

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN**



Báo cáo nghiên cứu môn học Chuyên Đề Công Nghệ

**Dự đoán kết quả trận đấu bóng đá cho giải đấu
English Premier League - EPL**

Sinh viên: Nguyễn Đắc Phong, 21021525, K66-CA-CLC2

Đào Xuân Nghĩa, 21020472, K66-CA-CLC2

Nguyễn Minh Quân, 21021534, K66-CA-CLC2

Lê Bùi Sơn, 21020662, K66-CA-CLC2

Phạm Ngọc Thạch, 21020113, K66-CA-CLC2

Giảng viên: PGS.TS. Nguyễn Việt Anh

Hà Nội, 2024

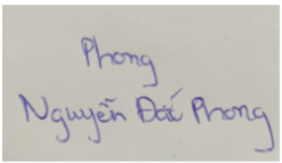
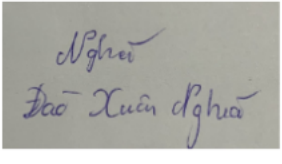
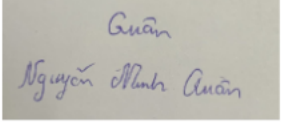
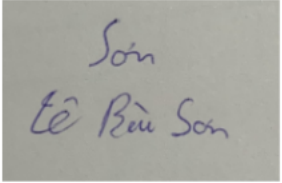
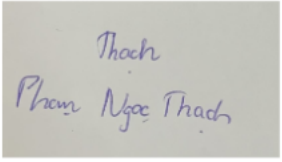
Mục lục

1	Introduction	4
1.1	Motivation	4
1.2	Problem statements	4
1.3	Research questions	5
2	Materials and Methods	5
2.1	Data Collection	5
2.2	Data Preparing	7
2.3	Data Preprocessing	8
2.4	Data Visualization	9
2.5	Method	10
2.5.1	Random Forest Classifier	11
2.5.2	Logistic Regression	11
2.5.3	Support Vector Machine	12
2.5.4	Gradient Boosting Classifier	13
2.5.5	Phương thức đánh giá	14
3	Experiments and Results	15
3.1	Experiments procedure	15
3.1.1	Data collecting	15
3.1.2	Data Exploration	17
3.1.3	Training Model	19
3.2	Results	19

Danh sách hình vẽ

1	Dữ liệu thô đã tổng hợp phục vụ cho việc phân tích	8
2	Bàn thắng kỳ vọng so với bàn thắng thực tế trong các trận đấu của Manchester City	9
3	Bàn thắng kỳ vọng của Manchester City so với bàn thắng kỳ vọng của đối thủ	10
4	Khác biệt phong độ của Manchester City và kết quả tương ứng từ mùa giải 2020	11
5	Khác biệt phong độ của Manchester United trong mùa giải 2023	12
6	Mô hình Random Forest Classifier	13
7	Mô hình Logistic Regression	13
8	Mô hình Support Vector Machine	14
9	Mô hình Gradient Boosting Classifier	14
10	Script thu thập thông tin về từng đội bóng trong mùa giải đang xét	16
11	Đoạn mã tạo khoảng nghỉ mỗi lần lấy dữ liệu nhằm tránh rate limiting . . .	17
12	Đoạn mã bỏ các trường dữ liệu không cần thiết	17
13	Tổng số trận của từng đội bóng trong khoảng thời gian 2028 - 2024	18
14	Mã hóa các giá trị object trong dữ liệu ban đầu	18
15	Mã nguồn hàm rolling_averages	19
16	Bảng dữ liệu mới sau quá trình xử lý dữ liệu ban đầu	19
17	Mã nguồn của các mô hình dự đoán đã được xây dựng	20
18	Kết quả dự đoán ban đầu	21
19	Chi tiết kết quả ban đầu	21
20	Kết quả khi chỉ dự đoán thắng hoặc không thắng	21
21	Kết quả khi dự đoán cho từng đội bóng	22
22	Kết quả khi dự đoán cho từng đội bóng	23
23	Kết quả khi dự đoán cho từng đội bóng	23

Bảng đánh giá điểm

Sinh viên	Tự đánh giá	Nhóm đánh giá	Chữ ký
Nguyễn Đức Phong	B	B	
Đào Xuân Nghĩa	A	A	
Nguyễn Minh Quân	A	A	
Lê Bùi Sơn	C	C	
Phạm Ngọc Thạch	A	A	

1 Introduction

1.1 Motivation

Bóng đá - trò chơi được mệnh danh là 'môn thể thao vua' với lượng người hâm mộ yêu thích và theo dõi đông đảo nhất thế giới. Liên đoàn Bóng đá Quốc tế - Fédération Internationale de Football Association (FIFA), ước tính rằng vào đầu thế kỷ 21 có khoảng 250 triệu cầu thủ bóng đá và hơn 1,3 tỷ người quan tâm đến môn thể thao này. Sự nổi tiếng của bộ môn này đã tạo ra số lượng lớn các người hâm mộ đến từ mọi nơi trên khắp thế giới, đặc biệt là ở giải đấu được coi là hấp dẫn hàng tinh - English Premier League - EPL. Do đó, việc dự đoán và ước tính kết quả một trận đấu trước khi nó diễn ra là rất cần thiết và hấp dẫn, nhất là đối với những fan hâm mộ, nhà cái bóng đá cũng như các chuyên gia và nhà nghiên cứu. Thế nhưng, việc dự đoán chính xác kết quả của một trận đấu bóng đá là rất khó khăn do có rất nhiều yếu tố có thể ảnh hưởng đến kết quả trận đấu như kỹ năng, sự kết hợp giữa các cầu thủ, phong độ của cầu thủ chủ chốt, tinh thần đồng đội, lợi thế sân nhà và nhiều yếu tố khác. Ngoài ra, việc dự đoán còn trở nên khó khăn hơn khi các biến số trong trận đấu thay đổi như việc cho phép thêm thời gian hiệp phụ, thay đổi cầu thủ dự bị hoặc chấn thương, ... Nhiều nghiên cứu còn chỉ ra kết quả trận đấu còn bị ảnh hưởng bởi màu áo của hai đội.

