ỨNG DỤNG KỸ THUẬT HỌC MÁY TRÊN DỮ LIỆU MẤT CÂN BẰNG HỖ TRỢ DỰ ĐOÁN SỚM KHẢ NĂNG THÔI HỌC CỦA HỌC SINH TRUNG HỌC PHỔ THÔNG

Võ Đức Quang [(1)], Nguyễn Thị Lan Anh [(2)], Mai Hồng Mận [(3)], Cao Thanh Sơn [(4)]

[1] *Viện Kỹ thuật và Công nghệ, Trường Đại học Vinh*
Nghiên cứu sinh, Trường Đại học Bách Khoa Hà Nội
[2] *Trường THPT Đông Hiếu, Thái Hòa, Nghệ An*
[3] *Lớp 58K4 Công nghệ thông tin, Viện Kỹ thuật và Công nghệ, Trường Đại học Vinh*
[4] *Viện Kỹ thuật và Công nghệ, Trường Đại học Vinh*

Ngày nhận bài 29/5/2020, ngày nhận đăng 13/7/2020

Tóm tắt: Bài báo đề xuất một mô hình học máy cho bài toán phân lớp trên tập dữ liệu mất cân bằng, trong đó sử dụng kết hợp kỹ thuật sinh mẫu tổng hợp SMOTE và giải thuật AdaBoost cho thuật toán Cây quyết định. Các tác giả đã tiến hành thực nghiệm đánh giá so sánh hiệu quả phân lớp của mô hình đã đề xuất với các giải thuật Cây quyết định sử dụng entropy và chỉ số Gini trên bộ dữ liệu thực tế thu thập tại Trường trung học phổ thông (THPT) Đông Hiếu, Thái Hòa, Nghệ An từ năm 2014 đến năm 2019. Kết quả nghiên cứu có thể sử dụng làm nền tảng để xây dựng ứng dụng hỗ trợ dự đoán sớm khả năng thôi học của học sinh THPT, có ý nghĩa góp phần nâng cao chất lượng giáo dục đào tạo của nhà trường và các cấp quản lý giáo dục.

Từ khóa: Học máy; khai phá dữ liệu; dữ liệu mất cân bằng; Cây quyết định; AdaBoost, SMOTE.

1. Mở đầu

Phân lớp dữ liệu là một bài toán phổ biến trong các ứng dụng khai phá dữ liệu, xây dựng các hệ dự đoán, dự báo hay khuyến nghị... nhằm hỗ trợ con người trong nhiều lĩnh vực của đời sống. Các phương pháp giải quyết bài toán phân lớp thường sử dụng mô hình dạng luật hoặc sử dụng các giải thuật học máy như: Cây quyết định, Mạng nơ-ron, Naïve Bayes, Support Vector Machines… Trong nhiều trường hợp, các giải thuật này không đạt hiệu quả cao khi các bộ dữ liệu có sự chênh lệch lớn về số lượng mẫu học của các nhãn lớp, gọi là bộ dữ liệu mất cân bằng. Trong bộ dữ liệu đó, nhãn lớp có số lượng mẫu học lớn được gọi là lớp đa số (nhãn âm, nhãn tiêu cực, thường được ký hiệu là 0 hoặc -1); nhãn lớp có số lượng mẫu học ít được gọi là lớp thiểu số (nhãn dương, nhãn tích cực, thường được ký hiệu là +1). Tuy nhiên, các bộ dữ liệu mất cân bằng này lại xuất hiện rất phổ biến trong các bài toán quan trọng phát hiện các trường hợp hiếm gặp, như chuẩn đoán bệnh trong y học, dự đoán sớm khả năng thôi học trong trường học, phát hiện sự cố môi trường, phát hiện gian lận giao dịch, phát hiện tấn công mạng... Tùy thuộc vào từng trường hợp và hoàn cảnh cụ thể, độ mất cân bằng có thể khác nhau, từ mức độ nhỏ, vừa phải, cho đến các trường hợp tỷ lệ mất cân bằng rất lớn đến cực kỳ lớn, có thể lên đến 1:100 thậm chí 1:10.000. Khi đó, nếu áp dụng các thuật toán học máy truyền thống trên các tập dữ liệu mất cân bằng, hầu như các phần tử thuộc lớp nhãn đa số sẽ được phân lớp đúng và các phần tử thuộc lớp thiểu số cũng sẽ dễ bị nhận diện gán nhãn nhầm là nhãn lớp đa số. Điều này là dễ hiểu bởi vì các giải thuật học máy sẽ điều chỉnh theo hướng phân lớp chính xác tối đa số mẫu, trong trường hợp mất cân bằng thì số mẫu nhãn âm là đa số dẫn đến mô hình phân lớp sẽ quá "khớp" với dữ liệu lớp đa số. Điều này dẫn

Email: quangvd@vinhuni.edu.vn (V. Đ. Quang)

đến mô hình phân lớp sẽ cho kết quả với độ chính xác (accuracy) rất cao trong khi giá trị độ nhạy (sensitivity) lại rất thấp. Do vậy, các giải thuật phân lớp có thể thực hiện hiệu quả trên dữ liệu khá cân bằng nhưng lại cho kết quả không tốt với các tập dữ liệu mất cân bằng. Điều này tạo nên sự thú vị và tạo động lực trong việc nghiên cứu các phương pháp cải tiến mô hình phân lớp khi áp dụng cho bài toán dữ liệu mất cân bằng.

