Phân Tích và Dự Đoán Kết Quả Bóng Đá Sử Dụng Dữ Liệu Lịch Sử

Lương Anh Huy
Trường Đại học Công nghệ Thông tin
ĐHQG TP.HCM
Khoa Khoa Học Máy tính
22520550@gm.uit.edu.vn

Phan Công Minh
Trường Đại học Công nghệ Thông tin
ĐHQG TP.HCM
Khoa Khoa Học Máy tính
22520884@gm.uit.edu.vn

Tóm tắt

Bóng đá là một trong những môn thể thao nổi tiếng và được yêu thích nhất trên thế giới, vì thế việc theo dõi diễn biến các trận đấu và dự đoán kết quả từ chúng là mối quan tâm chung đối với nhiều đối tượng như: người hâm mộ, ban huấn luyện, truyền thông, hay thậm chí là người chơi cá cược. Điều này thúc đẩy sự phát triển không ngừng của các hệ thống hỗ trợ dự đoán kết quả trận đấu bóng đá. Trong báo cáo này, nhóm đề xuất quá trình phân tích dữ liệu cùng với áp dụng phương pháp máy học để dự đoán kết quả trận đấu bóng đá, sử dụng dữ liệu lịch sử từ các trận đấu đã xảy ra trước đó. Kết quả thử nghiệm được đúc kết từ nhiều mô hình, cho thấy hiệu quả dự đoán đáng khích lệ.

1 Giới thiệu

Bóng đá - môn thể thao vua - là môn thể thao được đông đảo người hâm mộ trên toàn thế giới yêu thích, quan tâm và theo dõi. Mỗi trận đấu không chỉ mang lại cảm xúc mạnh mẽ mà còn chứa đựng một kho dữ liệu phong phú, có giá trị trong việc phân tích và khai thác. Việc dự đoán kết quả bóng đá từ dữ liệu quá khứ không chỉ giúp người xem có cái nhìn khách quan về trận đấu mà còn là một ứng dụng thực tế, hỗ trợ ra quyết định trong lĩnh vực phân tích dữ liệu ở chủ đề thể thao. Việc mô hình hóa một hệ thống dự đoán các trận đấu bóng đá không chỉ được quan tâm trong giới học thuật mà còn có ý nghĩa to lớn về mặt giá trị kinh tế. Một bài báo từ BBC ước tính giá trị của thị trường cá cược bóng đá rơi vào khoảng 700 tỷ đến 1000 tỷ đô la (https://www.bbc.com/sport/football/24354124). Tuy vậy, "tính khó đoán" của các trận bóng đá làm cho việc cá cược này trở nên rất khó khăn nếu không dựa trên bất kỳ phân tích sâu sắc nào về thông tin các trận đấu. Bởi kết quả một trận bóng đá dựa trên rất nhiều yếu tố, có thể là trình độ, phong độ tổng quan của cả đội tại một thời điểm nhất định, hay kĩ năng cá nhân, tâm lý của từng cầu thủ, và đôi khi còn ảnh hưởng bởi biến số như may mắn trong một số tình huống. Do đó, các hệ thống có thể thống kê và học hỏi được dữ liệu đã xảy ra, từ đó đưa ra kết quả cho một trận đấu, có thể là kèm theo cả độ tin cậy tương ứng, sẽ giúp đưa ra các dự đoán đáng giá hơn (Rodrigues and Pinto, 2022). Đề tài của nhóm tập trung vào phân tích, trích xuất dữ liệu, tạo ra thêm các đặc trưng thống kê mang ý nghĩa quan trọng đối với kết quả trận đấu, từ đó cung cấp cho các mô hình máy học phổ biến học hỏi và ra quyết định. Bài toán dự đoán trận đấu bóng đá là bài toán phân lớp đa nhãn, với kết quả sẽ rơi vào một trong ba lớp: (Đội đang xét thắng, Đội đối thủ thắng, Hai đội hòa nhau).

Phát biểu bài toán:

- Input:

+ Dữ liệu lịch sử về các trận đấu bóng đá của các đội.

+ Tên của hai đội cần dự đoán, kèm thông tin về ngày diễn ra và đội nào thi đấu trên sân nhà.

- Output:

+ Kết quả của đội chủ nhà với một trong ba khả năng: Thắng (W), Hòa (D) và Thua (L). Mỗi kết quả được gán một giá trị xác suất phản ánh mức độ tin cậy của dự đoán.

Phần còn lại của bài viết được tổ chức như sau. Phần 2 tóm tắt các nghiên cứu liên quan. Ở Phần 3, dữ liệu thu thập, phân tích và tiền xử lý để thu thập các thông tin quan trọng. Một số mô hình máy học được áp dụng và đánh giá hiệu suất tại Phần 4 và Phần 5 của bài viết. Cuối cùng, Phần 6 đưa ra các kết luận tổng quan cùng các hướng phát triển.

2 Nghiên cứu liên quan

Đề tài dự đoán kết quả trận bóng đá thu hút rất nhiều nghiên cứu, tài liệu liên quan. Các nghiên cứu có thể được chia thành hai nhóm chính: dự đoán tỉ số bàn thắng và dự đoán kết quả trận đấu. Bài viết của nhóm tập trung vào nhóm mục tiêu thứ hai.