Sau quá trình tìm hiểu và chọn lọc, nhóm đã nghiên cứu và phát triển một mô hình dự hiệu quả để có thể đưa ra các dự đoán về kết quả của các trận đấu bóng đá cũng như duy trì được mức độ chính xác nhất định.

1.2 Problem statements

Bóng đá không chỉ là một môn thể thao, mà còn là một văn hóa, một phong cách sống được ngàn người trên khắp thế giới đam mê và theo đuổi. Với hàng tỉ người hâm mộ và hàng triệu cầu thủ, việc dự đoán kết quả của một trận đấu bóng đá không chỉ là một trò chơi, mà còn là một nghiên cứu sâu sắc về các yếu tố ảnh hưởng đến kết quả cuối cùng.

Ngoài ra tại rất nhiều quốc gia, trò chơi cá cược là hợp pháp và có rất nhiều người chơi. Như đã trình bày ở phần trước, bóng đá hiện tại là một môn thể thao hấp dẫn. Tất nhiên, số lượng khán giả xem bóng đá nói chung và người tham gia cá cược bóng đá là rất nhiều. Đối với nhóm người này, nhu cầu dự đoán về kết quả của các trận đấu bóng đá là rất lớn và không phải ai cũng có trình độ am hiểu đủ, kinh nghiệm đủ nhiều trong lĩnh vực này để có thể làm điều đó. Vậy nên mô hình này có thể là công cụ hỗ trợ họ đưa ra các quyết định sáng suốt. Không những vậy, bóng đá hiện đại cũng được xem như một lĩnh vực kinh doanh và kết quả trận đấu ảnh hưởng rất nhiều đến vấn đề này. Kết quả dự đoán sẽ giúp ích

rất nhiều cho các nhà quản lý trong các chiến lược quảng bá, tiếp thị cũng như phân bổ tài chính.

Tuy nhiên, việc dự đoán kết quả của một trận đấu bóng đá là một vấn đề phức tạp. Có nhiều yếu tố không thể dự đoán trước được như những tình huống bất ngờ trong trận đấu, yếu tố tâm lý tác động,... Vì vậy, việc dự đoán chính xác 100% kết quả là không thể và trong dự án này, nhóm sẽ cố gắng đạt được độ chính xác của dự đoán cao nhất có thể và có mục tiêu không chỉ là dự đoán kết quả của các trận đấu bóng đá, mà còn là cung cấp một mô hình dữ liệu hiệu quả, giúp ích những người quản lý, huấn luyện viên và nhà nghiên cứu bóng đá.

1.3 Research questions

Đầu tiên nhóm cần tìm hiểu để tìm ra được các yếu tố nào ảnh hưởng đến kết quả trận đấu bóng đá? Có thể kể đến các yếu tố như đội hình, hiệu suất của cầu thủ, sự chấn thương, điều kiện thời tiết, sân nhà hay sân khách, v.v.

Từ đó, nhóm sẽ xem xét các dữ liệu liên quan đến lịch sử của trận đấu, giải đấu English Premier League - EPL. Từ đó trả lời câu hỏi: Liệu rằng ta có thể dựa vào các dữ liệu này để đưa ra dự đoán kết quả của các trận đấu bóng đá hay không?

Phương pháp dự đoán nào được sử dụng phổ biến nhất trong việc dự đoán kết quả trận đấu bóng đá? Các phương pháp này có thể bao gồm mô hình học máy, phân tích thống kê, hay sự kết hợp của cả hai. Tiếp theo cần trả lời những câu hỏi: mô hình dự đoán nào có hiệu suất tốt nhất trong việc dự đoán kết quả trận đấu bóng đá? Làm thế nào để đánh giá và so sánh hiệu suất giữa các mô hình khác nhau?

Câu hỏi về ứng dụng thực tiễn như làm thế nào để áp dụng các phương pháp dự đoán kết quả trận đấu bóng đá vào các lĩnh vực như cá cược, quản lý đội bóng, hoặc phân tích thể thao?

2 Materials and Methods

2.1 Data Collection

Hiện tại, có rất nhiều cách thức, công cụ có thể được sử dụng để thu thập dữ liệu cho một mô hình dự đoán kết quả trận đấu bóng đá. Dưới đây là một số cách phổ biến:

- Dữ liệu Trận Đấu Trực Tiếp (Live Match Data): Thu thập dữ liệu trực tiếp từ các trận đấu đang diễn ra có thể cung cấp thông tin về các sự kiện trong trận đấu như bàn thắng, thẻ phạt, thời gian kiểm soát bóng, số lần sút bóng,... và các thống kê khác. Các dịch vụ như Opta, StatsBomb, ... cung cấp dữ liệu trực tiếp này.