Nhiều phương pháp đã được đề xuất để giải quyết vấn đề này [5], chủ yếu được phân thành hai nhóm cơ bản: tiếp cận ở mức giải thuật và tiếp cận ở mức dữ liệu.

- Tiếp cận ở mức giải thuật hướng tới việc điều chỉnh các thuật toán phân lớp mạnh truyền thống để vẫn có hiệu quả cao trên các tập dữ liệu mất cân bằng. Một số phương pháp đã được các nhà nghiên cứu đề xuất như: Điều chỉnh xác suất ước lượng, sử dụng các hằng số phạt khác nhau cho các nhãn lớp khác nhau.

- Tiếp cận ở mức dữ liệu nhằm tạo ra sự phân bố cân bằng hơn về số lượng mẫu giữa các nhãn lớp, các kỹ thuật thường sử dụng: (*i*) sinh thêm các phần tử cho lớp thiểu số: SMOTE [7], ADA-SYN, OSD... (*ii*) loại bỏ bớt các phần tử thuộc lớp đa số: NearMiss, SMOTE với Tomek links [8]…

Bên cạnh đó, trong lĩnh vực giáo dục, các báo cáo thống kê về giáo dục phổ thông trong những năm gần đây từ các sở giáo dục và đào tạo cho thấy hiện tượng học sinh thôi học ở bậc trung học cơ sở (THCS) và THPT xảy ra khá phổ biến. Có nhiều nguyên nhân dẫn đến tình trạng học sinh thôi học như: do kết quả học tập, hoàn cảnh gia đình và cá nhân, môi trường tác động, cơ sở hạ tầng giáo dục, sự thay đổi cách dạy và học từ THCS lên THPT... Trước những thách thức đó, các nhà hoạch định chính sách cũng như quản lý giáo dục ở cơ sở cần phải tìm hiểu các nguyên nhân để cải thiện chất lượng công tác giảng dạy và hỗ trợ người học. Song song với đó, việc theo dõi, rà soát hiện trạng, đưa ra những dự đoán phát hiện sớm các học sinh có thể thôi học có ý nghĩa rất lớn trong việc đưa ra các giải pháp, tư vấn, hỗ trợ kịp thời để giảm thiểu tối đa việc học sinh thôi học [3, 4].

Với nền tảng là các giải thuật học máy, chúng ta có thể hoàn toàn xây dựng được một hệ thống dự đoán sớm khả năng thôi học của học sinh thông qua các giải thuật phân lớp nhị phân, với nhãn -1 là dữ liệu học sinh theo học bình thường, nhãn +1 gán cho dữ liệu học sinh thôi học; các thuộc tính của mỗi mẫu dữ liệu cần thu thập là các thông tin của học sinh có thể ảnh hưởng đến việc thôi học như: giới tính, hoàn cảnh gia đình, hạnh kiểm, học lực, điểm đầu vào, nghề nghiệp bố và mẹ… Do số lượng học sinh thôi học so với học sinh theo học chiếm tỷ lệ rất nhỏ nên bộ dữ liệu này trở thành bộ dữ liệu mất cân bằng nhị phân.

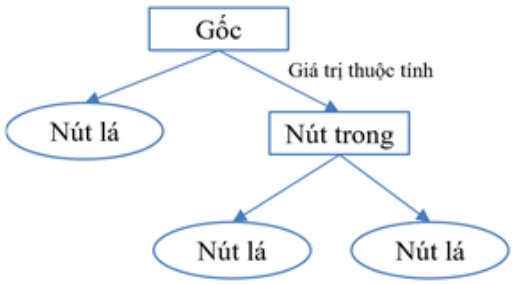
Trong bài báo này, các tác giả tiến hành thu thập dữ liệu thực tế về tình trạng học sinh thôi học tại Trường THPT Đông Hiếu, Thái Hòa, Nghệ An, từ đó thử nghiệm sử dụng các phương pháp học máy trên dữ liệu mất cân bằng để xây dựng mô hình phân lớp dự đoán sớm học sinh thôi học. Bài báo tiến hành đánh giá, so sánh các kết quả phân lớp dự đoán đối với các giải thuật Cây quyết định [3], Cây quyết định kết hợp AdaBoost và đánh giá hiệu quả của kỹ thuật lấy mẫu OverSampling SMOTE.

2. Các giải thuật phân lớp cơ sở

2.1. Thuật toán phân lớp dựa vào Cây quyết định

Cây quyết định (decision tree) là một kiểu mô hình dự đoán có cấu trúc phân cấp của các nút và các nhánh được biểu diễn dưới dạng cây [1, 2]. Cây quyết định có ba loại nút: nút gốc (root), nút trong (internal node) và nút lá (leaf node). Nó có thể được dùng

để phân lớp bằng cách xuất phát từ nút gốc và di chuyển theo các nhánh cho đến khi gặp nút lá. Trên cơ sở phân lớp này, chúng ta có thể chuyển đổi về các luật quyết định (dạng if-then). Mỗi nút trong biểu diễn một thuộc tính, các nhánh biểu diễn giá trị có thể có của thuộc tính, mỗi nút lá biểu diễn giá trị của nhãn lớp. Hình 1 biểu diễn một cây quyết định tổng quát.



