các mẫu dữ liệu được phân loại sai.

### Resampling

Lấy mẫu dữ liệu đã nhận được sự quan tâm đáng kể trong ngành nghiên cứu khai thác dữ liệu. Drummond và Holte (Drummond & Holte 2003) đã so sánh hai phương pháp là lấy mẫu trên và lấy mẫu dưới kết hợp sử dụng cây quyết định C4.5. Kết quả cho thấy lấy mẫu dưới hiệu quả hơn lấy mẫu trên để cải thiện hiệu suất của các mô hình được xây dựng bằng C4.5. Tuy nhiên, nghiên cứu của Maloof (Maloof 2003) cho thấy rằng việc lấy mẫu trên và lấy mẫu dưới gần như tương đương khi sử dụng C5.0 (mô hình kế nhiệm của C4.5) và Naive Bayes. Ngoài việc sử dụng lấy mẫu trên và dưới ngẫu nhiên, một số các kỹ thuật lấy mẫu dữ liệu thông minh hơn đã được đề xuất để lấy mẫu dữ liệu theo cách có lợi cho bộ phân lớp. Barandela và cộng sự. (Barandela et al. 2004) và Han et al. (Han, Wang, & Mao 2005) đã kiểm tra hiệu suất của một số kỹ thuật lấy mẫu dữ liệu thông minh như SMOTE, borderline-SMOTE và Wilson’s Ending. Van Hulse và cộng sự. (Van Hulse, Khoshgoftaar, & Napolitano 2007) đã kiểm tra hiệu suất của bảy kỹ thuật lấy mẫu khác nhau bao gồm cả kỹ thuật ngẫu nhiên và kỹ thuật thông minh.

# Mô hình đề xuất

## Mô hình phân lớp quan điểm



Hình ‑. Mô hình phân lớp quan điểm

Mô hình phân lớp quan điểm trên bao gồm 4 pha: Tiền xử lý, Biểu diễn dữ liệu, Phân lớp khía cạnh và Đầu ra. Đầu tiên dữ liệu sẽ được xử lý qua pha tiền xử lý sau đó được đưa vào pha Biểu diễn dữ liệu bằng cách biểu diễn Chi-square hoặc cách biểu diễn One-hot thành các vector. Sau pha này, các vector dữ liệu được đưa vào các mô hình học máy để phân lớp khía cạnh để được đầu ra là các vector gồm 2 lớp 0 và 1, trong đó lớp 1 là dữ liệu có khía cạnh đó, lớp 0 là dữ liệu không có khía cạnh đó.

### Pha 1: Tiền xử lý dữ liệu

Ví dụ trong câu: “shop đóng gói oke và giao hàng nhanhh ssssssssss. ,…”

Quá trình tiền xử lý gồm 6 bước:

* Tách từ: “đóng gói” -> “đóng\_gói”
* Loại bỏ kí tự lặp: “nhanhh” -> “nhanh”
* Xóa những từ có độ dài > 7: xóa “ssssssssss”
* Xóa các từ không có nhiều ý nghĩa: xóa từ “và”
* Thay thế các từ acronym: “oke” -> “ok”
* Xóa các dấu kí tự: xóa “. ,…”

Câu sau khi được tiền xử lý: “shop đóng\_gói ok giao\_hàng nhanh”

### Pha 2: Biểu diễn và trích chọn đặc trưng dữ liệu

**One-hot**

One-hot encoding là cách biểu diễn từng giá trị thành các đặc trưng nhị phân chỉ chứa 0 hoặc 1. Trong bài nghiên cứu này, chúng ta coi mỗi dữ liệu đầu vào là các câu, sẽ được biến đổi thành 1 vectơ có kích thước n, với n là số lượng từ xuất hiện trong tập từ vựng đã tạo ở trên.

Tuy nhiên, việc sử dụng vec-tơ one-hot dường như sẽ không mang lại được kết quả cao vì tập từ vựng dùng để biểu diễn các từ có thể sẽ phải rất lớn (lên đến hàng triệu từ). Qua khảo sát, chúng tôi lựa chọn phương pháp: Áp dụng chi2 kết hợp one-hot để biểu diễn dữ liệu đầu vào cho bài toán.

**Chi-square**

Kiểm định thống kê Chi-quare (*c*2) đã được chấp nhận rộng rãi như một kiểm định giả thuyết thống kê để đánh giá mức độ phụ thuộc giữa hai biến. Trong xử lý ngôn ngữ tự nhiên, phép thử Chi-square thường được áp dụng để kiểm tra sự độc lập giữa sự xuất hiện của thuật ngữ và sự xuất hiện của lớp. Nó thường được sử dụng như một phương pháp trích chọn đặc trưng trong xử lý ngôn ngữ tự nhiên. Công thức dưới đây được sử dụng để xếp hạng các thuật ngữ xuất hiện trong tập từ vựng.

trong đó *et*và *ec* là các biến nhị phân được xác định trong bảng tương quan; *et =* 1có nghĩa là tài liệu chứa thuật ngữ *t* và *et =* 0 có nghĩa là tài liệu không chứa thuật ngữ *t; ec =* 1có nghĩa là tài liệu thuộc lớp *c* và *ec =* 0có nghĩa là tài liệu không thuộc lớp *c; N* là tần số quan sát được trong và *E* là tần số kỳ vọng. Ví dụ, *N*11 là tần suất quan sát được các tài liệu xuất hiện trong lớp *c* chứa thuật ngữ *t*; *E*11 là tần suất kỳ vọng của *t* và *c* xuất hiện cùng nhau trong một tài liệu giả sử thuật ngữ và lớp là độc lập. Giá trị cao hơn của *c*2 chỉ ra rằng thuật ngữ *t* và lớp *c* là phụ thuộc, làm cho thuật ngữ *t* trở thành một đặc trưng hữu ích vì sự xuất hiện của *t* có nghĩa là tài liệu có khả năng được nhìn thấy trong lớp *c* nhiều hơn.

Sử dụng tính chất của *c*2 cao hơn giá trị *c*2 của thuật ngữ *t* cho biết khả năng xuất hiện trong lớp *c* cao hơn, chúng tôi sử dụng *c*2 để gán trọng số cho các từ ngữ cảnh trong mô hình phân lớp đa nhãn. Khía cạnh quan trọng là thống kê của những từ có *c*2 cao hơn có xu hướng là từ khoá để xác định lớp. Vì vậy, chúng tôi đang sử dụng kiểm định thống kê Chi-square để chọn từ vựng đặc biệt phục vụ cho nhiệm vụ xác định lớp các câu ngắn. Cơ sở lý luận của chúng tôi là trong mô hình phân lớp của chúng tôi, các từ có trọng số theo số liệu thống kê *c*2 của chúng: Những từ có thể có giá trị đối với nhiệm vụ phân lớp có trọng số lớn hơn do đó giảm sự xáo trộn của những từ nhiễu vốn không hữu ích với nhiệm vụ sau này.

Thay vì sử dụng phương pháp vec-tơ one-hot như cơ bản (tạo 1 tập từ vựng chứa tất cả các từ trong dữ liệu đầu vào để ánh xạ lên mỗi vec-tơ cho từng câu), dựa trên điểm *c*2 có được, chúng tôi sẽ lựa chọn ngưỡng điểm *c*2 phù hợp để lọc lấy những từ quan trọng trong từng nhãn khía cạnh của dữ liệu và cho vào tập từ vựng của khía cạnh đó, sau đó sử dụng chính điểm *c*2 có được của từng từ để thay vào giá trị của từ đó trong vecto one-hot. Như vậy, với mỗi khía cạnh của của bài toán, một câu đầu vào sẽ được biểu diễn là 1 vec-tơ có độ dài bằng số lượng từ trong tập từ vựng của khía cạnh đó. Từ nào xuất hiện trong câu và có trong tập từ vựng thì giá trị của từ đó trong vec-tơ câu sẽ có trọng số là *c*2 của nó.

### Pha 3: Mô hình phân lớp

**Phân lớp đa nhãn có giám sát**

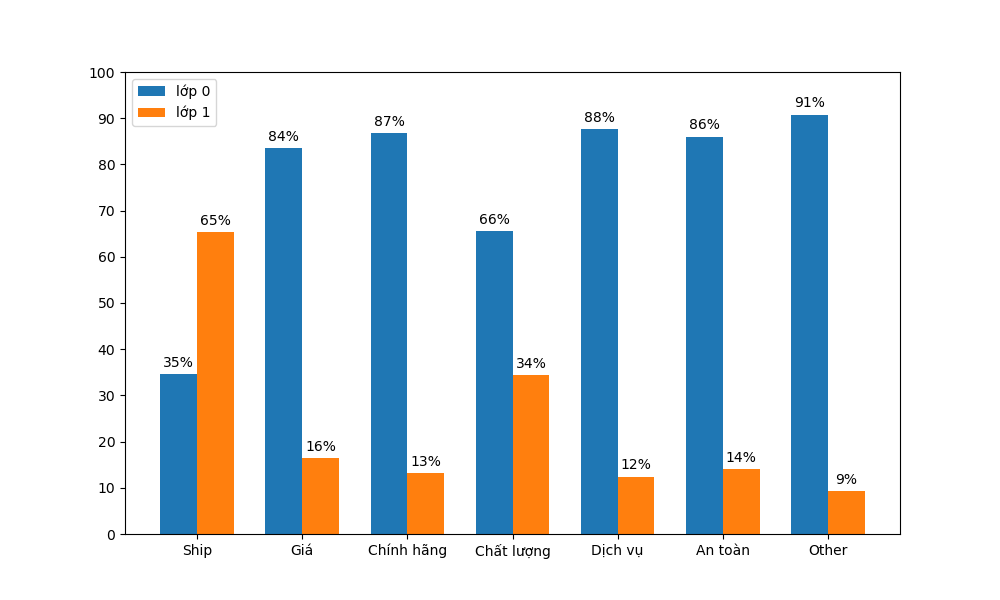
Gọi là tài liệu đầu vào đã gán nhãn với tập *L* gồm *q* nhãn, tức là *L* ={*l1, l2, … , lq*}, trong đó mỗi tài liệu trong được gán một tập hợp con không rỗng của các nhãn *label() .* Nhiệm vụ của phân lớp đa nhãn là xây dựng hàm phân lớp , do đó, với một tài liệu chưa được gán nhãn mới *du,* hàm xác định một tập hợp các nhãn có liên quan .