(Min et al., 2008) đề xuất một phương pháp kết hợp suy luận luật và mạng Bayes để dự đoán kết quả các trận bóng đá. Nghiên cứu sử dụng cả các đặc trưng phi tỉ số: thể lực, tinh thần, chiến thuật,.... Thực hiện mô phỏng trận đấu theo thời gian, mô phỏng luôn cả hiệp phụ và loạt luân lưu. Kết quả cuối cùng thu được được đánh giá là tốt và vượt trội so với các baseline mà họ so sánh, trong bối cảnh bài toán dự đoán World Cup 2022 - dữ liệu khó đoán, biến động lớn). Đây là một nghiên cứu mang tính sáng tạo cao so với số đông các framework dự đoán cùng đề tài.

Trong khi đó, tài liệu từ (Baboota and Kaur, 2019) kết hợp các đặc trưng cơ bản của trận đấu thành các đặc trưng mới cung cấp nhiều thông tin hữu ích cho việc phân tích. Một số đặc trưng mới được đề xuất như: Streak - đánh giá hiệu suất thắng thua trong thời gian gần của đội bóng, hay Form - cũng là chỉ số đánh giá hiệu suất đội bóng, nhưng sẽ tăng giảm lượng khác nhau dựa trên việc đối đầu các đội mạnh hay yếu hơn bản thân. Nghiên cứu cũng áp dụng nhiều mô hình dự đoán và thu được kết quả đáng khích lệ khi so sánh với các kết quả từ một số nền tảng cá cược như Bet365 và Pinnacle Sports. Dù vậy, bài nghiên cứu vẫn hứa hẹn có thể cải thiện hiệu suất cho các hệ thống đánh giá dựa trên sự kết hợp thêm từ một số đặc trưng như thông tin chấn thương, cầu thủ chủ chốt,...

Mặc dù sự khó đoán trong kết quả các trận đấu được biết đến rộng rãi, đôi khi một số trường hợp thật sự bất ngờ vẫn có thể diễn ra, như sự kiện về chức vô địch của đội tuyển Leicester City tại giải Premier League vào mùa giải 2015/2016. Một phân tích sâu sắc từ (Ruiz et al., 2017) đã chỉ ra rằng một đội bóng có thành tích lịch sử không quá nổi trội vẫn có thể giành được chức vô địch, dựa trên một số yếu tố như sự xuất sắc từ thủ môn, hay khả năng chuyển hóa bàn thắng từ các tình huống phản công là vô cùng hiệu quả.

3 Phân tích và tiền xử lý dữ liệu

3.1 Mô tả dữ liệu

Dữ liệu được sử dụng cho đề tài được tải từ trang web (https://fbref.com/) - trang chuyên cung cấp thông tin liên quan đến các giải đấu thể thao khá đầy đủ và uy tín, bên cạnh đó thì giao diện của trang web vô cùng thân thiện, dễ tiếp cận và thực hiện các thao tác thu thập dữ liệu. Dữ liệu xét trên giải đấu Premier League (Ngoại hạng Anh) trong vòng 7 mùa đấu liên tục (từ mùa 2017/18 đến hết mùa 2023/24), và bao gồm các đội bóng xuất hiện liên tục trong các mùa giải đó (không bị rớt hạng).

Ban đầu, nhóm sử dụng lệnh có sẵn từ thư viện request trong python để thu thập dữ liệu. Tuy nhiên, do vấn đề bị chặn IP khi thu thập nhiều dữ liệu từ trang web, nhóm đã thử sử dụng một hàm tự định nghĩa khác là safe_request_get thay vì hàm get mặc định từ thư viện, với sự bổ sung thêm một số chức năng như: thêm random delay, thêm các headers khác nhau như 'User-Agent'; 'Accept-Language'; 'Accept-Encoding',...

Dù đã thêm một số cải tiến, trang web vẫn tiếp tục chặn và từ chối yêu cầu truy cập dữ liệu. Sau cùng, nhóm sử dụng một công cụ cung cấp khả năng thu thập dữ liệu từ web miễn phí, không bị chặn là ScrapingBee (https://www.scrapingbee.com/. ScrapingBee sẽ cung cấp một API key để lấy dữ liệu từ web, và sẽ tự động lo liệu việc quay vòng proxy, vượt qua các biện pháp bảo vệ chống bot (như CAPTCHA) để đảm bảo rằng có thể thu thập dữ liệu mà không gặp các hạn chế này.

3.2 Khám phá dữ liệu

Dữ liệu chứa tổng cộng 1539 dòng cùng 41 cột thuộc tính. Một số thuộc tính có thể kể đến như: start_time - thời gian trận đấu diễn ra, home_team và opponent - tên đội đang xét và đội đối thủ, goals_for, tổng số bàn thắng đội đang xét ghi được, shots - tổng số cú sút, xg - số bàn thắng kỳ vọng, tackles - tổng số lần truy cản, errors - số lỗi dẫn đến cơ hội ghi bàn cho đối thủ,...