Hình 1: *Cây quyết định tổng quát*

Tạo cây quyết định chính là quá trình phân tích cơ sở dữ liệu, phân lớp và đưa ra dự đoán. Cây quyết định được tạo thành bằng cách lần lượt chia một tập dữ liệu thành các tập con, mỗi tập con được tạo thành chủ yếu từ các phần tử của cùng một lớp. Lựa chọn thuộc tính để tạo nhánh thường dựa vào entropy hoặc chỉ số Gini.

Xét các tập dữ liệu sau:

+ $C = \{C_1, C_2, \ldots, C_m\}$: thuộc tính phân lớp;

+ D: tập dữ liệu huấn luyện có thuộc tính phân lớp C;

+ $D = D_1 \cup D_2 \cup \ldots \cup D_t$: phân hoạch trên D với $D_i \cap D_j = \varnothing$.

Để thực hiện quá trình phân lớp, chúng ta cần tìm kiếm các độ đo để đánh giá mức độ đồng nhất của các đối tượng dựa trên thuộc tính phân lớp và từ đó chọn độ đo để tìm ra phân hoạch của D có mức độ đồng nhất cực đại. Một số độ đo phổ biến thường được dùng gồm entropy hoặc chỉ số Gini.

- Entropy của tập dữ liệu là lượng thông tin cần thiết để phân loại một phần tử trong tập dữ liệu huấn luyện D, ký hiệu: $Info(D)$.

- Đặt p_i là xác suất của một phần tử bất kỳ trong D thuộc vào lớp C_i với $1 \le i \le m$.

- Đặt D_i là tập các phần tử trong D thuộc về lớp C_i. Ta có:

$$p_i = \frac{|D_i|}{|D|},$$
$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i). \quad (2.1)$$

- Entropy của tập dữ liệu ứng với thuộc tính A là lượng thông tin cần để phân loại một phần tử trong tập dữ liệu D dựa trên thuộc tính A, ký hiệu $Info_A(D)$, trong đó thuộc tính A dùng để tách D thành t phân hoạch tương ứng là $D_1, D_2, \ldots, D_t$. Mỗi phân hoạch D_j có $|D_j|$ phần tử, với $1 \le j \le t$. Lượng thông tin này cho biết mức độ trùng lặp giữa các phân hoạch và mong đợi $Info_A(D)$ càng nhỏ càng tốt.

Ta có công thức tính $Info_A(D)$ như sau:

$$Info_A(D) = \sum_{j=1}^{t} \frac{|D_j|}{|D|} \times Info(D_j). \quad (2.2)$$

- Độ lợi thông tin (information gain) có mục đích tối thiểu hoá lượng thông tin cần thiết để phân lớp các mẫu dữ liệu. Độ lợi thông tin ứng với thuộc tính *A*, ký hiệu *Gain*(*A*), là độ sai lệch giữa entropy ban đầu của tập dữ liệu (trước khi phân hoạch) và entropy của dữ liệu với thuộc tính *A* (sau khi phân hoạch bởi *A*). Để tạo nhánh trong cây quyết định, ta chọn thuộc tính có độ lợi thông tin *Gain*(A) lớn nhất.

$$Gain(A) = Info(D) - Info_A(D). \quad (2.3)$$

- Chỉ số Gini (Gini Index) dựa vào bình phương các xác suất thành viên cho mỗi thể loại đích trong nút. Giá trị của nó tiến đến bằng 0 khi mọi trường hợp trong nút rơi vào một thể loại đích duy nhất.

Giả sử $y = \{1, 2,..., n\}$, gọi $f(i, j)$ là tần suất của giá trị j trong nút i, khi đó $f(i, j)$ là tỷ lệ các bản ghi với $y = j$ được xếp vào nhóm i. Ta có công thức:

$$I_G(i) = 1 - \sum_{j=1}^{n} f(i,j)^2. \quad (2.4)$$

2.2. Kỹ thuật AdaBoost kết hợp Cây quyết định

Boosting là kỹ thuật sử dụng kết hợp các thuật toán học máy trên quần thể không gian mẫu một cách tuần tự, sau đó thực hiện tổng hợp các kết quả phân lớp riêng để được một bộ phân lớp hiệu quả. Một giải thuật hiệu quả trong Boosting là AdaBoost (Adaptive Boosting) [6], sử dụng các trọng số phân bổ lỗi gán cho từng mẫu như được chỉ ra trong Giải thuật 1. Thuật toán ban đầu phân bổ các trọng số tương đương trên mỗi mẫu huấn luyện. Trong mỗi bước lặp, thuật toán tiến hành: (*i*) huấn luyện mẫu bởi một bộ phân loại yếu; (*ii*) kiểm tra lại kết quả phân lớp trên mẫu huấn luyện đó có chính xác không; (*iii*) tính toán lại phân bổ trọng số lỗi trên các mẫu theo hướng: tăng trọng số lỗi trên các mẫu bị phân loại sai và giảm trọng số lỗi trên các mẫu được phân loại đúng. Sau khi kết thúc các vòng lặp, giải thuật sẽ tiến hành tổng hợp các bộ phân lớp thành viên thành bộ phân lớp tổng hợp.