Với từng mẫu dữ liệu, chúng tôi có nhãn *Y* là một mảng với *n* phần tử nhị phân, với *n* là số lượng nhãn của dữ liệu. Phần tử *Y[i]* sẽ có giá trị là *0* nếu mẫu dữ liệu đó không bao gồm nhãn *i* và có giá trị là *1* nếu mẫu dữ liệu đó bao gồm nhãn *i*. Với mô hình phân lớp đa nhãn, chúng tôi sẽ huấn luyện *n* mô hình. Mỗi một mô hình chính là một bộ phân lớp nhị phân, nhãn của từng mẫu dữ liệu là *1* hoặc *0* với *1* tương ứng với sự xuất hiện của nhãn đó và *0* tương ứng với nhãn đó không xuất hiện trong mẫu dữ liệu đó.

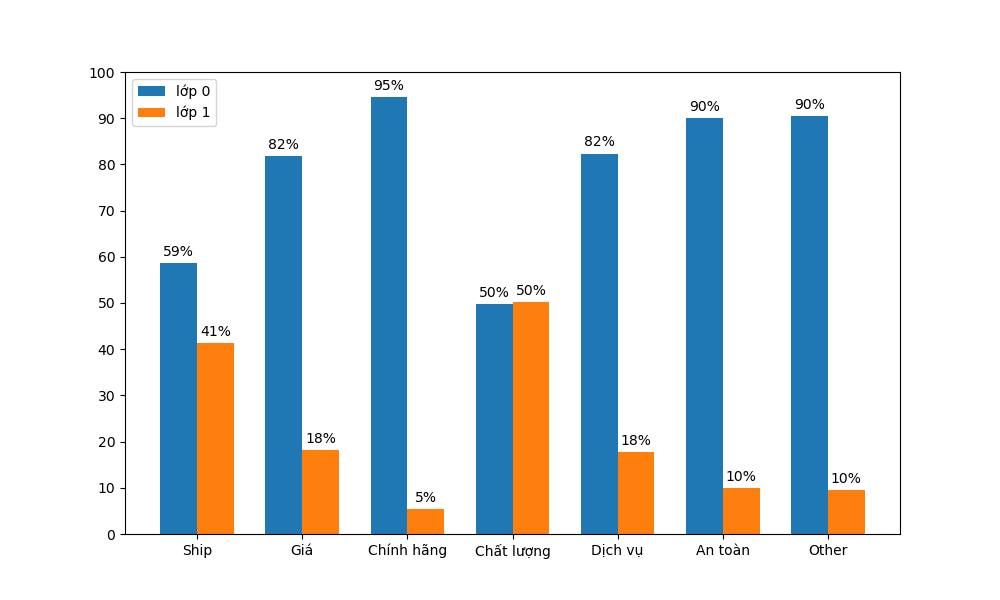
## Kiến trúc THANOS

### Động lực

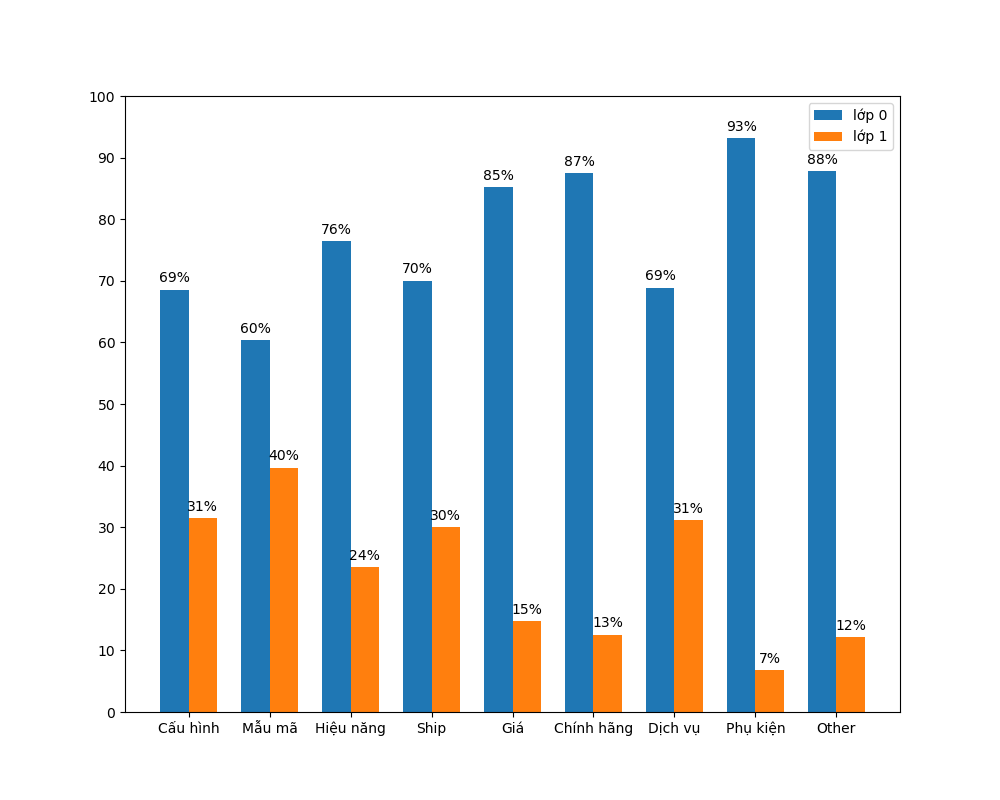
Dưới đây là phân bố lớp từng nhãn trong các tập dữ liệu:



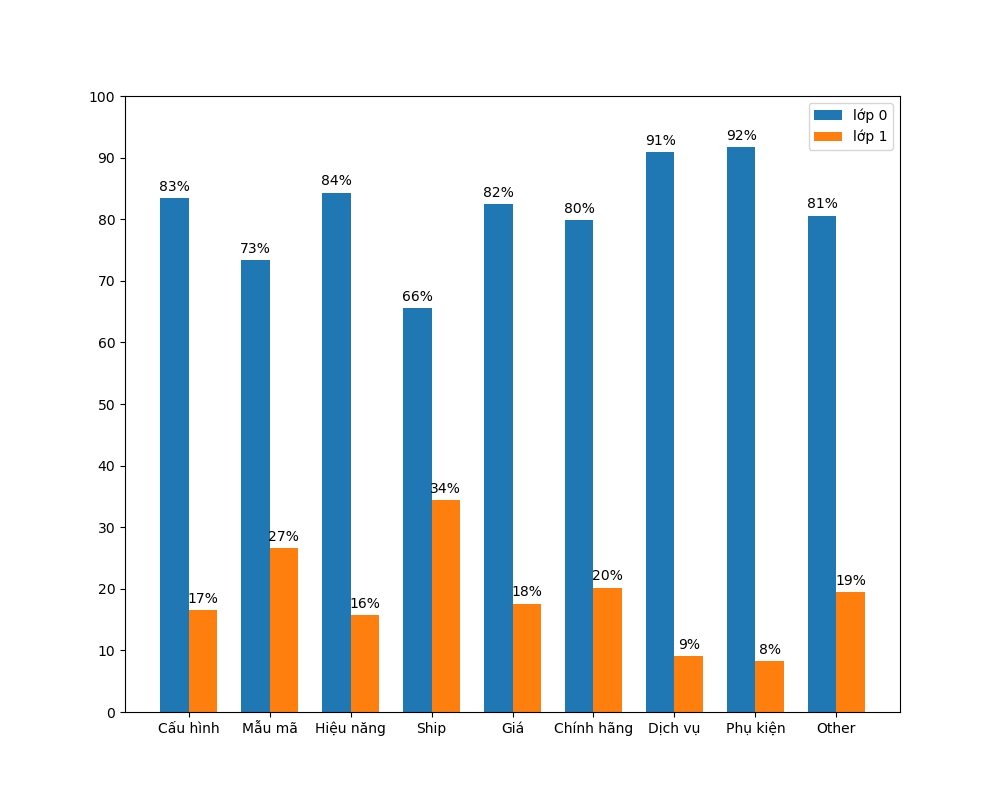
Hình ‑. Phân bố lớp của từng nhãn trong dữ liệu mẹ & bé shopee



Hình ‑. Phân bố lớp của từng nhãn trong dữ liệu mẹ & bé tiki



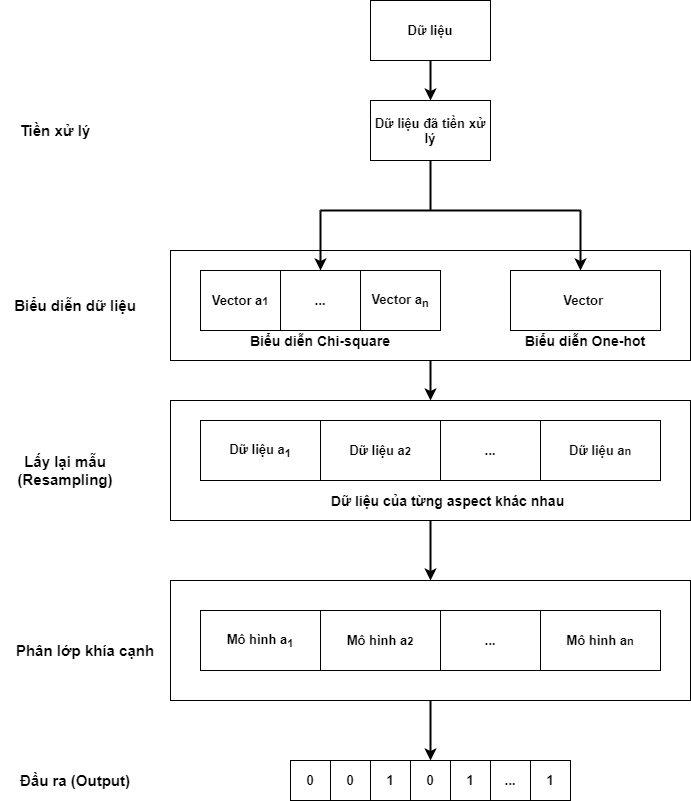
Hình ‑. Phân bố lớp của từng nhãn trong dữ liệu công nghệ shopee



Hình ‑. Phân bố lớp của từng nhãn trong dữ liệu công nghệ tiki

Ta có thể dễ dàng nhận thấy phân bố lớp trong nhiều nhãn đang bị mất cân bằng rõ rệt. Nếu chúng ta sử dụng trực tiếp dữ liệu mất cân bằng như vậy để huấn luyện mô hình thì kết quả dự đoán không đạt yêu cầu vì mô hình sẽ dự đoán gần như tất cả dữ liệu đều thuộc về lớp đa số. Chính vì vậy chúng tôi đề xuất kiến trúc mô hình THANOS để xử lý vấn đề mất cân bằng dữ liệu.

### Mô hình kiến trúc THANOS



Hình ‑. Mô hình kiến trúc THANOS

Với kiến trúc THANOS, chúng tôi có thêm 1 pha sau pha biểu diễn dữ liệu để xử lý vấn đề mất cân bằng dữ liệu. Mô hình của chúng tôi là mô hình đa nhãn, vì vậy với mỗi một mô hình tương ứng với nhãn đó sẽ có sự phân bố lớp nhị phân khác nhau với những mô hình khác.