Có tổng 8 cột dạng object và 33 cột dạng số học. Các cột dạng object chủ yếu là những đặc trưng phân loại, trong khi các cột số học chứa dữ liệu số liên quan đến thống kê trận đấu. Bộ dữ liệu được thu thập ngay từ đầu không tồn tại giá trị thiếu.

Khi sử dụng boxplot để kiểm tra giá trị ngoại lệ, nhóm nhận thấy các đặc trưng như goals_for và goals_against chứa giá trị ngoại lệ rất lớn, xảy ra do tỉ số của các trận đấu cao, khi hai đội tồn tại sự chênh lệch lớn về trình độ, chiến thuật hoặc phong độ [Figure 1].

Ngoài ra, ứng với mỗi đội tuyển, nhóm đã thống kê số lượng trận cho từng kết quả của các nhãn dự đoán, từ đó biết được những đội có khả năng chiến thắng cao hơn [Figure 2]

3.3 Tiền xử lý dữ liệu

Bộ dữ liệu sẽ được xử lý và đánh giá dựa trên chuẩn **Medalion Architecture**

Bronze Layer:
Là dữ liệu thô được thu thập trong quá trình crawling từ trang web.

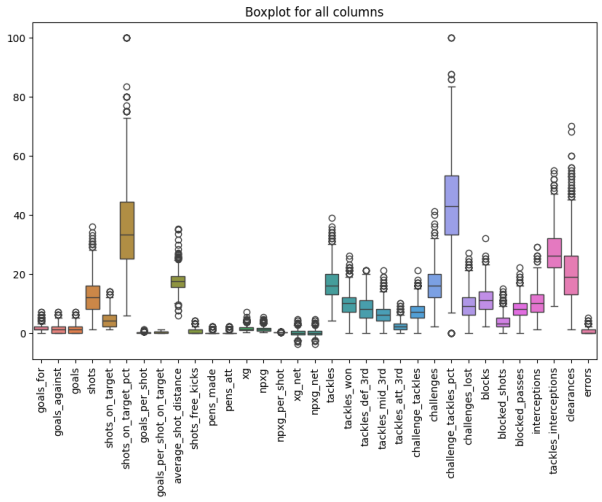

Figure 1: Biểu đồ giá trị ngoại lệ cho từng đặc trưng

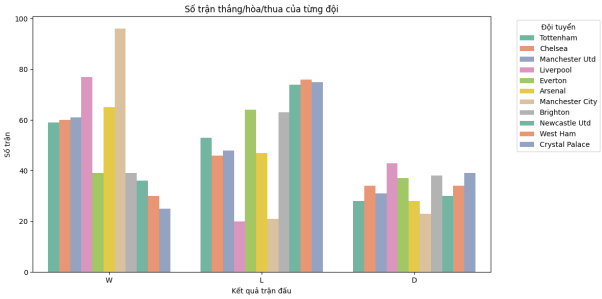

Figure 2: Biểu đồ thống kê số trận theo kết quả cho từng đội tuyển

Các thao tác kiểm tra dữ liệu bao gồm:
- In ra thông tin tổng quan về bộ dữ liệu, bao gồm số lượng bản ghi và kiểu dữ liệu của từng cột.
- Kiểm tra số lượng giá trị thiếu ở từng cột.

Silver Layer:
Các thao tác xử lý mà nhóm đã thực hiện để đánh giá bộ dữ liệu đạt chuẩn silver:
- Ánh xạ các đặc trưng không phải số học:
+ **venue**: ánh xạ các giá trị Home và Away thành các nhãn 0 và 1.
+ **result**: ánh xạ các kết quả L, W, D thành các nhãn 0, 1 và 2.
+ **home_team, opponent_team**: sử dụng từ điển ánh xạ để chuyển đổi tên các đội bóng thành các nhãn định nghĩa trước.
- Chuyển đổi dữ liệu cột **Date**: chuyển từ kiểu object thành kiểu DateTime để thuận tiện cho việc phân chia các tập huấn luyện và kiểm tra.
- Chuyển đổi các cột số lượng có kiểu dữ liệu là số thực thành số nguyên để phù hợp về mặt logic. Ví dụ như số cú sút, số lỗi,...
- Loại bỏ cột gốc sau khi ánh xạ và các cột không mang nhiều ý nghĩa cho dự đoán nhằm tránh dư thừa dữ liệu.

Gold Layer:
Tạo thêm các đặc trưng bằng kỹ thuật Rolling Window:
- Trong phần này, nhóm thực hiện tạo thêm các đặc trưng mới cho bộ dữ liệu bằng phương pháp **Rolling Window**. Các đặc trưng này có thể giúp mô hình có thêm thông tin về các trận đấu trước đó của đội đang xét, từ đó nâng cao khả năng dự đoán kết quả trận đấu trong tương lai.
+ Hàm create_rolling_feature() sẽ thực hiện các bước sau:

Sắp xếp dữ liệu theo đội đang xét và theo ngày nhằm đảm bảo áp dụng chính xác thời gian của mỗi đội bóng.