Giải thuật 1: Giải thuật AdaBoost

Input: Tập *N* mẫu dữ liệu X_{Train}, $X_{Validation}$, *M*: số lần lặp tối đa, ω_i: phân bổ trọng số lỗi
Output: *H*: Bộ phân lớp tổng hợp
Begin
Initialize: $\omega_i = 1/N$, $T=1$, h_m; /∗ bộ phân loại thành viên ∗/
For $m = 1, 2,\ldots, M$

(*a*) $X_{train}(x)$ sử dụng ω_i

(*b*) $h_m \leftarrow$ Train(X_{train})

(*c*) Tính đại lượng: $\varepsilon_m = \frac{\sum_{i=1}^{N} \omega_i I(y_i \neq Classify(X_{Train}, h_m))}{\sum_{i=1}^{N} \omega_i}$

(*d*) Tính toán tham số mô hình: $\alpha_m = \lambda.\log \frac{1-\varepsilon_m}{\varepsilon_m}$ $(0 < \lambda \leq 1)$

(*e*) Thiết lập lại phân bổ trọng số lỗi:

$\omega_{i+1} \leftarrow \omega_i . exp[\alpha_m . I(y_i \neq Classify(X_{Train}, h_m))]$, $i = 1, 2,\ldots,N$

(*f*) $H_m = sign[\sum_{j=1}^{m} \alpha_j h_j]$

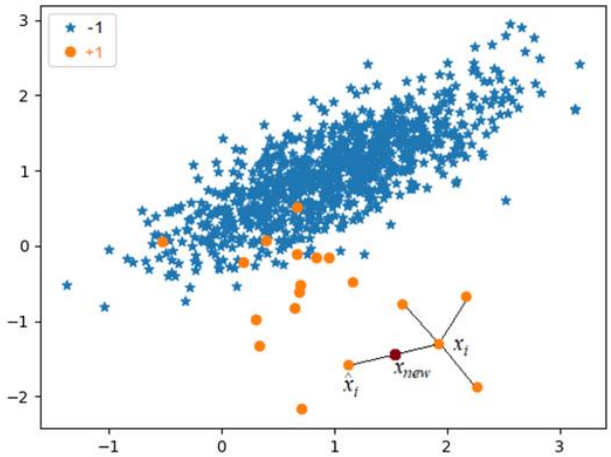
Return H_T.
End

Chú ý rằng trong Giải thuật 1, tập dữ liệu X với N mẫu ban đầu được chia vào tập X_{Train}, $X_{Validation}$, trong đó, mỗi mẫu được gán 1 trọng số lỗi ban đầu như nhau ω_i=1/N. Trong mỗi vòng lặp, bộ phân lớp h_m được sử dụng để phân lớp bộ X_{train}. Từ kết quả phân lớp nhận được, giải thuật sẽ kiểm tra việc phân loại chính xác trên mỗi mẫu. Tính toán tham số α_m cho bộ phân lớp h_m ở bước (*d*) thông qua đại lượng trung gian ε_m ở bước (*c*). Từ đó, tính toán phân bổ trọng số lỗi ω_{i+1} theo hướng tăng trọng số nếu mẫu bị phân loại sai, giảm bớt trọng số nếu mẫu được phân loại đúng. Việc tính toán này được thực hiện thông qua công thức ở bước (*e*). Bước (*f*) tiến hành tạo bộ phân lớp tổng hợp H_m dựa trên tham số α_m. Nhãn phân lớp được xác định dựa vào hàm dấu: nhãn (+1) khi $H_m > 0$ và ngược lại, nhãn (-1) khi $H_m < 0$.

2.3. Kỹ thuật lấy mẫu OverSampling SMOTE

Như đã đề cập ở Phần 1, một mô hình giải thuật học máy cho tỷ lệ chính xác cao trên bộ dữ liệu mất cân bằng, nhưng trong thực tế tỷ lệ này có thể không mang nhiều ý nghĩa. Ví dụ, giả sử bộ dữ liệu có 100 mẫu, với 95 nhãn âm (-1), 05 mẫu nhãn dương (+1). Nếu mô hình cho kết quả phân lớp dự đoán đúng 92 nhãn (-1) và 01 mẫu nhãn (+1), khi đó tỷ lệ phân loại chính xác lên đến 93%, tuy nhiên mô hình không có nhiều ý nghĩa vì chỉ phân lớp dự đoán đúng 01 trong 05 mẫu nhãn lớp quan trọng nhãn (+1). Để tận dụng và cải thiện chất lượng phân lớp của các giải thuật học máy, nhiều nghiên cứu đã tiếp cận theo hướng sử dụng các kỹ thuật lấy mẫu (Sampling): sinh ra các mẫu tổng hợp cho các nhãn (+1) (OverSampling) và giảm số lượng các mẫu nhãn (-1) (UnderSampling) nhằm mục đích cải thiện tỷ lệ số lượng mẫu giữa các nhãn lớp cân bằng hơn. Trong bài báo này, chúng tôi sử dụng kỹ thuật OverSampling phổ biến SMOTE (Synthetic Minority Over-sampling) [7] để điều chỉnh mức độ cân bằng của bộ dữ liệu. Kỹ thuật này nhằm mục đích tạo ra các dữ liệu nhân tạo dựa trên các không gian đặc tính tương đồng với các mẫu nhóm thiểu số. SMOTE sử dụng giải thuật K-láng giềng gần nhất KNN (K-Nearest Neighbor), tính toán các khoảng cách trên các không gian thuộc tính của các mẫu trong nhóm thiểu số; từ đó làm cơ sở để tạo ra mẫu tổng hợp mới với sự khác biệt không gian thuộc tính là nhỏ nhất. Mẫu tổng hợp x_{new} dựa trên việc chọn K láng giềng và một số thực $\delta \in [0, 1]$ để xác định vị trí:

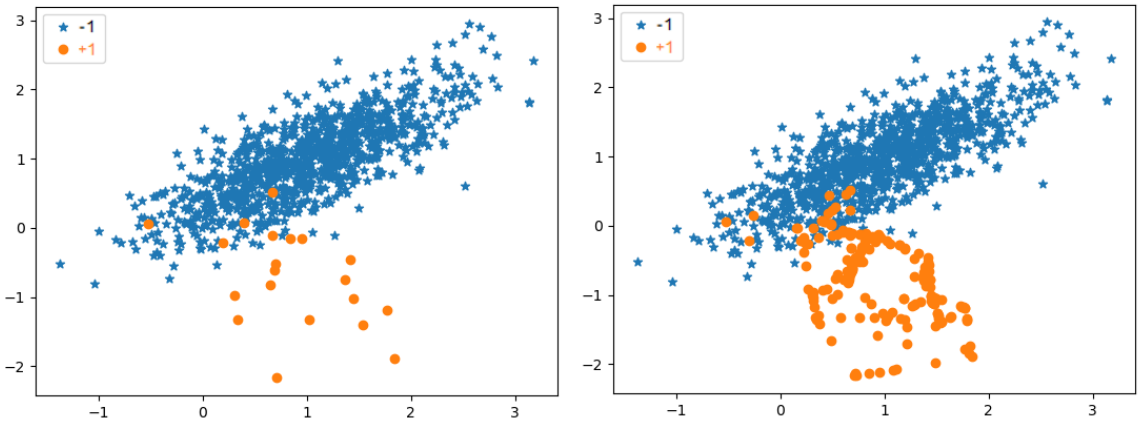
$$x_{new} = x_i + (\hat{x}_i - x_i) * \delta \quad (2.5)$$



Hình 2: *Minh họa quá trình lấy mẫu*

Để minh họa, giả sử các mẫu nhãn lớp thiểu số (+1) được biểu diễn là đối tượng hình tròn, nhãn lớp đa số (-1) được minh họa là đối tượng hình sao như Hình 2. Để sinh K mẫu tổng hợp từ mẫu x_i, giải thuật tìm K láng giềng gần x_i nhất. Sau đó, dựa vào δ để tổng hợp nên một mẫu mới x_{new} trên "đường đi" từ x_i đến các láng giềng Hình 2 dựa vào công thức (2.5).

Việc chọn K và δ sẽ tạo nên các mẫu tổng hợp mới cho đến khi đạt được tỷ lệ cân bằng mẫu mong muốn. Hình 3 minh họa bộ dữ liệu 1000 mẫu trong đó có 20 mẫu có nhãn (+1) và 980 mẫu nhãn (-1), tỷ lệ mất cân bằng 1:49; sau khi thử nghiệm kỹ thuật SMOTE thì đạt tỷ lệ cân bằng mẫu 1:5.

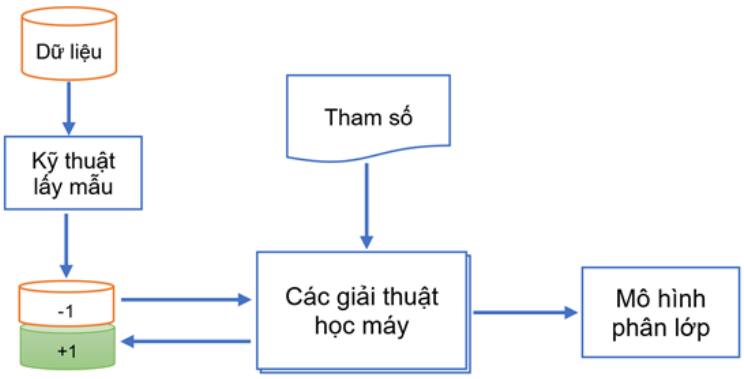


Hình 3: *Mẫu dữ liệu được sinh với kỹ thuật SMOTE*

3. Đề xuất mô hình phân lớp dự đoán

Dựa trên các phương pháp và giải thuật đã trình bày ở Mục 2, trong phần này, chúng tôi đề xuất mô hình phân lớp cho ứng dụng dự đoán sớm khả năng học sinh thôi học. Với bộ dữ liệu mất cân bằng, mô hình sử dụng kỹ thuật sinh mẫu tổng hợp SMOTE cho lớp nhãn thiểu số, đưa bộ dữ liệu về trạng thái cân bằng hơn. Áp dụng các giải thuật học máy phổ biến như Cây quyết định và AdaBoost kết hợp Cây quyết định để tiến hành huấn luyện trên mẫu, chúng tôi đề xuất mô hình phân lớp dự đoán như Hình 4.