- **Dữ liệu Lịch Sử Trận Đấu (Historical Match Data):** Thu thập dữ liệu từ các trận đấu đã diễn ra trong quá khứ để phân tích xu hướng và mẫu đặc điểm. Dữ liệu này bao gồm kết quả của các trận đấu, số bàn thắng, thẻ phạt, thời gian kiểm soát bóng,... và nhiều yếu tố khác.
- **Dữ liệu Cầu Thủ (Player Data):** Thu thập dữ liệu về các cầu thủ, bao gồm thông tin về kỹ năng, phong độ, thể lực, sức khỏe, và lịch sử chấn thương. Các yếu tố này có thể ảnh hưởng đến hiệu suất của mỗi cầu thủ trong một trận đấu.
- **Dữ liệu Đội Bóng (Team Data):** Thu thập dữ liệu về các đội bóng, bao gồm thông tin về đội hình, chiến thuật, lối chơi, và phong độ trong mùa giải hiện tại và mùa giải trước đó.
- **Dữ liệu Thị Trường Chuyển Nhượng (Transfer Market Data):** Thu thập dữ liệu về giá trị chuyển nhượng của cầu thủ, thông tin về các giao kèo, và các thương vụ chuyển nhượng có thể cung cấp thông tin về sự gia tăng hoặc giảm sút chất lượng của một đội bóng.

Các loại dữ liệu kể trên có thể kết hợp lại để tạo thành một bộ dữ liệu đa dạng và phong phú, giúp cải thiện độ chính xác của mô hình dự đoán kết quả trận đấu bóng đá. Để hỗ trợ cho việc thu thập các dữ liệu trên, nhóm sẽ chủ yếu crawl data từ trang *fbref.com*, cung cấp đầy đủ kết quả trận cho đến hiện tại của giải đấu EPL.

Dữ liệu cho báo cáo được thu thập thông qua quá trình crawl dữ liệu (web scraping) từ trang web FBref¹. FBref cung cấp một nguồn dữ liệu trực tuyến uy tín và toàn diện về bóng đá, cung cấp thông tin chi tiết về các trận đấu, cầu thủ, đội bóng và giải đấu trên toàn thế giới.

Thư viện BeautifulSoup⁴ được sử dụng để phân tích cú pháp HTML của các trang web FBref. Thư viện này cho phép trích xuất thông tin cụ thể từ các thẻ HTML, chẳng hạn như tên đội, tỷ số, ngày thi đấu và các số liệu thống kê khác liên quan đến trận đấu.

Quá trình thu thập dữ liệu được thực hiện tự động bằng một script Python có tên *data_collection* do nhóm chúng tôi xây dựng, script này cho phép thu thập một lượng lớn dữ liệu một cách hiệu quả và nhất quán. Dữ liệu thu thập được được lưu dưới dạng csv, bao gồm thông tin về tất cả các trận đấu trong một mùa giải EPL cụ thể, cung cấp một tập dữ liệu phong phú cho việc phân tích và nghiên cứu tiếp theo.

¹<https://fbref.com/en/comps/9/schedule/Premier-League-Scores-and-Fixtures>

2.2 Data Preparing

Sau khi đã xác định được nguồn dữ liệu và nghiên cứu sử dụng các công cụ cần thiết cho việc thu thập dữ liệu phục vụ cho việc dự đoán, nhóm đã tổng hợp được dữ liệu thô dưới dạng .csv, từ đó chuyển được sang dạng phù hợp hơn để xử lý, bao gồm các trường thông tin như sau:

- date : Ngày diễn ra trận đấu
- time : Giờ diễn ra trận đấu
- round: Trận đấu thuộc vòng thứ bao nhiêu trong mùa giải
- day: Ngày nào trong tuần diễn ra trận đấu
- season: Mùa giải trận đấu diễn ra
- team: Đội bóng trong trận đấu
- opponent: Đối thủ của đội bóng trong trận đấu
- venue: Trận đấu được diễn ra trên sân nhà hay sân khách
- result: Kết quả của trận đấu
- gf: Bàn thắng mà đội bóng ghi được
- ga: Bàn thắng mà đối thủ ghi được
- xg: Bàn thắng kì vọng của đội bóng
- xga: Bàn thắng kì vọng của đối thủ
- poss: Phần trăm kiểm soát bóng của đội bóng
- attendance: Số khán giả có trong trận đấu
- captain: Đội trưởng của đội bóng
- referee: Trọng tài bắt chính trong trận đấu
- sh: Tổng số cú sút của đội bóng trong trận đấu
- sot: Số cú sút trúng đích của đội bóng trong trận đấu
- dist: Khoảng cách trung bình của các cú sút của đội bóng

- fk: Số cú sút phạt của đội bóng trong trận đấu
- pk: Số cú sút penalty thành bàn của đội bóng trong trận đấu
- pkatt: Tổng số cú sút penalty của đội bóng trong trận đấu

Để phục vụ cho việc tiền xử lý, nhóm đã thu thập dữ liệu lịch sử đấu tất cả các câu lạc bộ của giải EPL từ đầu mùa giải 2018 đến nay với tổng cộng 5272 hàng dữ liệu và 23 cột dữ liệu tương đương với 23 trường dữ liệu thô đã trình bày ở trên. Chi tiết được thể hiện trong Bảng 1.