Trong số các đặc trưng gốc, nhóm thực hiện lựa chọn 4 đặc trưng để tạo ra các đặc trưng Rolling là **xg, npxg, shots, tackles**, đây là các đặc trưng mà nhóm cho rằng quan trọng đối với một trận đấu. Mỗi đặc trưng sẽ được tính toán bằng cách lấy giá trị trung bình trong một cửa sổ thời gian nhất định với kích thước được lựa chọn là 3.

Áp dụng Rolling Window với .shift(1) để tính các giá trị.
- Tiếp theo, nhóm thực hiện tạo các đặc trưng liên quan đến kết quả và thông số đối đầu trực tiếp giữa các cặp đội bóng bằng cách sử dụng Rolling Window. Mục tiêu của đặc trưng này là cung cấp cho mô hình thêm thông tin về lịch sử đối đầu gần đây, từ đó đánh giá khả năng chiến thắng, ghi bàn và bị thủng lưới của một đội bóng trong những lần gặp cụ thể, có phân biệt bối cảnh sân nhà và sân khách. Thuật toán được xây dựng để xử lý từng cặp đội theo hai chiều độc lập, khi đội A thi đấu trên sân nhà, đội B là đội khách và ngược lại.
+ Với mỗi chiều đối đầu, các đặc trưng được tính toán:

h2h_avg_result: trung bình kết quả đối đầu gần nhất, được tính trên cột target.

h2h_avg_goals: trung bình số bàn thắng ghi được trong các trận đối đầu gần nhất, được tính trên cột goals.

h2h_avg_goals_against: trung bình số bàn thua phải nhân trong các trận đối đầu gần nhất, được tính trên cột goals_against.
+ Trước hết, sắp xếp các dữ liệu theo thời gian. Sau đó các giá trị trên được tính thông qua Rolling Window với kích thước là 3 với .shift(1).
Sau khi tính toán các giá trị rolling, ba dòng đầu (tương ứng với ba trận đầu tiên trong mỗi cặp đội) sẽ có giá trị là NaN vì không đủ dữ liệu từ các trận

đấu trước đó để tính toán trung bình. Nhóm quyết định điền vào các ô NaN này bằng giá trị 0 , đại diện cho việc chưa có dữ liệu lịch sử cho các trận đấu này.

Dựa trên tham khảo từ (Baboota and Kaur, 2019), nhóm tạo thêm đặc trưng **Form** - cung cấp thông tin về phong độ thi đấu tại thời điểm đang xét của các đội bóng dựa trên các trận thi đấu trước đó, thể hiện sự nắm bắt yếu tố thời gian mạnh mẽ. Tuy nhiên, điểm khác biệt chính giữa Form và các đặc trưng kể trên nằm ở việc nó đo lường được hiệu suất thi đấu của một đội dựa trên sức mạnh tương ứng của đối thủ mà đội đó gặp. Cụ thể, công thức tính toán của Form đảm bảo rằng một lượng giá trị cộng thêm lớn hơn sẽ được cung cấp nếu một đội bóng yếu hơn (dựa trên giá trị Form hiện tại) chiến thắng trước một đội bóng mạnh, và ngược lại. Trong trường hợp hai đội hòa nhau, giá trị Form của đội yếu sẽ được tăng lên, ngược lại thì giá trị Form của đội mạnh sẽ bị giảm xuống.

- Giá trị Form của mỗi đội sẽ được khởi tạo là 1 ở đầu mỗi mùa giải, và được cập nhật sau mỗi trận đấu dựa trên kết quả của trận.

+ Giả sử đội α đánh bại đội β, khi đó giá trị Form (ξ) cho mỗi đội tại trận thứ j sẽ được cập nhật như sau:

$$\xi_j^{\alpha} = \xi_{(j-1)}^{\alpha} + \gamma \xi_{(j-1)}^{\beta}$$
$$\xi_j^{\beta} = \xi_{(j-1)}^{\beta} - \gamma \xi_{(j-1)}^{\beta}$$

+ Trong trường hợp kết quả là hòa, giá trị Form (ξ) cho mỗi đội tại trận thứ j sẽ được cập nhật như sau:

$$\xi_j^{\alpha} = \xi_{(j-1)}^{\alpha} - \gamma(\xi_{(j-1)}^{\alpha} - \xi_{(j-1)}^{\beta})$$
$$\xi_j^{\beta} = \xi_{(j-1)}^{\beta} - \gamma(\xi_{(j-1)}^{\beta} - \xi_{(j-1)}^{\alpha})$$

- Trong công thức trên, γ là tỉ lệ chiếm đoạt (stealing fraction) $\in (0, 1)$. Sau quá trình kiểm thử và đánh giá trên dữ liệu, nhóm tìm ra được giá trị tối ưu cho γ là 0.5.

Cuối cùng là loại bỏ các cột gây rò rỉ thông tin về kết quả trận đấu như **goals, goals_for, goals_against** để đảm bảo mô hình không thể gian lận khi dự đoán kết quả. Việc giữ lại các cột nêu trên khiến mô hình gần như biết trước được toàn bộ kết quả của trận đấu, giảm độ tin cậy của dự đoán và không phản ánh đúng bản chất của bài toán.