Trong quá trình huấn luyện trên mẫu thử, mô hình liên tục được đánh giá lại, dựa trên kỹ thuật AdaBoost đã trình bày ở Mục 2.2, để thích nghi với đặc trưng dữ liệu.



Hình 4: *Mô hình phân lớp dự đoán*

4. Thử nghiệm và kết quả

Các bước thực hiện thử nghiệm phân lớp dự đoán sớm khả năng thôi học của học sinh được mô tả như sau:

- Bước 1: Thu thập dữ liệu thông tin học sinh tại Trường THPT Đông Hiếu, Thị xã Thái Hòa, Nghệ An từ năm 2014 đến 2019;

- Bước 2: Tiền xử lý dữ liệu: xử lý những dữ liệu còn thiếu; chuyển đổi số hóa các giá trị thuộc tính; sử dụng kỹ thuật sinh mẫu SMOTE để giảm tỷ lệ mất cân bằng;

- Bước 3: Chia tập dữ liệu huấn luyện và kiểm tra theo các kịch bản thử nghiệm;

- Bước 4: Thực hiện huấn luyện tạo mô hình theo các giải thuật đề xuất;

- Bước 5: Phân tích đánh giá kết quả thử nghiệm.

Bộ dữ liệu thử nghiệm là kết quả thu thập thông tin học sinh tại Trường THPT Đông Hiếu, Thái Hòa, Nghệ An từ năm 2014 đến 2019. Bộ dữ liệu gồm 828 mẫu, trong đó có 101 mẫu thông tin về học sinh thôi học (nhãn +1), tức là tỷ lệ mất cân bằng xấp xỉ 1:8. Ngoài thuộc tính nhãn, mỗi mẫu dữ liệu có 14 thuộc tính được chuẩn hóa sang dạng số, bao gồm: Giới tính, Lịch sử kỷ luật, Lịch sử khen thưởng, Mức sống gia đình, Nghề nghiệp của bố, Nghề nghiệp của mẹ, Học lực THCS, Hạnh kiểm THCS, Điểm tuyển sinh, Khoảng cách địa lý, Tình trạng vắng học, Số anh chị em, Tình trạng gia đình, Ý thức học tập.

Kịch bản thử nghiệm đã tiến hành phân lớp dự đoán sử dụng các giải thuật Cây quyết định với chỉ số Gini và entropy; Cây quyết định kết hợp AdaBoost; sử dụng kỹ thuật lấy mẫu OverSampling SMOTE trên bộ dữ liệu đã thu thập với tỷ lệ mẫu huấn luyện và thử nghiệm (Training/Test) khác nhau. Các kết quả được đánh giá dựa trên các độ đo Accuracy, Confusion Matrix, Precision, Recall, F1-Score. Kết quả thu được với bộ dữ liệu ban đầu với tỷ lệ mất cân bằng khoảng 1:7 được thể hiện trong Bảng 1, trong đó *T* là tỷ lệ phần trăm của tập dữ liệu thử nghiệm trích từ tập dữ liệu ban đầu.

Bảng 1: *Tập dữ liệu với tỷ lệ mất cân bằng 1:7*

Giải thuật	***T* = 0.3**		***T* = 0.5**		***T* = 0.7**	
	Ma trận nhầm lẫn	**Độ chính xác**	**Ma trận nhầm lẫn**	**Độ chính xác**	**Ma trận nhầm lẫn**	**Độ chính xác**
Cây quyết định với chỉ số Gini	$\begin{bmatrix}214 & 5\\ 1 & 29\end{bmatrix}$	0.9759	$\begin{bmatrix}358 & 6\\ 4 & 46\end{bmatrix}$	0.9758	$\begin{bmatrix}506 & 3\\ 13 & 58\end{bmatrix}$	0.9724
Cây quyết định với entropy	$\begin{bmatrix}216 & 3\\ 1 & 29\end{bmatrix}$	0.9839	$\begin{bmatrix}358 & 6\\ 4 & 46\end{bmatrix}$	0.9758	$\begin{bmatrix}506 & 3\\ 13 & 58\end{bmatrix}$	0.9724
Cây quyết định với AdaBoost	$\begin{bmatrix}\mathbf{217} & 2\\ 1 & 29\end{bmatrix}$	**0.9879**	$\begin{bmatrix}\mathbf{362} & 2\\ 5 & 45\end{bmatrix}$	**0.9830**	$\begin{bmatrix}505 & 5\\ \mathbf{6} & 65\end{bmatrix}$	**0.9810**