	date	time	round	day	venue	result	gf	ga	opponent	xg	...	captain	referee	sh	sot	dist	fk	pk	pkatt	season	team
1	8/12/2023	12:30	Matchweek 1	Sat	Home	W	2	1	Nott'ham Forest	0.8	...	Martin Ødegaard	Michael Oliver	15	7	19.1	0	0	0	2024	Arsenal
2	8/21/2023	20:00	Matchweek 2	Mon	Away	W	1	0	Crystal Palace	2.0	...	Martin Ødegaard	David Coote	13	2	16.4	0	1	1	2024	Arsenal
3	8/26/2023	15:00	Matchweek 3	Sat	Home	D	2	2	Fulham	3.2	...	Martin Ødegaard	Paul Tierney	18	9	13.8	0	1	1	2024	Arsenal
4	9/3/2023	16:30	Matchweek 4	Sun	Home	W	3	1	Manchester Utd	2.3	...	Martin Ødegaard	Anthony Taylor	17	5	15.0	0	0	0	2024	Arsenal
5	9/17/2023	16:30	Matchweek 5	Sun	Away	W	1	0	Everton	1.0	...	Martin Ødegaard	Simon Hooper	13	4	17.4	0	0	0	2024	Arsenal
...
38	4/15/2018	16:00	Matchweek 34	Sun	Away	W	1	0	Manchester Utd	0.7	...	Chris Brunt	Paul Tierney	10	4	18.1	0	0	0	2018	West Bromwich Albion
39	4/21/2018	12:30	Matchweek 35	Sat	Home	D	2	2	Liverpool	1.3	...	Chris Brunt	Stuart Attwell	13	6	17.7	0	0	0	2018	West Bromwich Albion
40	4/28/2018	15:00	Matchweek 36	Sat	Away	W	1	0	Newcastle Utd	0.7	...	Chris Brunt	David Coote	9	2	20.1	0	0	0	2018	West Bromwich Albion
41	5/5/2018	15:00	Matchweek 37	Sat	Home	W	1	0	Tottenham	1.6	...	Chris Brunt	Mike Jones	9	1	10.2	0	0	0	2018	West Bromwich Albion
42	5/13/2018	15:00	Matchweek 38	Sun	Away	L	0	2	Crystal Palace	0.2	...	Chris Brunt	Jonathan Moss	7	1	24.8	1	0	0	2018	West Bromwich Albion

5272 rows × 23 columns

Hình 1: Dữ liệu thô đã tổng hợp phục vụ cho việc phân tích

2.3 Data Preprocessing

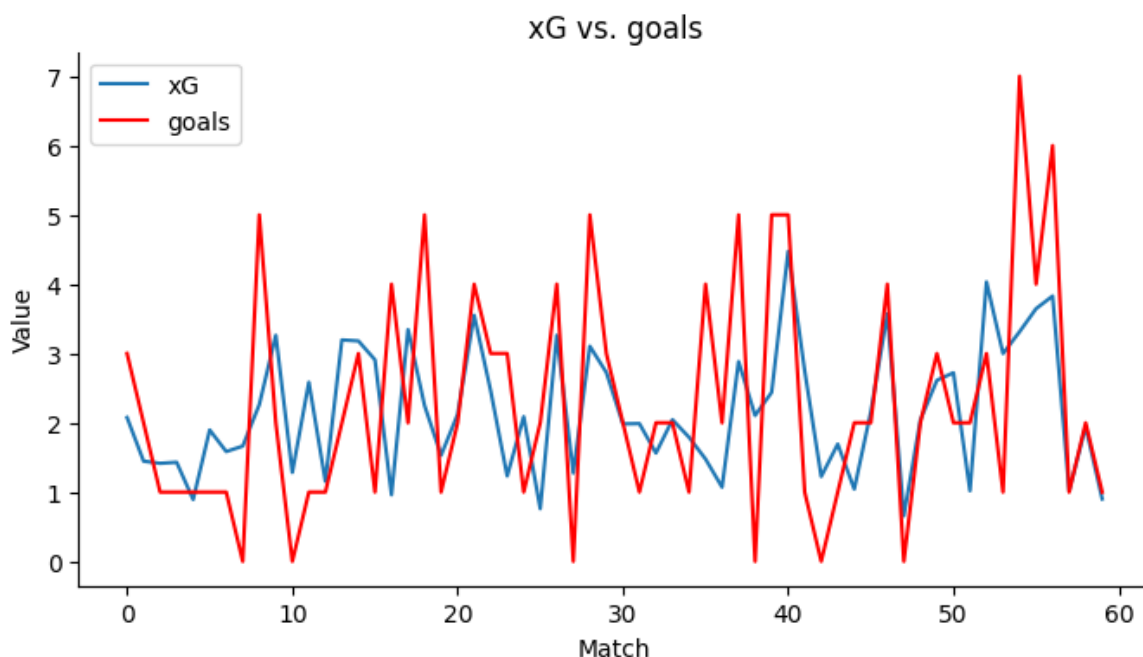
Từ Bảng dữ liệu thô 1, ta có thể thấy chỉ với một số trường dữ liệu các thông số cơ bản của các trận đấu là không đủ để có thể sử dụng làm dữ liệu chính cho việc dự đoán. Vì vậy nhóm đã áp dụng các bước tiền xử lý dữ liệu cho bộ dữ liệu thô đã có để có được bộ dữ liệu đủ chất lượng. Cụ thể gồm ba bước như sau:

- **Làm sạch dữ liệu:** Xác định các giá trị không tồn tại hoặc không hợp lệ trong bộ dữ liệu. Chỉnh sửa các sai sót được phát hiện bằng cách tính giá trị trung bình (mean) của các feature có sẵn và sử dụng giá trị này để thay thế các giá trị không tồn tại hoặc nhiễu. Và cuối cùng là đánh giá lại dữ liệu.
- **Chuyển dạng dữ liệu:** Dữ liệu sẽ được xử lý qua các bước làm trơn (Smoothing), chuẩn hóa (Normalization), ... để đưa về dạng có thể xử lý được ở các pha xử lý cao hơn.
- **Xây dựng thuộc tính mới:** Sau khi đã có được dữ liệu thô đã được chuẩn hóa về dạng có thể phân tích và xử lý phức tạp, bước tiếp theo là tạo các thuộc tính mới vào các trường dữ liệu do dữ liệu thô ban đầu là không đủ cho việc phân tích dự đoán. Các thuộc tính này được xây dựng dựa theo trường hợp sử dụng.