Chúng tôi coi rằng sự mất cân bằng dữ liệu xảy ra khi tỷ lệ số lượng mẫu dữ liệu có lớp đa số với số lượng mẫu dữ liệu có lớp thiểu số lớn hơn 3, tức là dữ liệu của lớp nhị phân phân bố với tỷ lệ 75/25 hoặc lớn hơn. Khi đó, chúng tôi sẽ sử dụng những kĩ thuật xử lý mất cân bằng dữ liệu để áp dụng với từng bộ phân lớp trong mô hình phân lớp đa nhãn của chúng tôi.

1. **Random Oversampling**

Trong quá trình over sampling, chúng ta tạo dữ liệu bổ sung cho lớp thiểu số bằng cách tạo ra các bản sao dữ liệu từ lớp thiểu số hoặc bằng một số phương pháp để tạo thêm dữ liệu đại diện cho lớp thiểu số dựa trên những dữ liệu đã tồn tại. Oversampling hoạt động lấy ngẫu nhiên một điểm từ lớp thiểu số và tính toán k-láng giềng gần nhất cho điểm này. Mặc dù vậy, lặp lại các dữ liệu từ lớp thiểu số có thể gây nên hiện tượng overfitting.

Phương pháp dễ dàng nhất là random over sampling, trong đó chúng ta sẽ sao chép ngẫu nhiên các dữ liệu trong lớp thiểu số cho đến khi cả hai lớp có cùng kích thước hoặc đáp ứng đến một tỷ lệ nào đó.

Các mẫu dữ liệu lấy từ tập huấn luyện được lựa chọn ngẫu nhiên và có sự thay thế. Điều đó nghĩa là các mẫu dữ liệu từ lớp thiểu số có thể được chọn và thêm vào dữ liệu huấn luyện mới - là dữ liệu huấn luyện đã cân bằng hơn. Chúng được chọn từ tập dữ liệu huấn luyện ban đầu, được thêm vào lớp thiểu số của tập dữ liệu mới, sau đó được trả lại hoặc thay thế trong dữ liệu ban đầu để chúng được chọn lại.

Kỹ thuật này có hiệu quả đối với những thuật toán học máy bị ảnh hưởng bởi phân phối lệch và trong đó nhiều mẫu dữ liệu trùng lặp cho một lớp nhất định có thể ảnh hưởng đến sự phù hợp của mô hình. Điều này có thể bao gồm các thuật toán học lặp đi lặp lại các hệ số, như các mạng nơ-ron nhân tạo sử dụng giảm độ dốc ngẫu nhiên. Nó cũng có thể ảnh hưởng đến các mô hình tìm kiếm sự phân tách dữ liệu tốt, chẳng hạn như SVM và Decision Trees.

Điều này hữu ích điều chỉnh phân phối lớp mục tiêu. Trong một vài trường hợp, việc tìm kiếm phân phối cân bằng cho tập dữ liệu mất cân bằng đáng kể có thể khiến các thuật toán overfit trong lớp thiểu số bị ảnh hưởng, dẫn đến tăng lỗi tổng quát hóa. Kết quả có thể là hiệu suất tốt hơn trên tập dữ liệu huấn luyện nhưng hiệu suất kém hơn trên tập dữ kiểm thử.

Do đó, theo dõi hiệu suất trên cả hai tập dữ liệu huấn luyện và kiểm thử sau khi lấy mẫu quá mức (oversampling) và so sánh các kết quả với cùng một thuật toán trên tập dữ liệu ban đầu là một cách tốt để có được cái nhìn sâu hơn về tác động của phương pháp.

Sự gia tăng số lượng các mẫu dữ liệu của lớp thiểu số, đặc biệt nếu sự chênh lệch lớp nghiêm trọng, cũng có thể dẫn đến sự tăng đáng kể trong chi phí tính toán khi lắp mô hình, nhất là khi thấy lại các mẫu dữ liệu lặp lại tương tự trong tập dữ liệu huấn luyện khi đang xem xét.

1. **Random Undersampling**

Lấy mẫu dưới (Undersampling) liên quan đến một nhóm các kỹ thuật được thiết kế để cân bằng phân phối các lớp cho tập dữ liệu phân loại có phân phối lớp lệch. Kỹ thuật lấy mẫu dưới loại bỏ các mẫu dữ liệu khỏi tập dữ liệu huấn luyện thuộc về lớp đa số để cân bằng việc phân phối lớp tốt hơn, ví dụ như giảm độ lệch từ 1:100 xuống 1:10, 1:2 hoặc thậm chí là lớp phân phối 1:1. Điều này khác với việc cố gắng lấy thêm các mẫu dữ liệu vào lớp thiểu số để giảm bớt sự sai lệch phân phối lớp của kỹ thuật lấy mẫu quá mức (oversampling). Phương pháp lấy mẫu dưới có thể được sử dụng trực tiếp trên tập dữ liệu huấn luyện mà sau đó có thể được sử dụng để phù hợp với mô hình học máy. Thông thường, các phương pháp lấy mẫu dưới được sử dụng cùng với kỹ thuật lấy mẫu quá mức cho lớp thiểu số và sự kết hợp này thường mang lại hiệu suất cao hơn so với việc chỉ sử dụng lấy mẫu quá mức hoặc lấy mẫu dưới trên tập dữ liệu huấn luyện.

Kỹ thuật lấy mẫu dưới đơn giản nhất liên quan đến việc chọn ngẫu nhiên các mẫu dữ liệu từ lớp đa số và xóa chúng khỏi tập dữ liệu huấn luyện. Điều này được gọi là lấy mẫu dưới ngẫu nhiên (random undersampling). Mặc dù đơn giản và hiệu quả, một hạn chế của kỹ thuật này là các mẫu dữ liệu bị loại bỏ mà không quan tâm đến mức độ hữu ích hoặc quan trọng của chúng trong việc xác định ranh giới quyết định giữa các lớp. Điều này có nghĩa là các thông tin hữu ích có thể, hoặc thậm chí có khả năng sẽ bị xóa. Việc mất dữ liệu như vậy có thể làm cho ranh giới quyết định giữa các trường hợp thiểu số và đa số khó để học hơn, dẫn đến giảm hiệu suất phân lớp

1. **SMOTE**

Một cách tiếp cận để giải quyết các tập dữ liệu không cân bằng là lấy mẫu quá mức đối với lớp thiểu số - cách tiếp cận đơn giản nhất liên quan đến việc sao chép các tập dữ liệu trong lớp thiểu số, mặc dù những tập dữ liệu này không thêm bất kỳ thông tin mới nào vào mô hình. Thay vào đó những tập dữ liệu mới có thể được tổng hợp từ tập dữ liệu có sẵn. Đây là một kiểu tăng dữ liệu cho lớp thiểu số và được gọi là kỹ thuật lấy mẫu quá mức thiểu số tổng hợp (Synthetic Minority Oversampling Technique), viết tắt là SMOTE.

Một cải tiến trong việc sao chép các tập dữ liệu trong lớp thiểu số là tổng hợp mẫu dữ liệu mới từ lớp thiểu số. Đó là một kiểu tăng dữ liệu cho dữ liệu dạng bảng và có thể rất hiệu quả.

SMOTE hoạt động bằng cách chọn các mẫu dữ liệu gần nhau trong không gian đối tượng, vẽ một đường thẳng giữa mẫu dữ liệu trong không gian đối tượng và vẽ một mẫu dữ liệu mới tại một điểm dọc theo đường đó.

Phương pháp này có thể được sử dụng để tạo ra nhiều mẫu dữ liệu cho lớp thiểu số theo yêu cầu. Trước tiên nó đề xuất sử dụng lấy mẫu dưới ngẫu nhiên (undersampling) để cắt bớt số lượng mẫu dữ liệu trong lớp đa số, sau đó sử dụng SMOTE để lấy mẫu quá mức (oversample) cho lớp thiểu số nhằm cân bằng phân bố lớp.

Nhược điểm của phương pháp này là các mẫu dữ liệu được tạo ra mà không xem xét dữ liệu lớp đa số, nó có thể tạo ra các mẫu dữ liệu không rõ ràng nếu có sự chồng chéo mạnh giữa các lớp.

1. **Bagging Classifier**

Bagging (Bootstrap Aggregration) là một thuật toán tổng hợp (Ensemble Algorithm) mà nó áp dụng với nhiều mô hình trên các tập con khác nhau của tập dữ liệu huấn luyện, sau đó kết hợp các dự đoán từ tất cả các mô hình. Rừng ngẫu nhiên (Random Forest) là một phần mở rộng của bagging - tức là cũng chọn ngẫu nhiên các tập hợp con của các đặc trưng (Features) được sử dụng trong mỗi mẫu dữ liệu. Cả Bagging và rừng ngẫu nhiên đã chứng minh được hiệu quả đối với một loạt các vấn đề mô hình dự đoán khác nhau. Mặc dù hiệu quả, chúng không phù hợp với các bài toán phân loại có phân bố các lớp lệch. Tuy nhiên, nhiều sửa đổi đối với các thuật toán đã được đề xuất để thích nghi với chúng và làm cho chúng phù hợp hơn với sự mất cân bằng lớp đáng kể.

Có nhiều cách để áp dụng Bagging vào xử lý mất cân bằng dữ liệu. Trong đó cách tiếp cận đơn giản nhất là lấy lại mẫu dữ liệu (Resampling) trên các mẫu dữ liệu con (Bootstrap) trước khi áp dụng lên các mô hình học yếu - là các mô hình giống hoặc khác nhau trên các tập dữ liệu bootstrap. Điều này có nghĩa là chúng ta có thể áp dụng lấy mẫu quá mức cho lớp thiểu số hoặc lấy mẫu dưới mức cho lớp đa số.

Lấy mẫu quá mức lớp thiểu số trong tập dữ liệu bootstrap được gọi là OverBagging; tương tự như vậy, việc lấy mẫu dưới lớp đa số trong bootstrap được gọi là UnderBagging và kết hợp cả hai cách tiếp cận được gọi là OverUnderBagging. Cụ thể, nó cung cấp một phiên bản Bagging sử dụng chiến lược lấy mẫu dưới ngẫu nhiên (Random Under Sampling) trên lớp đa số trong mẫu bootstrap để cân bằng hai lớp.