4 Phương pháp

4.1 Chia tập dữ liệu

Việc chia dữ liệu sẽ dựa trên thuộc tính 'Date' để phân chia làm 2 tập huấn luyện và kiểm tra.

Dữ liệu về các trận đấu thu thập được bắt đầu từ ngày 12-08-2017 và kết thúc ngày 19-05-2024.

Nhóm quyết định sử dụng 6 mùa giải (từ 2017/2018 đến 2022/2023) dùng cho huấn luyện, còn lại mùa giải 2023/2024 sẽ dùng cho kiểm thử, với tỉ lệ là 85-15. Với lí do là bộ dữ liệu kiểm thử nên chứa toàn bộ trận đấu của mùa giải, bởi vì xuyên suốt từ đầu tới cuối mùa thì phong độ của các đội bóng có thể thay đổi đa dạng (Rodrigues and Pinto, 2022).

4.2 Mô hình sử dụng

Logistic Regression: là mô hình hồi quy tuyến tính dùng để dự đoán xác suất của các sự kiện. Mô hình này sử dụng một hàm Logistic để biến đổi đầu ra tuyến tính thành xác suất nằm trong khoảng từ 0 đến 1. Đây là mô hình đơn giản, dễ hiểu và nhanh chóng, đặc biệt phù hợp với các bài toán có mối quan hệ tuyến tính giữa các đặc trưng đầu vào và kết quả. Mặc dù vậy, Logistic Regression không phù hợp với dữ liệu phi tuyến tính và có thể bị ảnh hưởng mạnh mẽ bởi các Outliers.

Random Forest: là một thuận toán Ensemble, sử dụng nhiều cây quyết định (Decision Tree) để đưa ra dự đoán, với mỗi cây được huấn luyện trên một phần mẫu ngẫu nhiên của dữ liệu. Random Forest có khả năng xử lý tốt với dữ liệu phi tuyến và ít bị overfitting nhờ vào việc kết hợp nhiều mô hình học riêng lẻ. Nó cũng có thể xử lý các giá trị thiếu một cách hiệu quả. Tuy nhiên, mặc dù Random Forest có khả năng dự đoán mạnh mẽ, nó vẫn là một mô hình khó giải thích do tính chất "black-box" của các cây quyết định kết hợp.

XGBoost (Extreme Gradient Boosting): là một thuật toán boosting mạnh mẽ được xây dựng trên nền tảng của gradient boosting. Nó kết hợp nhiều mô hình yếu (weak learners) để tạo ra một mô hình mạnh mẽ hơn, bằng cách sử dụng thuật toán gradient descent để tối ưu hóa các mô hình cây quyết định. XGBoost đặc biệt mạnh mẽ trong việc xử lý các bộ dữ liệu phức tạp và không đồng đều, đồng thời có khả năng huấn luyện nhanh và hiệu suất cao. Tuy nhiên, XGBoost yêu cầu một lượng tài nguyên tính toán lớn và có thể dễ dàng bị overfitting nếu không điều chỉnh đúng các tham số.

4.3 Thực nghiệm

Nhóm thực hiện các kịch bản để có cái nhìn tổng quan về hiệu quả mô hình và các đặc trưng như sau:

- Kịch bản 1: Sử dụng các tham số mặc định, đặc trưng gốc kết hợp với đặc trưng từ phương pháp

Rolling Windows.
- Kịch bản 2: Sử dụng các tham số tối ưu với GridSearchCV, đặc trưng gốc kết hợp với đặc trưng từ phương pháp Rolling Windows.
- Kịch bản 3: Sử dụng các tham số mặc định, đặc trưng gốc kết hợp với đặc trưng từ phương pháp Rolling Windows và đặc trưng Form
- Kịch bản 4: Sử dụng các tham số tối ưu với GridSearchCV, đặc trưng gốc kết hợp với đặc trưng từ phương pháp Rolling Windows và đặc trưng Form.

4.3.1 Kịch bản 1

Khởi tạo các tham số mặc định cho ba mô hình lần lượt như sau:
- Logistic Regression (multi_class = 'multinomial', solver = 'lbfgs')
- Random Forest ()
- XGBoostClassifier (objective = 'multi:softprob', num_class = 3)

Kết quả thu được như sau:

Table 1: Kết quả Logistic Regression

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	49	15	19
Actual win	5	71	7
Actual draw	10	19	25
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.77	0.59	0.67
Win	0.68	0.86	0.76
Draw	0.49	0.46	0.48
Accuracy			0.66

Table 2: Kết quả Random Forest

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	54	20	9
Actual win	6	75	2
Actual draw	17	25	12
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.66	0.64	0.68
Win	0.59	0.84	0.55
Draw	0.41	0.17	0.24
Accuracy			0.60

Table 3: Kết quả XGBoost

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	49	17	17
Actual win	6	69	8
Actual draw	15	19	29
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.73	0.58	0.64
Win	0.65	0.81	0.72
Draw	0.47	0.44	0.46
Accuracy			0.63

4.3.2 Kịch bản 2

GridSearchCV là kỹ thuật dùng để tìm các giá trị tham số tối ưu từ một tập hợp các tham số đã cho dưới dạng lưới (Grid). Về bản chất, đây là một kỹ thuật kiểm định chéo (cross-validation). Cả mô hình và các tham số cần được cung cấp đầu vào. Sau đó tìm ra bộ tham số tốt nhất, mô hình sẽ được sử dụng để thực hiện dự đoán.