2.4 Data Visualization

Trực quan hóa dữ liệu là một bước quan trọng trong việc phân tích và diễn giải dữ liệu. Nó liên quan đến việc trình bày dữ liệu dưới dạng biểu đồ, đồ thị và bản đồ để tạo điều kiện thuận lợi cho việc hiểu và thu thập thông tin từ dữ liệu. Trong ngữ cảnh dự đoán kết quả các trận đấu bóng đá trong giải Ngoại hạng Anh (EPL), trực quan hóa dữ liệu có thể cung cấp thông tin quan trọng về các mẫu, xu hướng và mối quan hệ giữa các trường dữ liệu liên quan đến việc dự đoán kết quả trận đấu. Trong phần này, nhóm sẽ sử dụng các kỹ thuật, thư viện hỗ trợ trực quan hóa dữ liệu cụ thể để khám phá và truyền đạt dữ liệu theo cách rõ ràng hơn để giúp chúng ta hiểu và áp dụng vào mô hình dự đoán kết quả các trận đấu bóng đá trong giải đấu EPL. Cụ thể, các biểu đồ nhóm đã xây dựng được trình bày trong Hình 2 đến 5.

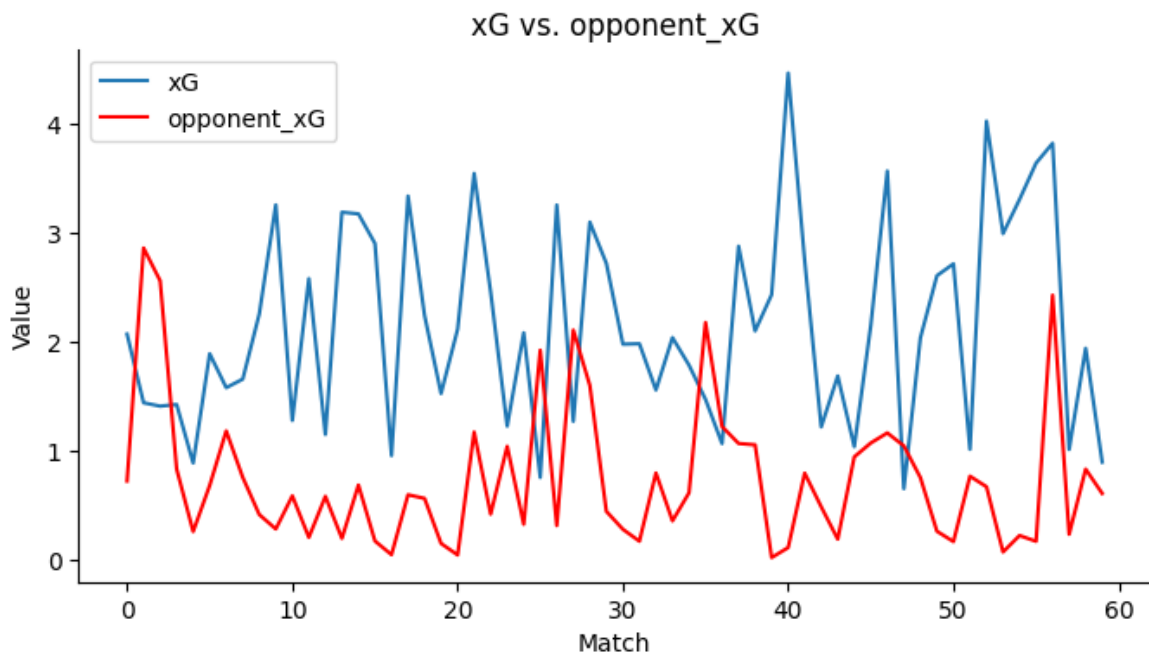
Trong Hình 2, **goals** cho chúng ta biết số lượng bàn thắng thực sự được ghi bởi một đội bóng trong một trận đấu, trong khi **xG** cho chúng ta biết số lượng bàn thắng dự kiến mà một đội bóng nên ghi dựa trên các cơ hội tạo ra và chịu phải trong suốt trận đấu. Như trong hình vẽ, đa phần goals đều cao hơn so với xG, điều này có thể là do các cầu thủ có khả năng tận dụng cơ hội thành bàn thắng tốt hơn so với dự đoán. Việc tận dụng cơ hội tốt thường liên quan mật thiết đến khả năng chiến thắng của một đội, đội bóng nào có khả năng ghi bàn cao hơn sẽ có cơ hội cao hơn để giành chiến thắng. Thực tế đã cho thấy rằng, Manchester City là một trong những đội bóng có tỉ lệ chiến thắng cao nhất tại châu Âu.



Hình 2: Bàn thắng kỳ vọng so với bàn thắng thực tế trong các trận đấu của Manchester City

Nếu xét về sự chênh lệch **xG** giữa Manchester City và đối thủ trong Hình 3, ta có thể

thấy rõ ràng Manchester City trong hầu hết các trận đấu đều có ưu thế lớn hơn so với đối thủ khi mà có nhiều cơ hội ghi bàn hơn trong trận đấu và sẽ có thể giành được chiến thắng nếu thành công chuyển hóa được các cơ hội đó thành bàn thắng.