1. **Random Forest với trọng số class và Random Undersampling**

Rừng ngẫu nhiên (Random Forest) là một tập hợp của các mô hình cây quyết định và có thể được coi là một cải tiến so với kỹ thuật Bagging. Giống như Bagging, rừng ngẫu nhiên thực hiện việc chọn các mẫu bootstrap từ tập dữ liệu đào tạo và lắp một cây quyết định trên mỗi mẫu. Sự khác biệt chính là không phải tất cả các đặc trưng (các cột) được sử dụng; thay vào đó, một tập con nhỏ, được chọn ngẫu nhiên của các đặc trưng (cột) được chọn cho mỗi mẫu bootstrap. Điều này có tác dụng loại bỏ tương quan giữa các cây quyết định (làm cho chúng độc lập hơn), và do đó, cải thiện dự đoán tổng hợp.

Giống như kỹ thuật Bagging, rừng ngẫu nhiên rất hiệu quả trên bài toán hay tập dữ liệu lớn và trải rộng, nhưng hiệu suất của thuật toán nguyên gốc không lớn đối với các bài toán phân loại mất cân bằng.

Một kỹ thuật đơn giản để sửa đổi cây quyết định để xử lý mất cân bằng là thay đổi trọng số mà mỗi lớp có khi tính điểm “tạp chất” của một điểm phân nhánh. Tạp chất đo lường mức độ trộn lẫn của các nhóm mẫu dữ liệu đối với một phần đã cho trong tập dữ liệu huấn luyện và thường được đo bằng độ đo Gini hoặc entropy. Việc tính toán có thể bị sai lệch để một phần dự đoán cho tầng lớp thiểu số được ưu tiên, cho phép một số kết quả False Positive đối với tầng lớp đa số. Ta có thể thực hiện phương pháp này bằng cách thay đổi tham số class\_weight trên lớp RandomForestClassifier.

Nếu mỗi cây quyết định được xây dựng từ một mẫu bootstrap, phân phối lớp trong mẫu dữ liệu sẽ khác nhau đối với mỗi cây. Do đó, chúng ta có thể thay đổi trọng số lớp dựa trên phân phối lớp trong mỗi mẫu bootstrap, thay vì toàn bộ tập dữ liệu huấn luyện như trên. Ta có thể thực hiện bằng cách đặt tham số class\_weight thành giá trị ‘balanced\_subsample‘.

Một kỹ thuật hữu ích khác để xử lý mất cân bằng dữ liệu đối với rừng ngẫu nhiên là thực hiện lấy mẫu lại dữ liệu trên mẫu bootstrap để thay đổi phân bố lớp. Lớp BalancedRandomForestClassifier từ thư viện imbalanced-learn thực hiện điều này và thực hiện lấy mẫu dưới ngẫu nhiên (Random Oversampling) của lớp đa số trong mẫu bootstrap. Đây thường được gọi là Rừng Ngẫu nhiên Cân bằng (Balanced Random Forest).

1. **Easy Ensemble Classifier**

Khi thực hiện kỹ thuật Bagging để xử lý mất cân bằng dữ liệu, ta có thể sử dụng lấy mẫu lại ngẫu nhiên (Random Resampling) của lớp đa số để tạo ra nhiều bộ dữ liệu với phân bố lớp cân bằng. Cụ thể, một tập dữ liệu được tạo từ tất cả các mẫu dữ liệu trong lớp thiểu số và một mẫu nhỏ được chọn ngẫu nhiên từ lớp đa số để tạo nên sự phân bố lớp cân bằng. Sau đó, một mô hình học yếu (weaker learner) được sử dụng trên tập dữ liệu này. Quá trình này có thể được lặp lại nhiều lần và dự đoán trung bình trên toàn bộ mô hình có thể được sử dụng để đưa ra dự đoán cuối cùng. Việc xây dựng có chọn lọc các mẫu con được coi là một kiểu lấy mẫu dưới (Undersampling) của lớp đa số. Việc tạo ra nhiều mẫu con cho phép phương pháp tổng hợp (ensemble) khắc phục nhược điểm của việc lấy mẫu dưới trong đó thông tin có giá trị bị loại bỏ khỏi quá trình huấn luyện.

Easy Ensemble là phương pháp liên quan đến việc tạo ra các mẫu dữ liệu cân bằng của tập dữ liệu huấn luyện bằng cách chọn tất cả các mẫu dữ liệu từ lớp thiểu số và một tập hợp con từ lớp đa số, sau đó sử dụng cây quyết định cho mỗi tập dữ liệu. Thay vì sử dụng cây quyết định đã được lược bớt các đặc trưng, cây quyết định tăng cường được sử dụng trên mỗi tập dữ liệu con, cụ thể là thuật toán AdaBoost. AdaBoost hoạt động bằng cách trước tiên áp dụng cây quyết định trên tập dữ liệu, sau đó xác định các lỗi do cây tạo ra và tính toán trọng số các mẫu dữ liệu trong tập dữ liệu theo các lỗi đó, do đó các mẫu dữ liệu bị phân loại sai sẽ được để ý hơn và các mẫu dữ liệu được phân loại chính xác sẽ ít được quan tâm hơn. Sau đó, một cây tiếp theo được áp dụng với tập dữ liệu có trọng số nhằm sửa lỗi. Quá trình sau đó được lặp lại cho một số cây quyết định nhất định.

# Thực nghiệm và kết quả

## Cài đặt và cấu hình

### Triển khai mô hình

Mô hình của chúng tôi được cài đặt dựa trên nền tảng Python version 3.9 và thư viện Imblearn. Thư viện imblearn được thiết kế đặc biệt để giải quyết các bộ dữ liệu không cân bằng. Nó cung cấp các phương pháp khác nhau như lấy mẫu dưới, lấy mẫu quá mức và SMOTE để xử lý và loại bỏ sự mất cân bằng khỏi tập dữ liệu. Thư viện này bao gồm các phương pháp tổng hợp khác nhau như các bộ phân lớp bagging, rừng ngẫu nhiên và bộ phân lớp tăng cường có thể được sử dụng để đào tạo mô hình cho các tập dữ liệu không cân bằng với độ chính xác rất hiệu quả.

Một vài gói thư viện khác được sử dụng trong dự án:

* numpy
* pandas
* sklearn
* pickle
* matplotlib
* collections

### Môi trường thực nghiệm

Mô hình của chúng tôi chạy trên hệ điều hành Windows 10 với các thông số sau:

+ CPU: Intel(R) Core(TM) i5-8300H CPU @2.30GHz

+ RAM: 8GB

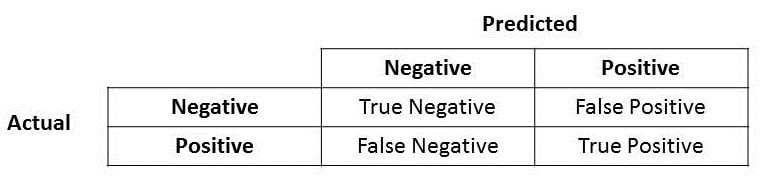
+ GPU: NVIDIA GTX 1050Ti

## Phương pháp đánh giá

Trong các bài toán phân lớp có hai lớp dữ liệu, người ta thường định nghĩa lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp Positive, lớp còn lại được gọi là Negative. Ta định nghĩa True Positive là số điểm Positive mô hình dự đoán đúng, False Positive là số điểm Positive mô hình dự đoán sai, True Negative là số điểm Negative mô hình dự đoán đúng và False Negative là số điểm Negative mô hình dự đoán sai.

1. **Precision và Recall**

Ta có hình minh họa về confusion matrix:



Hình ‑. Confusion matrix

Precision được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm mô hình dự đoán là Positive. Recall được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng số điểm thật sự là Positive (hay tổng số điểm được gán nhãn là Positive ban đầu).

Công thức tính Precision:

Công thức tính Recall:

Precision càng cao, tức là số điểm mô hình dự đoán là Positive đều là Positive càng nhiều. Precision = 1, tức là tất cả số điểm mô hình dự đoán là Positive đều đúng, hay không có điểm nào có nhãn là Negative mà mô hình dự đoán nhầm là Positive.

Recall càng cao, tức là số điểm là Positive bị bỏ sót càng ít. Recall = 1, tức là tất cả số điểm có nhãn là Positive đều được mô hình nhận ra.

Tuy nhiên, chỉ có Precision hay chỉ có Recall thì không đánh giá được chất lượng mô hình.

Chỉ dùng Precision, mô hình chỉ đưa ra dự đoán cho một điểm mà nó chắc chắn nhất. Khi đó Precision = 1, tuy nhiên ta không thể nói mô hình này tốt.

Chỉ dùng Recall, nếu mô hình dự đoán tất cả các điểm đều là positive. Khi đó, Recall = 1, tuy nhiên ta cũng không thể nói đây là mô hình tốt.

1. **F1-score**

Đối với các ứng dụng mà độ chính xác precision và độ hồi tưởng recall có tầm quan trọng tương tự nhau, thường thuận tiện khi kết hợp chúng thành một đại lượng duy nhất được gọi là độ đo F1. Độ đo F1 được định nghĩa là trung bình điều hoà (harmonic mean) của độ chính xác precision và độ hồi tưởng recall (giả sử hai đại lượng này khác 0). Việc diễn giải độ đo F1 một cách trực quan sẽ khó hơn một chút so với độ chính xác precision và độ hồi tưởng recall một cách riêng lẻ, nhưng có thể kỳ vọng tóm tắt hai đại lượng thành một chỉ số dễ so sánh.

Độ đo F1 được tính theo công thức

Để có được độ đo F1 cao, một mô hình cần có độ chính xác precision và độ hồi tưởng recall cao. Điều này là do độ đo F1 bị kéo xuống khá nhiều khi lấy trung bình điều hoà nếu một trong những độ chính xác precision hoặc độ hồi tưởng recall thấp.

Với các bài toán phân lớp nhiều lớp, ta lần lượt xem một lớp là positive, các lớp còn lại là negative. Khi đó ta có hai cách tính F1-score: macro F1-score và micro F1-với score.

1. **Marco F1-score**

Marco F1-score được tính bằng công thức

với *macro-precision* và *macro-recall* là trung bình cộng của precision và recall theo các lớp.

1. **Mirco F1-score**

Mirco F1-score được tính bằng công thức

với *micro-precision* là tỉ lệ tổng số điểm của toàn bộ các điểm positive thuộc các lớp trên tổng số điểm mô hình dự đoán là positive thuộc các lớp; *micro-recall* được tính bằng cách tương tự.