Bảng 2: *Kết quả thực nghiệm*

Tỷ lệ dữ liệu	T	Cây quyết định với chỉ số Gini			Cây quyết định với Entropy			Cây quyết định với AdaBoost		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
1:7	**0.3**	0.9241	0.9719	0.9462	**0.9508**	0.9765	**0.9632**	0.9470	0.9598	0.9533
	0.4	0.9250	0.9522	0.9380	0.9250	0.9522	0.9380	**0.9520**	0.9324	**0.9419**
	0.5	0.9368	0.9518	0.9441	0.9368	0.9518	0.9441	**0.9792**	0.9286	**0.9520**
	0.6	**0.9489**	0.9359	0.9423	**0.9489**	0.9359	0.9423	0.9378	0.9503	**0.9439**
	0.7	**0.9629**	0.9055	0.9316	**0.9629**	0.9055	0.9316	0.9487	0.9378	**0.9431**
	0.8	0.9191	0.9378	0.9281	0.9191	0.9378	0.9281	**0.9427**	0.9477	**0.9452**
1:5 SMOTE	**0.3**	**0.9664**	0.9822	**0.9741**	**0.9664**	0.9822	**0.9741**	0.9634	0.9713	0.9673
	0.4	0.9474	0.9806	**0.9630**	0.9474	0.9806	**0.9630**	**0.9527**	0.9730	0.9625
	0.5	0.9243	0.9348	0.9295	0.9243	0.9348	0.9295	**0.9524**	0.9692	**0.9606**
	0.6	0.9274	0.9674	0.9459	0.9274	0.9674	0.9459	**0.9552**	0.9682	**0.9616**
	0.7	0.9522	0.9522	0.9522	0.9522	0.9522	0.9522	**0.9603**	0.9491	**0.9546**
	0.8	0.9401	0.9496	0.9448	0.9401	0.9496	0.9448	**0.9481**	0.9513	**0.9497**
1:3 SMOTE	**0.3**	**0.9768**	0.9680	**0.9723**	**0.9768**	0.9680	**0.9723**	0.9661	0.9703	0.9681
	0.4	0.9536	0.9702	0.9615	0.9536	0.9702	0.9615	**0.9663**	0.9698	**0.9681**
	0.5	0.9478	0.9708	0.9586	0.9478	0.9708	0.9586	**0.9651**	0.9733	**0.9691**
	0.6	**0.9657**	0.9700	0.9679	**0.9657**	0.9700	0.9679	0.9652	0.9758	**0.9704**
	0.7	**0.9629**	0.9609	0.9619	**0.9629**	0.9609	0.9619	0.9594	0.9651	**0.9622**
	0.8	**0.9560**	0.9576	0.9568	**0.9560**	0.9576	0.9568	0.9557	0.9620	**0.9588**
1:2 SMOTE	**0.3**	0.9453	0.9583	0.9514	0.9495	0.9606	0.9547	**0.9742**	0.9766	**0.9755**
	0.4	0.9569	0.9653	0.9609	0.9602	0.9669	0.9634	**0.9772**	0.9754	**0.9763**
	0.5	0.9603	0.9659	0.9647	0.9629	0.9709	0.9667	**0.9696**	0.9764	**0.9729**
	0.6	0.9673	0.9747	0.9709	0.9694	0.9759	0.9725	**0.9750**	0.9805	**0.9777**
	0.7	0.9713	0.9765	0.9738	**0.9742**	0.9794	**0.9767**	0.9561	0.9561	0.9561
	0.8	**0.9647**	0.9584	**0.9614**	**0.9647**	0.9584	**0.9614**	0.9598	0.9583	0.9591
1:1.5 SMOTE	**0.3**	0.9778	0.9828	0.9801	0.9818	0.9851	0.9829	**0.9886**	0.9886	**0.9886**
	0.4	0.9775	0.9840	0.9805	0.9775	0.9840	0.9805	**0.9859**	0.9879	**0.9869**
	0.5	0.9780	0.9844	0.9810	0.9780	0.9844	0.9810	**0.9867**	0.9890	**0.9878**
	0.6	0.9777	0.9828	0.9801	0.9777	0.9828	0.9801	**0.9802**	0.9828	**0.9814**
	0.7	**0.9780**	0.9833	**0.9804**	**0.9780**	0.9833	**0.9804**	0.9745	0.9794	0.9768
	0.8	**0.9791**	0.9828	**0.9802**	**0.9791**	*0.9828*	**0.9802**	0.9640	0.9682	0.9659
1:1 SMOTE	**0.3**	0.9722	0.9732	0.9725	0.9722	0.9732	0.9725	**0.9815**	0.9819	**0.9817**
	0.4	0.9776	0.9780	0.9777	0.9776	0.9780	0.9777	**0.9810**	0.9812	**0.9811**
	0.5	0.9781	0.9782	0.9780	0.9781	0.9782	0.9780	**0.9835**	0.9836	**0.9835**
	0.6	**0.9818**	0.9817	**0.9817**	**0.9818**	0.9817	**0.9817**	**0.9818**	0.9817	**0.9817**
	0.7	**0.9816**	0.9812	**0.9813**	**0.9816**	0.9812	**0.9813**	0.9807	0.9802	0.9803
	0.8	**0.9822**	0.9818	**0.9819**	**0.9822**	0.9818	**0.9819**	0.9761	0.9758	0.9759

Kết quả thử nghiệm mô tả trong Bảng 2 đưa đến một số nhận xét sau:

- Với bộ dữ liệu ban đầu tỷ lệ mất cân bằng ~1:7 giải thuật AdaBoost kết hợp Cây quyết định cho kết quả phân lớp cao hơn so với việc chỉ sử dụng Cây quyết định với entropy và chỉ số Gini, đặc biệt khi tỷ lệ mẫu thử nghiệm tăng dần. Khi chiếu theo hàng ngang, các giá trị được bôi đậm là giá trị lớn nhất của độ đo Precision, các giá trị gạch chân là giá trị lớn nhất của độ đo Recall và các giá trị gạch chân - đậm là giá trị lớn nhất của độ đo F1-Score tương ứng với từng giải thuật Cây quyết định với chỉ số Gini, entropy hay AdaBoost (trong đó *T* là tỷ lệ phần trăm của tập dữ liệu thử nghiệm trích từ tập dữ liệu ban đầu).