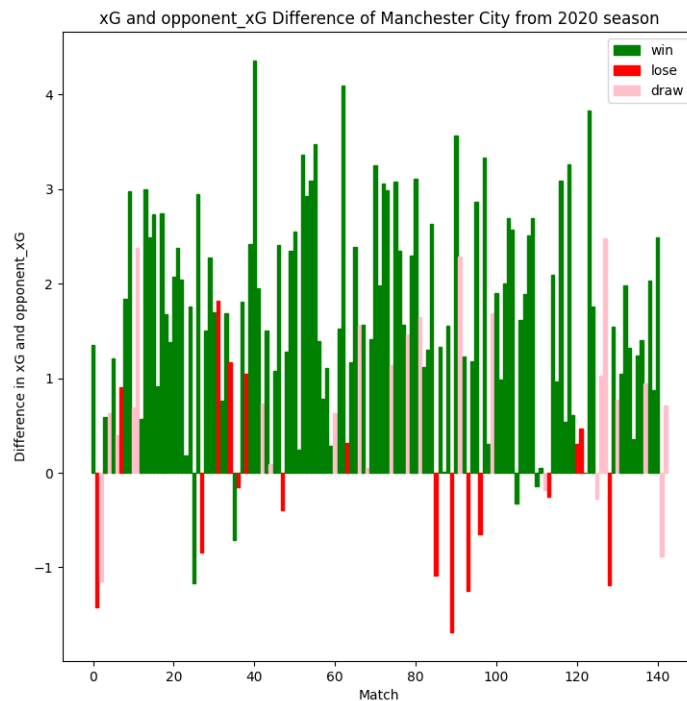


Hình 3: Bàn thắng kỳ vọng của Manchester City so với bàn thắng kỳ vọng của đối thủ

Về **phong độ** của các đội bóng, khi xét Manchester City với đối thủ (Hình 4), ta có thể thấy việc có phong độ cao hơn (biểu hiện ở việc độ khác biệt phong độ so với đối thủ lớn hơn 0) sẽ dẫn đến khả năng chiến thắng cao hơn. Đa số các cột chỉ số dương đều là kết quả chiến thắng, ngược lại, các chỉ số âm thường là kết quả bất lợi. Từ đây, việc dự đoán kết quả cho các đội như Manchester City thường sẽ cho độ chính xác khá cao, do không có quá nhiều sự đột biến trong dữ liệu. Tuy nhiên, khi xét đến của các đội bóng khác có phong độ được cho là thất thường trong các mùa giải gần đây chẳng hạn như Manchester United thì kết quả thường sẽ có nhiều sự đột biến. Cụ thể được thể hiện trong Hình 5, có thể thấy rằng khoảng một nửa số trận khi đội bóng này có phong độ cao hơn đối thủ thì kết quả trận đấu lại là thua hoặc hòa và ngược lại, gần một nửa số trận Manchester United có phong độ thấp hơn thì thắng. Khi đây, việc dự đoán kết quả trận đấu cho các đội bóng có phong độ như trên thường sẽ cho ra kết quả không cao.

2.5 Method

Trong phần này, nhóm sẽ giới thiệu một số mô hình và thuật toán học máy được sử dụng trong báo cáo cũng như một số phương thức để đánh giá kết quả của các mô hình này.



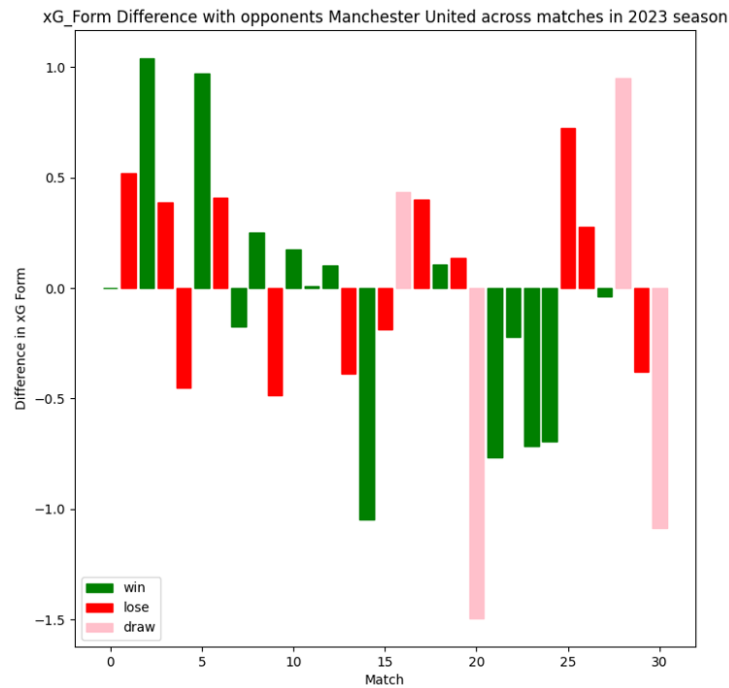
Hình 4: Khác biệt phong độ của Manchester City và kết quả tương ứng từ mùa giải 2020

2.5.1 Random Forest Classifier

Random Forest là một thuật toán học máy được sử dụng khá phổ biến. "Random" là ngẫu nhiên, "Forest" là rừng. Vì vậy, thuật toán này là sự kết hợp giữa nhiều cây quyết định (Decision Tree), mỗi cây lại quyết định các yếu tố khác nhau. Sau đó, kết quả sẽ được tổng hợp từ các cây quyết định. Random Forest có khả năng xử lý các tập dữ liệu lớn một cách hiệu quả và thường không cần dành quá nhiều công sức cho việc tiền xử lý dữ liệu hoặc tinh chỉnh tham số phức tạp. Nó cũng có khả năng xử lý các tập dữ liệu chứa các giá trị bị khuyết (missing) một cách linh hoạt, không cần phải điền vào các giá trị thiếu trước khi xử lý. Khác với các cây quyết định đơn lẻ, Random Forest thường không bị overfitting nhờ có sự đa dạng và ngẫu nhiên trong quá trình xây dựng các cây quyết định. Thêm vào đó, thuật toán này có thể được sử dụng cho cả bài toán phân loại và hồi quy. Với những đặc điểm trên, Random Forest trở nên khá phù hợp với bài toán dự đoán kết quả thể thao nói chung và bóng đá nói riêng.