## Kết quả thực nghiệm

Đầu tiên, chúng tôi sử dụng cách biểu diễn one-hot thì thu được kết quả như bảng dưới đây:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Độ đo | Các khía cạnh (%) | | | | | | | Macro (%) | Micro (%) |
| Ship | Giá | Chính hãng | Chất lượng | Dịch vụ | An toàn | Other |
| *Naïve Bayes* | P | 87.21 | 65.66 | 50 | 66.67 | 60.56 | 68.42 | 45.95 | 63.50 | 73.40 |
| R | 96.22 | 63.11 | 41.25 | 63.68 | 48.86 | 56.52 | 32.08 | 57.39 | 70.85 |
| F1 | 91.50 | 64.36 | 45.21 | 65.14 | 54.09 | 61.90 | 37.78 | 60.00 | 72.10 |
| *SVM* | P | 95.26 | 00.00 | 00.00 | 00.00 | 00.00 | 00.00 | 20.98 | 16.61 | 67.72 |
| R | 91.18 | 00.00 | 00.00 | 00.00 | 00.00 | 00.00 | 88.68 | 25.69 | 39.48 |
| F1 | 93.18 | 00.00 | 00.00 | 00.00 | 00.00 | 00.00 | 33.94 | 18.16 | 49.88 |
| *Random Forest* | P | 96.46 | 94.38 | 84.31 | 81.56 | 92.31 | 88.73 | 31.58 | 81.33 | 84.22 |
| R | 95.97 | 81.55 | 53.75 | 51.57 | 40.91 | 68.48 | 67.92 | 65.74 | 73.17 |
| F1 | 96.21 | 87.50 | 65.65 | 63.19 | 56.69 | 77.30 | 43.11 | 69.95 | 78.31 |
| *Logistic Regression* | P | 96.98 | 93.75 | 73.13 | 79.70 | 95.56 | 86.52 | 37.08 | 80.39 | 85.08 |
| R | 96.98 | 87.38 | 61.25 | 72.20 | 48.86 | 83.70 | 62.26 | 73.23 | 80.89 |
| F1 | 96.98 | 90.45 | 66.67 | 75.76 | 64.66 | 85.08 | 46.48 | 75.15 | 82.93 |

Bảng ‑. Kết quả mô hình với cách biểu diễn One-hot trên dữ liệu mẹ & bé shopee

Từ những số liệu thống kê ở trên, có thể dễ dàng nhận thấy kết quả của mô hình sử dụng cách biểu diễn One-hot không cao do trọng số của One-hot chưa thể hiện được mức độ quan trọng của các từ đối với từng nhãn. Dữ liệu mất cân bằng cũng làm cho hiệu suất của mô hình bị giảm, đặc biệt là SVM vì đối với một số nhãn bị mất cân bằng mạnh mô hình SVM sẽ dự đoán ra toàn 0 do mô hình SVM có xu hướng dự đoán thiên về lớp đa số.

Tiếp theo, chúng tôi áp dụng cách biểu diễn Chi2 thay cho One-hot và được kết quả như bảng:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Độ đo | Các khía cạnh (%) | | | | | | | Macro (%) | Micro (%) |
| Ship | Giá | Chính hãng | Chất lượng | Dịch vụ | An toàn | Other |
| *Naïve Bayes* | P | 96.38 | 93.14 | 72.15 | 74.36 | 83.93 | 87.10 | 35.11 | 77.45 | 82.60 |
| R | 93.95 | 92.23 | 71.25 | 65.02 | 53.41 | 88.04 | 62.26 | 75.17 | 80.21 |
| F1 | 95.15 | 92.68 | 71.70 | 69.38 | 65.28 | 87.57 | 44.90 | 75.24 | 81.39 |
| *SVM* | P | 96.71 | 93.14 | 73.08 | 83.66 | 90.38 | 86.96 | 31.43 | 79.34 | 84.14 |
| R | 96.22 | 92.23 | 71.25 | 57.40 | 53.41 | 86.96 | 62.26 | 74.25 | 79.34 |
| F1 | 96.46 | 92.68 | 72.15 | 68.09 | 67.14 | 86.96 | 41.77 | 75.04 | 81.67 |
| *Random Forest* | P | 95.99 | 95.29 | 80.70 | 79.76 | 92.31 | 94.81 | 30.93 | 81.40 | 84.92 |
| R | 96.47 | 78.64 | 57.50 | 60.09 | 40.91 | 79.35 | 56.60 | 67.08 | 75.58 |
| F1 | 96.23 | 86.17 | 67.15 | 68.54 | 56.69 | 86.39 | 40.00 | 71.60 | 79.98 |
| *Logistic Regression* | P | 96.26 | 92.08 | 70.77 | 78.22 | 87.27 | 89.29 | 40.28 | 79.17 | 85.20 |
| R | 97.23 | 90.29 | 57.50 | 70.85 | 54.55 | 81.52 | 54.72 | 72.38 | 80.60 |
| F1 | 96.74 | 91.18 | 63.45 | 74.35 | 67.13 | 85.23 | 46.40 | 74.93 | 82.84 |

Bảng ‑. Kết quả mô hình với cách biểu diễn Chi2 trên dữ liệu mẹ & bé shopee

Dựa vào kết quả trên, sau khi áp dụng biểu diễn Chi2 đã cho kết quả tăng cao rõ rệt đối với tất cả các khía cạnh so với One-hot (trung bình tăng 26% với mô hình Naïve Bayes, một số khía cạnh tăng 5% vF một khía cạnh giảm 1% với mô hình Random Forest, mô hình Logistic Regression không có sự thay đổi nhiều) bởi vì trọng số của cách biểu diễn. Chi2 đã thể hiện được mức độ quan trọng của từ đối với từng nhãn.

Chúng tôi nhận thấy rằng cách biểu diễn với trọng số của Chi2 đã làm hiệu suất phân lớp của mô hình tăng cao hơn so với cách biểu diễn One-hot. Vì vậy, chúng tôi sẽ sử dụng biểu diễn Chi2 cho các mô hình về sau.

Áp dụng cách biểu diễn Chi2 kết hợp với lọc các từ vựng có độ quan trọng cao hơn thông qua trọng số của Chi2 và resampling các nhãn có độ lệch giữa hai lớp lớn, chúng tôi thu được kết quả như bảng:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Độ đo | Các khía cạnh (%) | | | | | | | Macro (%) | Micro (%) |
| Ship | Giá | Chính hãng | Chất lượng | Dịch vụ | An toàn | Other |
| *Naïve Bayes (Random Over sampling = 0.6)* | P | 96.38 | 93.14 | 72.15 | 74.36 | 85.71 | 87.10 | 34.74 | 77.65 | 82.62 |
| R | 93.95 | 92.23 | 71.25 | 65.02 | 54.55 | 88.04 | 62.26 | 75.33 | 80.31 |
| F1 | 95.15 | 92.68 | 71.70 | 69.38 | 66.67 | 87.57 | 44.59 | 75.39 | 81.45 |
| *SVM (Random Over sampling = 0.8 + Score > 50)* | P | 96.74 | 92.45 | 60.68 | 83.66 | 90.38 | 87.10 | 31.96 | 77.57 | 82.79 |
| R | 97.23 | 95.15 | 88.75 | 57.40 | 53.41 | 88.04 | 58.49 | 76.92 | 81.27 |
| F1 | 96.98 | 93.78 | 72.08 | 68.09 | 67.14 | 87.57 | 41.33 | 75.28 | 82.03 |
| *SVM (Random Under sampling = 0.4 + Score > 50)* | P | 96.74 | 93.14 | 68.04 | 83.66 | 90.00 | 86.96 | 32.35 | 78.70 | 83.72 |
| R | 97.23 | 92.23 | 82.50 | 57.40 | 51.14 | 86.96 | 62.26 | 75.67 | 80.41 |
| F1 | 96.98 | 92.68 | 74.58 | 68.09 | 65.22 | 86.96 | 42.58 | 75.30 | 82.03 |
| *Random Forest (Random Over sampling = 0.7 + Score > 15)* | P | 95.79 | 88.07 | 62.50 | 84.18 | 85.96 | 87.37 | 37.65 | 77.36 | 83.35 |
| R | 97.48 | 93.20 | 87.50 | 66.82 | 55.68 | 90.22 | 60.38 | 78.75 | 83.59 |
| F1 | 96.63 | 90.57 | 72.92 | 74.50 | 67.59 | 88.77 | 46.38 | 76.76 | 83.47 |
| *Random Forest (Random Under sampling = 0.5 + Score > 15)* | P | 95.79 | 89.81 | 62.61 | 84.18 | 86.44 | 87.37 | 38.10 | 77.76 | 83.59 |
| R | 97.48 | 94.17 | 90.00 | 66.82 | 57.95 | 90.22 | 60.38 | 79.57 | 84.07 |
| F1 | 96.63 | 91.94 | 73.85 | 74.50 | 69.39 | 88.77 | 46.72 | 77.40 | 83.83 |
| *Logistic Regression (Random Over sampling = 0.6)* | P | 96.27 | 91.26 | 66.67 | 78.22 | 71.62 | 82.83 | 41.43 | 75.47 | 82.89 |
| R | 97.48 | 91.26 | 80.00 | 70.85 | 60.23 | 89.13 | 54.72 | 77.67 | 83.69 |
| F1 | 96.87 | 91.26 | 72.73 | 74.35 | 65.43 | 85.86 | 47.15 | 76.24 | 83.29 |

Bảng ‑. Kết quả mô hình với cách biểu diễn Chi2 đã lọc từ kết hợp Resampling trên dữ liệu mẹ & bé shopee

Từ kết quả thu được thống kê ở bảng trên, bằng cách resampling các nhãn có sự mất cân bằng đáng kể giữa hai lớp, kết quả phân lớp của các mô hình đã được cải thiện hơn so với các mô hình còn lại.

Cuối cùng, chúng tôi kết hợp biểu diễn Chi2 cùng với kỹ thuật Bagging:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Độ đo | Các khía cạnh (%) | | | | | | | Macro (%) | Micro (%) |
| Ship | Giá | Chính hãng | Chất lượng | Dịch vụ | An toàn | Other |
| *Balance Random Forest (Random Over sampling = 0.7 + Score > 35)* | P | 96.46 | 93.33 | 61.54 | 73.14 | 89.09 | 86.96 | 32.00 | 76.07 | 81.50 |
| R | 95.97 | 95.15 | 90.00 | 79.37 | 55.68 | 86.96 | 45.28 | 78.34 | 85.04 |
| F1 | 96.21 | 94.23 | 73.10 | 76.13 | 68.53 | 86.96 | 37.50 | 76.09 | 83.23 |
| *Balance Random Forest (Random Under sampling = 0.5 + Score > 35)* | P | 96.46 | 93.27 | 61.02 | 73.14 | 88.89 | 87.10 | 33.33 | 76.17 | 81.50 |
| R | 95.97 | 94.17 | 90.00 | 79.37 | 54.55 | 88.04 | 47.17 | 78.47 | 85.04 |
| F1 | 96.21 | 93.72 | 72.73 | 76.13 | 67.61 | 87.57 | 39.06 | 76.15 | 83.23 |
| *Bagging (Random Over sampling = 0.6 + Score > 50)* | P | 96.94 | 92.45 | 61.21 | 77.89 | 90.20 | 87.10 | 30.61 | 76.63 | 81.64 |
| R | 95.72 | 95.15 | 88.75 | 66.37 | 52.27 | 88.04 | 56.60 | 77.56 | 82.43 |
| F1 | 96.32 | 93.78 | 72.45 | 71.67 | 66.19 | 87.57 | 39.74 | 75.39 | 82.04 |
| *Bagging (Random Under sampling = 0.2 + Score > 50)* | P | 96.94 | 92.45 | 60.68 | 77.89 | 90.20 | 87.10 | 30.93 | 76.60 | 81.64 |
| R | 95.72 | 95.15 | 88.75 | 66.37 | 52.27 | 88.04 | 56.60 | 77.56 | 82.43 |
| F1 | 96.32 | 93.78 | 72.08 | 71.67 | 66.19 | 87.57 | 40.00 | 75.37 | 82.04 |

Bảng ‑. Kết quả với các mô hình kết hợp kỹ thuật Bagging trên dữ liệu mẹ & bé shopee

Theo như bảng trên, kết quả thu được bằng cách biểu diễn Chi2 và kỹ thuật Bagging có sự chênh lệch nhỏ nhưng không đáng kể.