Danh sách giá trị các tham số thử nghiệm mà nhóm sử dụng để đưa vào GridSearchCV:
- Logistic Regression:
+ 'penalty': 'l2'
+ 'C': [0.01, 0.1, 1, 10]
+ 'solver': 'lbfgs'
+ 'multi_class': 'multinomial'
+ 'max_iter': 1000
- Random Forest:
+ 'n_estimators': [100, 300, 500]
+ 'max_depth': [None, 10, 20]
+ 'min_samples_split': [2, 5, 10]
+ 'min_samples_leaf': [1, 2, 4]
- XGBoost:
+ 'n_estimators': [100, 300, 500]
+ 'max_depth': [3, 6, 10]
+ 'learning_rate': [0.01, 0.1, 0.2]
+ 'subsample': [0.8, 1]
+ 'colsample_bytree': [0.8, 1]

Tham số tốt nhất của từng mô hình và điểm tương ứng trên bộ dữ liệu huấn luyện:
- Logistic Regression: {'C': 0.1, 'max_iter': 1000, 'multi_class': 'multinomial', 'penalty': 'l2', 'solver': 'lbfgs'} - Điểm: 0.6717
- Random Forest: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 500} - Điểm: 0.6444
- XGBoost: {'colsample_bytree': 1, 'learning_rate': 0.2, 'max_depth': 10, 'n_estimators': 500, 'subsample': 0.8} - Điểm: 0.6535
Kết quả thu được như sau:

Table 4: Kết quả Logistic Regression

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	58	11	14
Actual win	5	71	7
Actual draw	11	19	24
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.78	0.70	0.74
Win	0.70	0.86	0.77
Draw	0.53	0.44	0.48
Accuracy			0.70

Table 5: Kết quả Random Forest

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	54	20	9
Actual win	6	75	2
Actual draw	17	25	12
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.70	0.65	0.68
Win	0.62	0.90	0.74
Draw	0.52	0.22	0.31
Accuracy			0.64

Table 6: Kết quả XGBoost

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	49	17	17
Actual win	6	69	8
Actual draw	15	19	20
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.70	0.59	0.64
Win	0.66	0.83	0.73
Draw	0.44	0.37	0.40
Accuracy			0.63

4.3.3 Kịch bản 3

Kết quả thu được như sau:

Table 7: Kết quả Logistic Regression

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	70	3	10
Actual win	0	78	5
Actual draw	4	13	37
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.95	0.84	0.89
Win	0.83	0.94	0.88
Draw	0.71	0.0.69	0.70
Accuracy			0.84

Table 8: Kết quả Random Forest

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	70	3	10
Actual win	0	78	5
Actual draw	8	16	30
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.90	0.84	0.87
Win	0.80	0.94	0.87
Draw	0.67	0.56	0.61
Accuracy			0.81

Table 9: Kết quả XGBoost

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	74	1	8
Actual win	0	79	4
Actual draw	3	3	48
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.96	0.89	0.93
Win	0.95	0.95	0.95
Draw	0.80	0.89	0.84
Accuracy			0.91

4.3.4 Kịch bản 4

Tham số tốt nhất của từng mô hình và điểm tương ứng trên bộ dữ liệu huấn luyện:
- Logistic Regression: {'C': 10, 'max_iter': 1000, 'multi_class': 'multinomial', 'penalty': 'l2', 'solver': 'lbfgs'} - Điểm: 0.9447
- Random Forest: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 500} - Điểm: 0.8416
- XGBoost: {'colsample_bytree': 0.8, 'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 500, 'subsample': 0.8} - Điểm: 0.356

Kết quả thu được như sau:

Table 10: Kết quả Logistic Regression

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	72	0	11
Actual win	0	78	5
Actual draw	1	1	52
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.99	0.87	0.82
Win	0.99	0.94	0.96
Draw	0.76	0.96	0.85
Accuracy			0.92

Table 11: Kết quả Random Forest

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	72	5	6
Actual win	0	77	6
Actual draw	8	14	32
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.90	0.87	0.88
Win	0.80	0.93	0.86
Draw	0.73	0.59	0.65
Accuracy			0.82

Table 12: Kết quả XGBoost

(a) Confusion matrix	Pred win	Pred loss	Pred draw
Actual loss	75	2	6
Actual win	0	79	4
Actual draw	2	8	44
(b) Precision–recall table			
	Precision	Recall	F1-score
Lose	0.97	0.90	0.94
Win	0.89	0.95	0.92
Draw	0.81	0.81	0.91
Accuracy			0.90