- Khi sử dụng kỹ thuật sinh mẫu tổng hợp SMOTE cho bộ dữ liệu để độ mất cân bằng giảm dần từ 1:7 xuống 1:5, 1:3, 1:2, 1:1.5, 1:1 các độ đo Precision, Recall, F1-Score đều tăng lên (chiếu theo cột dọc), việc phân lớp chính xác trên các mẫu nhãn dương cũng tăng. Điều này cho thấy hiệu quả của việc sinh mẫu tổng hợp tác động rất tốt đến mô hình phân lớp.

5. Kết luận

Trong bài báo này, chúng tôi đã đề xuất một mô hình phân lớp dự đoán áp dụng các kỹ thuật xử lý với dữ liệu mất cân bằng hai nhãn lớp bằng cách kết hợp kỹ thuật sinh tổng hợp mẫu mới SMOTE và giải thuật AdaBoost cho Cây quyết định. Áp dụng mô hình đã đề xuất, thực nghiệm dự đoán khả năng thôi học của học sinh Trường THPT Đông Hiếu, Thái Hòa, Nghệ An cho kết quả chính xác cao. Các kết quả thực nghiệm chỉ ra rằng khi kết hợp kỹ thuật sinh mẫu tổng hợp SMOTE với giải thuật AdaBoost với các giải thuật Cây quyết định cho chất lượng tốt hơn việc chỉ dùng các giải thuật Cây quyết định thuần túy khi ứng dụng trên bộ dữ liệu mất cân bằng.

Về mặt ứng dụng, chúng tôi đã xây dựng được một ứng dụng cho phép dự đoán sớm khả năng thôi học của học sinh; từ đó giúp nhà trường, các cơ sở đào tạo, cán bộ giảng dạy kịp thời đưa ra các giải pháp quan tâm, hỗ trợ, động viên học sinh để hạn chế khả năng thôi học. Theo đó, ứng dụng trở thành một yếu tố góp phần nâng cao chất lượng giáo dục và đào tạo của đơn vị, đóng góp cho sự phát triển của nền giáo dục nước nhà. Trong thời gian tới, chúng tôi sẽ cải tiến giải thuật AdaBoost để thu được những kết quả tốt hơn cho tập dữ liệu mất cân bằng với tỷ lệ chênh lệch lớn.

TÀI LIỆU THAM KHẢO

[1] Lê Văn Phùng, Quách Xuân Trưởng, *Khai phá dữ liệu*, NXB Thông tin và Truyền thông, 2017.

[2] Nguyễn Hà Nam, Nguyễn Trí Thành, Hà Quang Thụy, *Giáo trình khai phá dữ liệu*, NXB Đại học Quốc gia Hà Nội, 2013.

[3] P. Saurabh, "Mining educational data to reduce dropout rates of engineering students", *International Journal of Information Engineering and Electronic Business*, Vol. 2, pp. 1-7, 2012.

[4] S. Rai, P. Saini, A. K. Jain, "Model for Prediction of Dropout Student Using ID3 Decision Tree Algorithm", *International Journal of Advanced Research in Computer Science & Technology*, Vol. 2 (1), pp. 142-149, 2014.

[5] F. Hilario et. al., "Learning from Imbalanced Data Sets", *Artificial Intelligence*, Springer, 2018.

[6] Y. Sun, M.S. Kamel, Y. Wang, "Boosting for Learning Multiple Classes with Imbalanced Class Distribution", *Proc. Int'l Conf. Data Mining*, pp. 592-602, 2006.

[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique", *Journal of Artificial Intelligence Research*, Vol. 16, pp. 321-357, 2002.

[8] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang, "Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data", *In Proc. of IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, pp. 225-228, 2016.

SUMMARY

APPLYING MACHINE LEARNING TECHNIQUES ON IMBALANCED DATASETS FOR EARLY PREDICTION OF HIGH SCHOOL STUDENT DROPOUT

This paper proposes a model for the classification problem on imbalanced datasets, which uses a combination of the SMOTE model and the AdaBoost algorithm for the decision tree algorithm. We make a comparison between the proposed model and the decision tree algorithm using the Gini index and entropy on the collected datasets at Dong Hieu high school, Thai Hoa, Nghe An from 2014 to 2019. The research results can be used as a framework to develop applications supporting the early prediction of the ability of students' dropout. Based on that results, the managers can analyze and come up with appropriate solutions in order to decrease the school dropout rate.

Keywords: Machine learning; data mining; imbalanced dataset; decision tree; AdaBoost; SMOTE.