Kết luận, với tập dữ liệu Mẹ & Bé Shopee, các kỹ thuật xử lý mất cân bằng dữ liệu đã làm tăng đáng kể hiệu suất của các mô hình phân lớp, nhất là mô hình Random Forest đạt kết quả cao nhất.

Với ba tập dữ liệu còn lại, chúng tôi sẽ so sánh kết quả giữa cách biểu diễn one-hot và cách biểu diễn chi2 kết hợp với lọc từ vựng và xử lý mất cân bằng dữ liệu.

Dưới đây là bảng kết quả trên tập dữ liệu mẹ & bé tiki với cách biểu diễn one-hot:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Độ đo | Các khía cạnh (%) | | | | | | | Macro (%) | Micro (%) |
| Ship | Giá | Chính hãng | Chất lượng | Dịch vụ | An toàn | Other |
| *Naïve Bayes* | P | 86.40 | 68.81 | 16.67 | 79.06 | 66.34 | 38.64 | 25.00 | 59.31 | 75.15 |
| R | 91.13 | 58.14 | 6.06 | 76.04 | 65.69 | 30.36 | 24.00 | 54.57 | 70.53 |
| F1 | 88.70 | 63.03 | 8.89 | 77.52 | 66.01 | 34.00 | 24.49 | 56.35 | 72.77 |
| *SVM* | P | 97.52 | 0.00 | 0.00 | 88.41 | 0.00 | 0.00 | 14.19 | 30.99 | 92.92 |
| R | 66.24 | 0.00 | 0.00 | 50.35 | 0.00 | 0.00 | 82.00 | 19.43 | 35.74 |
| F1 | 78.89 | 00.00 | 00.00 | 64.16 | 00.00 | 00.00 | 24.19 | 23.84 | 51.62 |
| *Random Forest* | P | 94.50 | 94.32 | 0.00 | 81.85 | 88.24 | 86.67 | 24.84 | 74.26 | 88.66 |
| R | 86.92 | 64.34 | 0.00 | 70.49 | 44.12 | 46.43 | 76.00 | 52.05 | 66.63 |
| F1 | 90.55 | 76.50 | 0.00 | 75.75 | 58.82 | 60.47 | 37.44 | 60.35 | 76.08 |
| *Logistic Regression* | P | 93.86 | 91.23 | 75.00 | 81.00 | 87.88 | 80.00 | 28.71 | 84.83 | 86.96 |
| R | 90.30 | 80.62 | 27.27 | 78.47 | 56.86 | 64.29 | 58.00 | 66.30 | 76.57 |
| F1 | 92.04 | 85.60 | 40.00 | 79.72 | 69.05 | 71.29 | 38.41 | 72.95 | 81.43 |

Bảng ‑. Kết quả mô hình với cách biểu diễn One-hot trên dữ liệu mẹ & bé tiki

Bảng kết quả với cách biểu diễn chi2 kết hợp với lọc từ vựng và sample dữ liệu:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mô hình | Độ đo | Các khía cạnh (%) | | | | | | | Macro (%) | Micro (%) |
| Ship | Giá | Chính hãng | Chất lượng | Dịch vụ | An toàn | Other |
| *Naïve Bayes (Random Over sampling = 0.6)* | P | 91.07 | 80.71 | 70.59 | 51.04 | 34.66 | 89.74 | 00.00 | 69.64 | 60.07 |
| R | 86.08 | 87.60 | 36.36 | 94.10 | 85.29 | 62.50 | 00.00 | 75.32 | 85.44 |
| F1 | 88.50 | 84.01 | 48.00 | 66.18 | 49.29 | 73.68 | 00.00 | 68.28 | 70.54 |
| *SVM (Random Over sampling = 0.8 + Score > 50)* | P | 89.96 | 76.97 | 32.05 | 50.75 | 70.69 | 68.25 | 00.00 | 64.78 | 63.35 |
| R | 90.72 | 90.70 | 75.76 | 94.10 | 40.20 | 76.79 | 00.00 | 78.04 | 84.26 |
| F1 | 90.34 | 83.27 | 45.05 | 65.94 | 51.25 | 72.27 | 00.00 | 68.02 | 72.32 |
| *SVM (Random Under sampling = 0.4 + Score > 50)* | P | 89.96 | 78.00 | 62.50 | 50.75 | 83.67 | 73.33 | 00.00 | 78.70 | 66.57 |
| R | 90.72 | 90.70 | 45.45 | 94.10 | 40.20 | 78.57 | 00.00 | 75.67 | 83.20 |
| F1 | 90.34 | 83.87 | 52.63 | 65.94 | 54.30 | 75.86 | 00.00 | 75.30 | 73.96 |
| *Random Forest (Random Over sampling = 0.7 + Score > 15)* | P | 95.79 | 88.07 | 62.50 | 84.18 | 85.96 | 87.37 | 37.65 | 77.36 | 83.35 |
| R | 97.48 | 93.20 | 87.50 | 66.82 | 55.68 | 90.22 | 60.38 | 78.75 | 83.59 |
| F1 | 96.63 | 90.57 | 72.92 | 74.50 | 67.59 | 88.77 | 46.38 | 76.76 | 83.47 |
| *Random Forest (Random Under sampling = 0.5 + Score > 15)* | P | 95.79 | 89.81 | 62.61 | 84.18 | 86.44 | 87.37 | 38.10 | 77.76 | 83.59 |
| R | 97.48 | 94.17 | 90.00 | 66.82 | 57.95 | 90.22 | 60.38 | 79.57 | 84.07 |
| F1 | 96.63 | 91.94 | 0,7385 | 74.50 | 69.39 | 88.77 | 46.72 | 77.40 | 83.83 |
| *Logistic Regression (Random Over sampling = 0.6)* | P | 96.27 | 91.26 | 66.67 | 78.22 | 71.62 | 82.83 | 41.43 | 75.47 | 82.89 |
| R | 97.48 | 91.26 | 80.00 | 70.85 | 60.23 | 89.13 | 54.72 | 77.67 | 83.69 |
| F1 | 96.87 | 91.26 | 72.73 | 74.35 | 65.43 | 85.86 | 47.15 | 76.24 | 83.29 |

Bảng ‑. Kết quả mô hình với cách biểu diễn Chi2 đã lọc từ kết hợp Resampling trên dữ liệu mẹ & bé tiki

Dưới đây là bảng kết quả trên tập dữ liệu Công nghệ Shopee với cách biểu diễn one-hot:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro (%) | | 68.64 | 65.84 | 67.21 | 92.11 | 35.23 | 50.96 | 88.14 | 71.87 | 79.18 | 86.45 | 75.87 | 80.81 |
| Macro (%) | | 64.36 | 63.67 | 63.44 | 57.54 | 27.27 | 36.96 | 87.62 | 67.21 | 75.13 | 85.62 | 73.47 | 78.96 |
| Các khía cạnh (%) | Other | 22.05 | 48.28 | 30.27 | 14.29 | 00.00 | 25.00 | 30.54 | 87.93 | 45.33 | 35.17 | 87.93 | 50.25 |
| Phụ kiện | 36.99 | 55.10 | 44.26 | 00.00 | 00.00 | 00.00 | 88.46 | 46.94 | 61.33 | 82.86 | 59.18 | 69.05 |
| Dịch vụ | 79.57 | 73.63 | 76.49 | 97.00 | 48.26 | 64.45 | 88.51 | 76.62 | 82.13 | 87.36 | 75.62 | 81.07 |
| Chính hãng | 52.22 | 63.51 | 57.32 | 00.00 | 00.00 | 00.00 | 88.71 | 74.32 | 80.88 | 87.30 | 74.32 | 80.29 |
| Giá | 56.16 | 48.24 | 51.90 | 00.00 | 00.00 | 00.00 | 81.58 | 36.47 | 50.41 | 80.00 | 65.88 | 72.26 |
| Ship | 73.79 | 83.52 | 78.35 | 95.70 | 48.90 | 64.73 | 92.64 | 82.97 | 87.54 | 90.12 | 85.16 | 87.57 |
| Hiệu năng | 70.40 | 63.31 | 66.67 | 92.54 | 44.60 | 60.19 | 87.85 | 67.63 | 76.42 | 83.90 | 71.22 | 77.04 |
| Mẫu mã | 71.86 | 54.05 | 61.70 | 80.00 | 32.43 | 46.15 | 85.86 | 73.87 | 79.42 | 85.20 | 75.23 | 79.90 |
| Cấu hình | 73.91 | 68.00 | 70.83 | 95.06 | 44.00 | 60.16 | 87.34 | 78.86 | 82.88 | 88.20 | 81.14 | 84.52 |
| Độ đo | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Mô hình | | *Naïve Bayes* | | | *SVM* | | | *Random Forest* | | | *Logistic Regress-ion* | | |