5 Đánh giá mô hình

5.1 Đánh giá các mô hình trên các kịch bản

Việc lựa chọn và triển khai ba mô hình Logistic Regression, Random Forest và XGBoost nhằm mục đích đánh giá, so sánh hiệu năng của từng mô hình khi áp dụng vào bài toán dự đoán kết quả trận đấu trên dữ liệu chuỗi thời gian (Time-Series).
Dựa trên kết quả thực nghiệm với bốn kịch bản đã được trình bày trong phần trước, nhóm rút ra một số nhận định như sau:

Sử dụng đặc trưng gốc và Rolling Windows:
- Độ chính xác của ba mô hình dao động từ 60% đến 70%.
- Trong đó Logistic Regression với các tham số tối ưu cho kết quả tốt nhất, đạt độ chính xác 70%.
- Mô hình này cũng thể hiện hiệu suất dự đoán tốt ở cả các nhãn phổ biến là (Win, Lose) và nhãn ít xuất hiện hơn (Draw) với Precision lần lượt là 0.78, 0.70, 0.53.
- Điều này cho thấy Logistic Regression có khả năng dự đoán tương đối tốt với cả nhãn có tần suất thấp, tuy nhiên precision là 0.53 cho nhãn Draw vẫn là một con số chưa thật sự ấn tượng, và hiệu suất tổng thể của ba mô hình còn cần được cải thiện.

Kết hợp thêm đặc trưng Form:
- Khi bổ sung đặc trưng Form, độ chính xác của các mô hình được cải thiện rõ rệt, dao động từ 81% đến 91%.
- Hai mô hình có hiệu suất cao nhất là:
+ Logistic Regression đạt 92%.
+ XGBoost đạt 90%.
+ Trong khi đó Random Forest đạt 82%.
- Cả ba mô hình đều có hiệu suất cao trên từng nhãn, phản ánh sự đóng góp rõ rệt của đặc trưng Form vào khả năng dự đoán.

Tuy nhiên, với mục tiêu trọng tâm là đảm bảo độ tin cậy trong dự đoán, nhóm lựa chọn precision làm tiêu chí chính để đánh giá mô hình.
- Dù Logistic Regression có độ chính xác tổng thể cao hơn XGBoost (92% so với 90%), nhưng lại cho thấy xu hướng nghiêng mạnh về các nhãn phổ biến. Cụ thể, precision của Logistic Regression đối với nhãn **Win** và **Lose** đều đạt **0.99**, trong khi với nhãn **Draw** chỉ đạt **0.76**.
- Ngược lại, **XGBoost** có precision đồng đều và cao trên cả ba nhãn, lần lượt là **0.97 (Lose)**, **0.89 (Win)** và **0.81 (Draw)**.
- Do đó, nhóm quyết định lựa chọn **XGBoost** là mô hình cuối cùng, vì mô hình này mang lại sự **cân bằng giữa độ chính xác và độ tin cậy**, đặc biệt với nhãn ít xuất hiện hơn như **Draw**.

5.2 Đánh giá trên Time Series Split Validation

Time Series Split (TSS) là phần mở rộng của kỹ thuật K-Fold Cross Validation (KFCV), được thiết kế riêng và phù hợp với dữ liệu có dạng chuỗi thời gian (Time Series). Thay vì xáo trộn ngẫu nhiên dữ liệu như cách KFCV thực hiện thì TSS vẫn duy trì thứ tự thời gian. Tập dữ liệu được chia thành nhiều phần liên tiếp, trong đó mỗi phần sử dụng dữ liệu quá khứ (có giá trị thời gian nhỏ hơn) để huấn luyện và dữ liệu tương lai để kiểm tra (có giá trị thời gian lớn hơn). Điều này mô phỏng đúng với bài toán mà nhóm đang thực hiện, vì các trận đấu sẽ được dự đoán trên lịch sử trước đó.

Sau khi thực hiện GridSearchCV để tìm ra các tham số tối ưu cho từng mô hình ở phần kịch bản, nhóm thực hiện bổ sung bằng cách đánh giá từng mô hình đã nêu trên TSS với **n_splits=10** để đánh giá mức độ ổn định của các mô hình.

Kết quả về độ chính xác trung bình của các mô hình Logistic Regression, Random Forest và XGBoost lần lượt là **90.43%, 81.22%, 90.36%**. Điều này cho thấy hai mô hình Logistic Regression và XGBoost có độ ổn định tốt hơn trên bộ dữ liệu so

với Random Forest. Tuy nhiên dựa trên kết quả đã nêu ở phần kịch bản, nhóm vẫn quyết định lựa chọn mô hình XGBoost để thực hiện dự đoán bởi độ ổn định trên các nhãn.

6 Mở rộng và kết luận

6.1 Xây dựng ứng dụng bằng Streamlit

Sau khi đã thực hiện huấn luyện và lựa chọn mô hình phù hợp, nhóm tiến hành lưu dữ liệu cuối cùng (bao gồm cả đặc trưng Rolling Windows và đặc trưng Form) cùng với mô hình XGBoost đã lựa chọn.

Tiếp theo, nhóm tiến hành thực hiện xây dựng một ứng dụng thử nghiệm nhỏ với thư viện Streamlit trong Python. Streamlit là một framework được thiết kế nhằm hỗ trợ xây dựng ứng dụng web tương tác chỉ sử dụng ngôn ngữ Python mà không cần các kiến thức chuyên sâu về web.