2.5.2 Logistic Regression

Một mô hình học máy khác mà nhóm muốn đề cập trong báo cáo này đó là Logistic Regression. Logistic Regression là 1 thuật toán phân loại được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc (như 0, 1, 2, ... hay Thắng, Hoà, Thua trong kết quả trận đấu). Logistic Regression sử dụng một hàm logistic để chuyển đổi đầu vào tuyến tính thành một

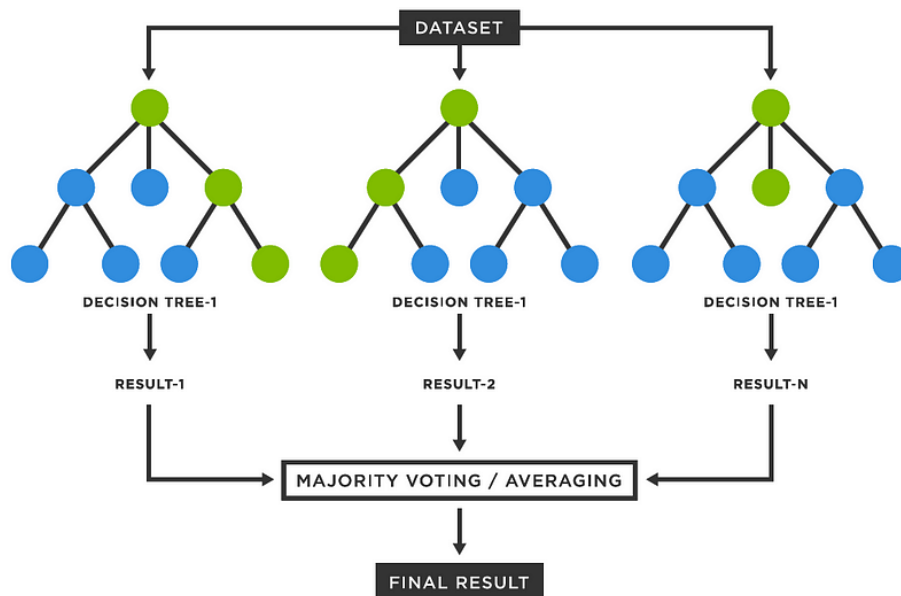


Hình 5: Khác biệt phong độ của Manchester United trong mùa giải 2023

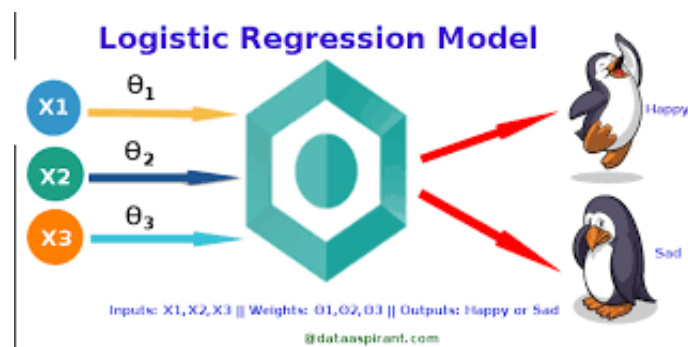
giá trị xác suất nằm trong khoảng từ 0 đến 1. Mô hình này có thể được diễn giải dễ dàng, giúp hiểu cách mỗi biến độc lập ảnh hưởng đến xác suất của kết quả phân loại. Quá trình huấn luyện Logistic Regression thường bao gồm việc tối ưu hóa các tham số của mô hình để tối thiểu hóa sai số giữa dự đoán và thực tế sử dụng các phương pháp như Gradient Descent. Tóm lại, Logistic Regression có thể được sử dụng cho bài toán dự đoán kết quả trận đấu bóng đá, nhưng cần phải xem xét các yếu tố về độ phức tạp của dữ liệu và tính chính xác của mô hình khi lựa chọn.

2.5.3 Support Vector Machine

Support Vector Machine (SVM) là một trong những thuật toán phân loại mạnh mẽ và linh hoạt trong lĩnh vực học máy. Mục tiêu chính của SVM là tìm ra một ranh giới phân chia tối ưu giữa các lớp dữ liệu sao cho khoảng cách giữa các điểm dữ liệu gần nhất đến ranh giới đó là lớn nhất, được gọi là biên lớn nhất. Điều này giúp SVM có khả năng phân loại tốt và tổng quát hóa tốt trên dữ liệu mới. Mặc dù ban đầu được thiết kế cho các bài toán phân loại tuyến tính, SVM đã được mở rộng để áp dụng cho các bài toán phân loại phi tuyến tính thông qua việc sử dụng các hàm kernel, giúp chuyển đổi dữ liệu vào một không gian chiều cao hơn để phân loại các lớp dữ liệu phức tạp hơn. SVM cũng có khả năng xử lý hiệu quả các tập dữ liệu lớn và ít bị ảnh hưởng bởi số lượng đặc trưng so với một số thuật toán khác. Nó cũng có các tham số quan trọng như hằng số điều chỉnh độ quan trọng của các điểm dữ liệu (C), và loại hàm kernel được sử dụng, cho phép điều chỉnh và tinh chỉnh