Bảng ‑. Kết quả mô hình với cách biểu diễn One-hot trên dữ liệu công nghệ shopee

Bảng kết quả với cách biểu diễn chi2 kết hợp với lọc từ vựng và sample dữ liệu:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro (%) | | 77.28 | 75.16 | 76.20 | 76.76 | | 71.52 | 74.05 | 75.39 | 64.15 | 69.32 |
| Macro (%) | | 75.90 | 77.59 | 75.90 | 76.31 | | 73.21 | 72.31 | 75.99 | 68.77 | 68.24 |
| Các khía cạnh (%) | Other | 38.95 | 63.79 | 48.37 | 31.15 | | 65.52 | 42.22 | 28.79 | 65.52 | 40.00 |
| Phụ kiện | 58.33 | 85.71 | 69.42 | 61.54 | | 81.63 | 70.18 | 50.57 | 89.80 | 64.71 |
| Dịch vụ | 86.39 | 63.18 | 72.99 | 75.45 | | 82.59 | 78.86 | 84.54 | 40.80 | 55.03 |
| Chính hãng | 83.10 | 79.73 | 81.38 | 77.63 | | 79.73 | 78.67 | 78.67 | 79.73 | 79.19 |
| Giá | 64.86 | 84.71 | 73.47 | 64.80 | | 95.29 | 77.14 | 61.54 | 94.12 | 74.42 |
| Ship | 74.74 | 79.67 | 77.13 | 82.98 | | 85.71 | 84.32 | 81.05 | 84.62 | 82.80 |
| Hiệu năng | 74.34 | 81.29 | 77.66 | 82.81 | | 38.13 | 52.22 | 90.00 | 38.85 | 54.27 |
| Mẫu mã | 84.70 | 69.82 | 76.54 | 77.48 | | 77.48 | 77.48 | 74.56 | 76.58 | 75.56 |
| Cấu hình | 80.72 | 76.57 | 78.59 | 87.78 | | 45.14 | 59.62 | 86.96 | 45.71 | 59.93 |
| Độ đo | | P | R | F1 | P | | R | F1 | P | R | F1 |
| Mô hình | | *Naïve Bayes (Random Over sampling = 0.6 + Score > 15)* | | | | *SVM (Random Over sampling = 0.8)* | | | *SVM (Random Under sampling = 0.4)* | | |

**Bảng 5‑8. Kết quả mô hình với cách biểu diễn Chi2 đã lọc từ kết hợp Resampling trên dữ liệu công nghệ shopee**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro(%) | | 88.14 | 71.87 | 79.18 | 82.11 | 82.25 | 82.18 | 81.88 | 81.81 | 81.85 |
| Macro (%) | | 87.62 | 67.21 | 75.13 | 79.28 | 82.60 | 80.33 | 79.45 | 82.83 | 80.58 |
| Các khía cạnh (%) | Other | 36.52 | 72.41 | 48.55 | 38.05 | 74.14 | 50.29 | 36.70 | 68.97 | 47.90 |
| Phụ kiện | 56.86 | 59.18 | 58.00 | 54.55 | 85.71 | 66.67 | 65.08 | 83.67 | 73.21 |
| Dịch vụ | 88.19 | 81.09 | 84.46 | 88.11 | 81.09 | 84.46 | 88.11 | 81.09 | 84.46 |
| Chính hãng | 77.78 | 75.68 | 76.71 | 80.28 | 77.03 | 78.62 | 77.92 | 81.08 | 79.47 |
| Giá | 60.36 | 78.82 | 68.36 | 68.87 | 85.88 | 76.44 | 63.41 | 91.76 | 75.00 |
| Ship | 90.85 | 87.36 | 89.08 | 90.96 | 88.46 | 89.69 | 89.94 | 88.46 | 89.20 |
| Hiệu năng | 80.00 | 80.57 | 80.28 | 81.75 | 80.58 | 81.16 | 78.62 | 82.01 | 80.28 |
| Mẫu mã | 84.77 | 75.23 | 79.71 | 84.77 | 75.23 | 79.71 | 84.90 | 73.42 | 78.74 |
| Cấu hình | 84.91 | 86.85 | 85.86 | 84.92 | 86.86 | 85.88 | 87.65 | 81.14 | 84.27 |
| Độ đo | | P | R | F1 | P | R | F1 | P | R | F1 |
| Mô hình | | *Random Forest (Random Over sampling = 0.7 + Score > 15)* | | | *Random Forest (Random Under sampling = 0.5 + Score > 15)* | | | *Logistic Regression (Random Over sampling = 0.7 + Score > 15)* | | |

**Bảng 5‑9. Kết quả mô hình với cách biểu diễn Chi2 đã lọc từ kết hợp Resampling trên dữ liệu công nghệ shopee (tiếp)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro(%) | | 79.54 | 85.18 | 82.26 | 79.09 | 85.27 | 82.07 | 79.72 | 80.57 | 80.14 |
| Macro (%) | | 76.30 | 83.58 | 79.55 | 76.45 | 85.43 | 80.23 | 77.48 | 82.51 | 78.69 |
| Các khía cạnh (%) | Other | 46.39 | 77.59 | 58.06 | 41.30 | 65.52 | 50.67 | 34.82 | 67.24 | 45.88 |
| Phụ kiện | 53.62 | 75.51 | 62.71 | 56.94 | 83.67 | 67.77 | 49.43 | 87.76 | 63.24 |
| Dịch vụ | 85.50 | 85.07 | 85.29 | 83.89 | 88.06 | 85.92 | 87.73 | 71.14 | 78.57 |
| Chính hãng | 78.21 | 82.43 | 80.26 | 77.92 | 81.08 | 79.47 | 83.10 | 79.73 | 81.38 |
| Giá | 63.11 | 76.47 | 69.15 | 62.90 | 91.76 | 74.64 | 61.54 | 94.12 | 74.42 |
| Ship | 89.30 | 91.76 | 90.51 | 88.65 | 90.11 | 89.37 | 85.00 | 84.07 | 84.53 |
| Hiệu năng | 77.70 | 82.73 | 80.14 | 76.32 | 83.45 | 79.73 | 80.85 | 82.01 | 81.43 |
| Mẫu mã | 82.27 | 81.53 | 81.90 | 82.04 | 76.13 | 78.97 | 86.96 | 72.07 | 78.82 |
| Cấu hình | 80.69 | 93.14 | 86.47 | 82.98 | 89.14 | 85.95 | 85.25 | 89.14 | 87.15 |
| Độ đo | | P | R | F1 | P | R | F1 | P | R | F1 |
| Mô hình | | *Bagging(base= Decision Tree) (Over Sampling = 0.6 + Score > 1)* | | | *Bagging(base =Logistic Regression) (Random Over sampling = 0.7 + Score > 15)* | | | *Bagging(base =SVM) (Random Over sampling = 0.1 + Score > 1)* | | |

Bảng ‑. Kết quả với các mô hình kết hợp kỹ thuật Bagging trên dữ liệu công nghệ shopee

Dưới đây là bảng kết quả trên tập dữ liệu Công nghệ Tiki với cách biểu diễn one-hot:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro (%) | | 62.37 | 63.01 | 62.69 | 90.38 | 14.45 | 24.91 | 81.23 | 59.43 | 68.64 | 78.54 | 64.14 | 70.61 |
| Macro (%) | | 58.21  58.19 | 58.20  57.555555 | 57.50 | 11.30 | 08.08 | 09.43 | 78.29 | 51.21 | 58.80 | 74.85 | 58.51 | 64.78 |
| Các khía cạnh (%) | Other | 41.76 | 52.21 | 46.41 | 26.22 | 94.85 | 41.08 | 44.35 | 75.00 | 55.74 | 45.24 | 69.85 | 68.18 |
| Phụ kiện | 40.91 | 43.55 | 42.19 | 00.00 | 00.00 | 00.00 | 72.41 | 33.87 | 46.15 | 64.29 | 43.55 | 64.07 |
| Dịch vụ | 50.00 | 27.78 | 35.71 | 00.00 | 00.00 | 00.00 | 66.67 | 05.56 | 10.26 | 64.29 | 25.00 | 54.91 |
| Chính hãng | 67.88 | 74.40 | 70.99 | 00.00 | 00.00 | 00.00 | 77.68 | 69.60 | 73.42 | 82.14 | 73.60 | 51.92 |
| Giá | 64.49 | 51.11 | 57.02 | 00.00 | 00.00 | 00.00 | 83.33 | 55.56 | 66.67 | 78.18 | 63.70 | 36.00 |
| Ship | 74.41 | 86.70 | 80.08 | 90.38 | 64.68 | 75.40 | 86.09 | 90.83 | 88.39 | 85.92 | 83.94 | 77.64 |
| Hiệu năng | 48.57 | 55.43 | 51.78 | 00.00 | 00.00 | 00.00 | 78.05 | 34.78 | 48.12 | 70.00 | 53.26 | 70.20 |
| Mẫu mã | 72.61 | 54.95 | 58.01 | 00.00 | 00.00 | 00.00 | 75.54 | 60.34 | 67.09 | 76.06 | 62.07 | 84.92 |
| Cấu hình | 79.19 | 66.57 | 69.69 | 00.00 | 00.00 | 00.00 | 86.57 | 59.18 | 70.30 | 79.27 | 66.33 | 60.49 |
| Độ đo | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Mô hình | | *Naïve Bayes* | | | *SVM* | | | *Random Forest* | | | *Logistic Regression* | | |

**Bảng 5‑11. Kết quả mô hình với cách biểu diễn One-hot trên dữ liệu công nghệ tiki**

Bảng kết quả với cách biểu diễn chi2 kết hợp với lọc từ vựng và sample dữ liệu:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro (%) | | 71.16 | 76.33 | 73.65 | 59.16 | | 84.73 | 69.67 | 63.49 | 81.97 | 71.56 |
| Macro (%) | | 68.28 | 74.61 | 70.70 | 57.73 | | 83.60 | 67.21 | 60.20 | 79.76 | 67.87 |
| Các khía cạnh (%) | Other | 51.2 | 47.06 | 49.04 | 41.82 | | 33.82 | 37.40 | 45.16 | 41.18 | 43.08 |
| Phụ kiện | 48.75 | 62.90 | 54.93 | 48.04 | | 79.03 | 59.76 | 47.52 | 77.42 | 58.90 |
| Dịch vụ | 52.75 | 66.67 | 58.90 | 27.04 | | 73.61 | 39.55 | 36.17 | 47.22 | 40.96 |
| Chính hãng | 78.68 | 85.60 | 81.99 | 67.25 | | 92.00 | 77.70 | 68.67 | 91.20 | 78.35 |
| Giá | 69.88 | 85.93 | 77.08 | 62.96 | | 88.15 | 73.46 | 67.04 | 88.89 | 76.43 |
| Ship | 79.67 | 88.07 | 83.66 | 85.65 | | 87.61 | 86.62 | 85.65 | 87.61 | 86.62 |
| Hiệu năng | 63.04 | 63.04 | 6304 | 49.65 | | 76.09 | 60.09 | 53.38 | 77.17 | 63.11 |
| Mẫu mã | 81.82 | 56.90 | 6712 | 70.71 | | 80.46 | 75.27 | 74.71 | 74.71 | 74.71 |
| Cấu hình | 71.67 | 87.76 | 789 | 50.56 | | 91.84 | 65.22 | 48.42 | 93.88 | 63.89 |
| Độ đo | | P | R | F1 | P | | R | F1 | P | R | F1 |
| Mô hình | | *Naïve Bayes (Random over sampling = 0.6)* | | | | *SVM (Random over sampling = 0.8)* | | | *SVM (Random under sampling = 0.4 + score>50)* | | |