Nhóm xây dựng hàm predict_future_match cho phép dự đoán kết quả của một trận đấu bóng đá sắp diễn ra dựa trên các yếu tố: Tên đội chủ nhà, tên đội khách, ngày diễn ra trận đấu, dữ liệu lịch sử, mô hình huấn luyện, bảng ánh xạ các đội tuyển và tham số window (dùng để lấy số trận gần nhất cho phép tính trung bình). Về ý tưởng, hàm hoạt động theo các bước:
- Mã hóa các đội đang xét, đội đối thủ.
- Lọc dữ liệu, giữ lại các trận đấu đã diễn ra trước ngày cần dự đoán.
- Tính các đặc trưng đầu vào:
+ Các đặc trưng Rolling về các chỉ số tấn công, phòng thủ được tính từ n (window) trận gần nhất của đội đang xét, các chỉ số về hệ số đối đầu được tính từ n trận gần nhất của cặp đấu của đội đang xét và đội đối thủ.
+ Các đặc trưng còn lại lấy giá trị trung bình từ n trận gần nhất.
- Lưu các đặc trưng và giá trị tương ứng vào một DataFrame và thực hiện dự đoán, mô hình sẽ đưa ra nhãn dự đoán và xác suất tương ứng, kết quả trả về gồm nhãn dự đoán và xác suất theo từng lớp.

Hàm predict_future_match đóng vai trò như một API Logic cho hệ thống dự đoán. Nhóm đã thực hiện việc tích hợp vào giao diện Streamlit để tạo thành một ứng dụng đơn giản, cho phép người dùng chọn đội, ngày thi đấu và nhận được kết quả một cách trực quan.

6.2 Kết luận

- Kết quả đạt được:
+ Nhóm đã hoàn thành mục tiêu chính của đồ án là xây dựng một mô hình học máy có khả năng dự đoán kết quả trận đấu bóng đá dựa trên dữ liệu lịch sử được thu thập từ web.
+ Ngoài ra, nhóm đã ứng dụng mô hình vào thực tế thông qua một giao diện thử nghiệm, cho phép người dùng nhập dữ liệu và nhận kết quả dự đoán.
- Hạn chế:
+ Quy mô dữ liệu còn hạn chế: Dữ liệu thu thập chủ yếu từ một số mùa giải gần đây của giải Ngoại Hạng Anh và chỉ bao gồm các đội thường xuyên trụ hạng. Điều này khiến mô hình khó khái quát với các đội yếu hơn hoặc các giải đấu khác.
+ Đặc trưng ban đầu còn đơn giản: Các đặc trưng được sử dụng chủ yếu là thống kê tổng quát từ trận đấu, chưa có các yếu tố chuyên sâu như thông tin về cầu thủ, huấn luyện viên, đội hình ra sân,...
+ Các đặc trưng mới còn cơ bản: Những đặc trưng được tạo ra chủ yếu dựa trên trung bình cộng theo thời gian (rolling window) và hiệu số đối đầu, chưa tận dụng được các kỹ thuật phức tạp hơn như biểu diễn thời gian, tương tác giữa các đặc trưng,...
- Hướng phát triển:
+ Mở rộng tập dữ liệu: Thu thập thêm dữ liệu từ nhiều giải đấu quốc nội và quốc tế, bao gồm cả các giải lớn như La Liga, Bundesliga, Serie A,... để cải thiện khả năng tổng quát của mô hình.
+ Tăng độ sâu đặc trưng: Bổ sung thêm các chỉ số chuyên sâu liên quan đến cầu thủ (phong độ, thể lực), huấn luyện viên (chiến thuật), đội hình thi đấu,... để nâng cao chất lượng dữ liệu đầu vào.
+ Thử nghiệm mô hình nâng cao: Áp dụng các mô hình học sâu tiên tiến như Neural Networks, hoặc Transformer để cải thiện độ chính xác và khả năng học phi tuyến tính của mô hình.
+ Phát triển ứng dụng hoàn chỉnh: Hoàn thiện giao diện người dùng, bổ sung các chức năng hỗ trợ như xem lịch sử dự đoán, so sánh với kết quả thực tế và triển khai trên nền tảng cloud.

Tài liệu tham khảo

Rahul Baboota and Harleen Kaur. 2019. Predictive analysis and modelling football results using machine learning approach for english premier league. *International Journal of Forecasting*, 35(2):741–755.

Byungho Min, Jinhyuck Kim, Chongyoun Choe, Hyeonsang Eom, and RI McKay. 2008. A compound framework for sports prediction: The case study of football. *Knowledge-Based Systems*, 21(7):551–562.

Fátima Rodrigues and Ângelo Pinto. 2022. Prediction

of football match results with machine learning. *Procedia Computer Science*, 204:463–470.

Hector Ruiz, Paul Power, Xinyu Wei, and Patrick Lucey. 2017. " the leicester city fairytale?" utilizing new soccer analytics tools to compare performance in the 15/16 & 16/17 epl seasons. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1991–2000.