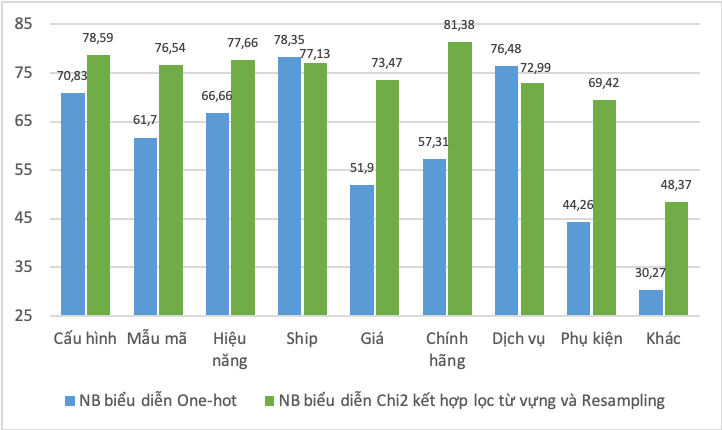
**Bảng 5‑12. Kết quả mô hình với cách biểu diễn Chi2 đã lọc từ kết hợp Resampling trên dữ liệu công nghệ tiki**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Micro (%) | | 70.72 | 74.48 | 72.55 | 68.46 | | 79.61 | 73.61 | 63.56 | 80.23 | 70.92 |
| Macro (%) | | 66.13 | 69.88 | 67.85 | 65.14 | | 77.90 | 70.48 | 60.71 | 78.71 | 68.15 |
| Các khía cạnh (%) | Other | 45.45 | 51.47 | 48.28 | 45.65 | | 46.32 | 45.99 | 46.15 | 35.29 | 40.00 |
| Phụ kiện | 56.52 | 62.90 | 59.54 | 50.00 | | 87.10 | 63.53 | 45.95 | 82.26 | 58.96 |
| Dịch vụ | 35.29 | 33.33 | 34.29 | 37.37 | | 51.39 | 43.27 | 42.48 | 66.67 | 51.89 |
| Chính hãng | 71.03 | 82.40 | 76.30 | 72.41 | | 84.00 | 77.78 | 73.47 | 86.40 | 79.41 |
| Giá | 68.94 | 82.22 | 75.00 | 68.86 | | 85.19 | 76.16 | 63.37 | 80.74 | 71.01 |
| Ship | 87.33 | 88.53 | 87.93 | 87.33 | | 88.53 | 87.93 | 76.52 | 86.70 | 81.29 |
| Hiệu năng | 62.07 | 58.70 | 60.34 | 57.41 | | 67.39 | 62.00 | 52.00 | 70.65 | 59.91 |
| Mẫu mã | 71.59 | 72.41 | 72.00 | 72.25 | | 71.84 | 72.05 | 72.49 | 78.74 | 75.48 |
| Cấu hình | 76.24 | 78.57 | 77.39 | 75.44 | | 87.76 | 81.13 | 59.38 | 77.55 | 67.26 |
| Độ đo | | P | R | F1 | P | | R | F1 | P | R | F1 |
| Mô hình | | *Random Forest (Random over sampling = 0.7 + score > 15)* | | | | *Random Forest (Random under sampling = 0.5 + score > 15)* | | | *Logistic Regression (Random Over sampling = 0.8 + score > 15)* | | |

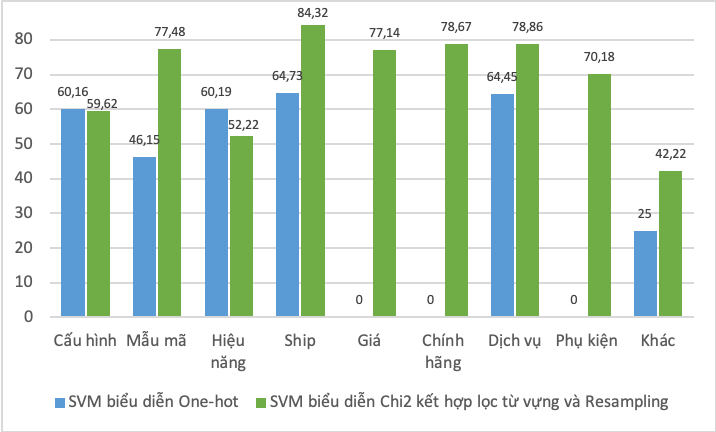
**Bảng 5‑13. Kết quả mô hình với cách biểu diễn Chi2 đã lọc từ kết hợp Resampling trên dữ liệu công nghệ tiki (tiếp)**

Đối với tập dữ liệu Công nghệ Shopee và tập dữ liệu Công nghệ Tiki, sau khi áp dụng biểu diễn Chi2 và xử lý mất cân bằng dữ liệu, đã cho kết quả tăng cao rõ rệt đối với tất cả các khía cạnh so với One-hot.

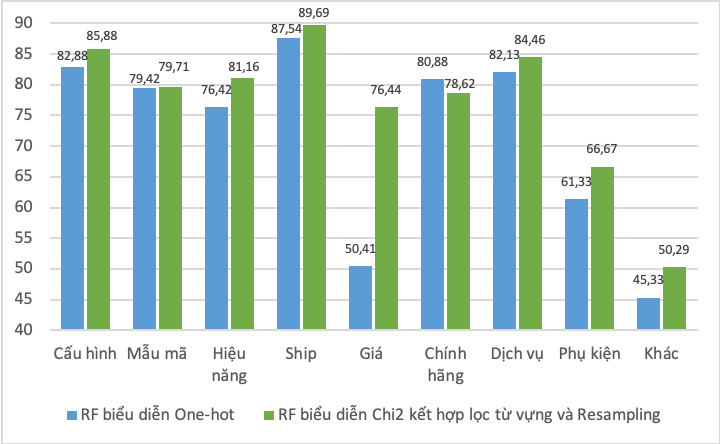
Dưới đây là các biểu đồ so sánh kết quả trước và sau khi áp dụng phương pháp xử lý mất cân bằng dữ liệu và biểu diễn Chi2 trên tập dữ liệu mẹ & bé shopee:



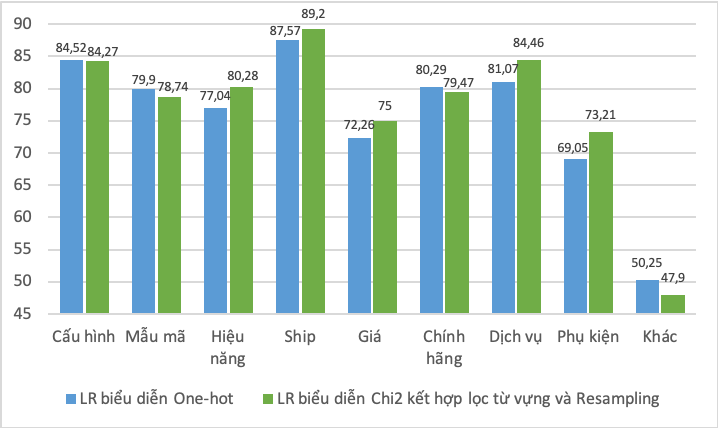
Hình ‑. Kết quả của mô hình Naïve Bayes



Hình ‑. Kết quả của mô hình SVM



Hình ‑. Kết quả của mô hình Random Forest



Hình ‑. Kết quả của mô hình Logistic Regression

Có thể thấy trước khi xử lý mất cân bằng dữ liệu, mô hình phân lớp cho kết quả khá thấp, thậm chí có những khía cạnh mà mô hình không thể dự đoán được. Tuy nhiên, sau khi xử lý mất cân bằng dữ liệu, mô hình đã có thể dự đoán được những khía cạnh đó và ở các khía cạnh còn lại thì hiệu suất đều tăng cao.

Kết luận và định hướng phát triển

Trong bài nghiên cứu này, chúng tôi đã trình bày mô hình xác định khía cạnh của các đánh giá trong dữ liệu Mẹ và Bé Shopee. Chúng tôi cũng đã đề xuất mô hình kiến trúc THANOS để giải quyết vấn đề mất cân bằng dữ liệu giữa các lớp của từng khía cạnh. Mặc dù vậy, do việc gán nhãn thủ công vẫn còn những khó khăn như đồng nhất quan điểm gán nhãn đã khiến cho kết quả mô hình không đươc cao như mong đợi.

Trong tương lai, chúng tôi sẽ xem xét lại việc gán nhãn đồng thời cải thiện hiệu suất phân lớp bằng những mô hình học sâu.

Tài liệu tham khảo

[1] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).

[2] Soman, K. P., Loganathan, R., & Ajay, V. (2009). *Machine learning with SVM and other kernel methods*. PHI Learning Pvt. Ltd..

[3] Segal, M. R. (2004). Machine learning benchmarks and random forest regression.

[4] Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, *35*(5-6), 352-359.

[5] Rodríguez, P., Bautista, M. A., Gonzalez, J., & Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, *75*, 21-31.

[6] Liu, H., & Setiono, R. (1995, November). Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence* (pp. 388-391). IEEE.

[7] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1-13.

[8] Khoshgoftaar, T. M., Golawala, M., & Van Hulse, J. (2007, October). An empirical study of learning from imbalanced data using random forest. In 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007) (Vol. 2, pp. 310-317). IEEE.

[9] Liu, X. Y., Wu, J., & Zhou, Z. H. (2008). Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2), 539-550.

[10] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[11] Han, H., Wang, W. Y., & Mao, B. H. (2005, August). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878-887). Springer, Berlin, Heidelberg.

[12] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221-232.

[13] Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter, 6(1), 20-29.

[14] A Survey of Predictive Modelling under Imbalanced Distributions

[15] R. Maclin, and D. Opitz. “An empirical evaluation of bagging and boosting.” AAAI/IAAI 1997 (1997): 546-551

[16] S. Wang, and X. Yao. “Diversity analysis on imbalanced data sets by using ensemble models.” 2009 IEEE symposium on computational intelligence and data mining. IEEE, 2009.

[17] S. Hido, H. Kashima, and Y. Takahashi. “Roughly balanced bagging for imbalanced data.” Statistical Analysis and Data Mining: The ASA Data Science Journal 2.5‐6 (2009): 412-426.

[18] Feng, W., Huang, W., & Ren, J. (2018). Class imbalance ensemble learning based on the margin theory. Applied Sciences, 8(5), 815.

[19] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2008, May). Building Useful Models from Imbalanced Data with Sampling and Boosting. In FLAIRS conference (pp. 306-311).

[20] Yap, B. W., Abd Rani, K., Abd Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)* (pp. 13-22). Springer, Singapore.