Hình 6: Mô hình Random Forest Classifier

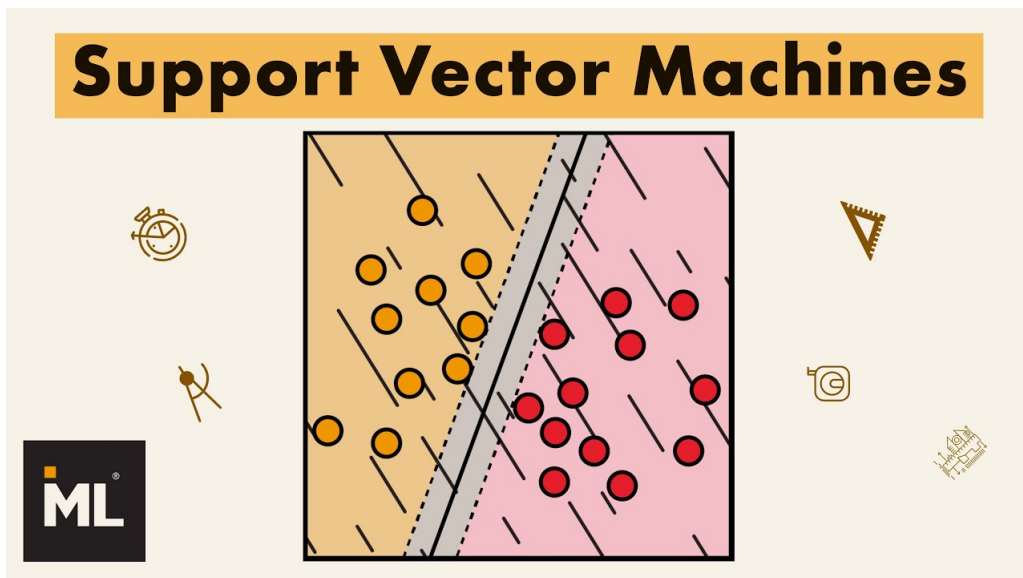


Hình 7: Mô hình Logistic Regression

SVM cho phù hợp với dữ liệu cụ thể và cải thiện hiệu suất. Không chỉ được sử dụng cho bài toán phân loại, SVM cũng có thể được áp dụng cho bài toán hồi quy, làm cho nó trở thành một công cụ linh hoạt và mạnh mẽ trong học máy.

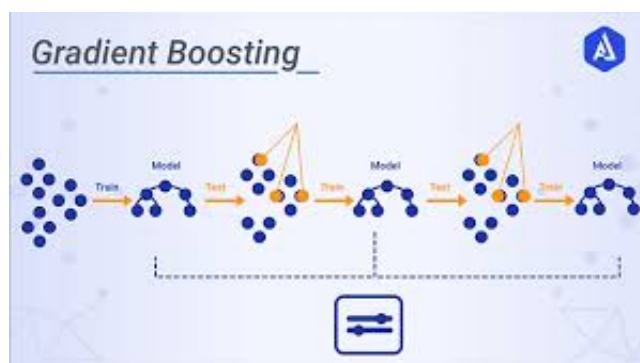
2.5.4 Gradient Boosting Classifier

Gradient Boosting Classifier là một thuật toán học máy mạnh mẽ được sử dụng rộng rãi cho các bài toán phân loại. Cũng giống như Random Forest, nó hoạt động bằng cách xây dựng các cây quyết định nhưng theo cách tuần tự, mỗi cây sẽ cố gắng cải thiện việc dự đoán của cây trước đó. Điều này được thực hiện bằng cách tập trung vào việc điều chỉnh sai số của mô hình trước đó thay vì xây dựng một mô hình hoàn toàn mới từ đầu. Mỗi cây quyết định mới được xây dựng dựa trên các lỗi dự đoán của các cây trước đó, với mục tiêu là dự đoán đúng những điểm mà các cây trước đó đã dự đoán sai. Thuật toán thường sử dụng các kỹ thuật regularization để tránh overfitting, bằng cách giảm độ sâu của các cây hoặc giảm



Hình 8: Mô hình Support Vector Machine

hệ số học của mỗi cây. Một điểm mạnh của Gradient Boosting là khả năng sử dụng các hàm loss function tùy chỉnh, cho phép tối ưu hóa cho nhiều loại bài toán phân loại khác nhau. Với hiệu suất cao và khả năng linh hoạt, Gradient Boosting Classifier thường được ưa chuộng trong các ứng dụng thực tế, đặc biệt là khi kết hợp với các siêu tham số tốt.



Hình 9: Mô hình Gradient Boosting Classifier

2.5.5 Phương thức đánh giá

Việc đánh giá được thực hiện thông qua phương thức `classification_report` trong thư viện `sklearn.metrics`. Phương thức này sẽ xuất ra một bảng báo cáo chi tiết về hiệu suất của mô hình, bao gồm các chỉ số như precision, recall, và F1-score cho mỗi lớp dữ liệu, cũng như trung bình của các chỉ số này qua tất cả các lớp. Nó cung cấp một cái nhìn tổng quan về cách mô hình của bạn hoạt động trên dữ liệu thử nghiệm và là một công cụ hữu ích để đánh giá hiệu suất của mô hình được sử dụng.

- precision: Độ chính xác của mô hình trong việc dự đoán một lớp. Được tính bằng cách

chia số lượng dự đoán đúng cho tổng số lượng dự đoán của lớp đó.

- **recall (hoặc độ nhạy):** Tỷ lệ giữa số lượng dự đoán đúng và tổng số lượng mẫu thực sự thuộc về lớp đó.
- **f1-score:** Điểm số F1 là trung bình điều hòa của precision và recall. Điểm số F1 càng cao, mô hình càng tốt.
- **support:** Số lượng mẫu thực tế trong mỗi lớp trong tập dữ liệu kiểm tra.
- **accuracy:** Đây là tỷ lệ tổng số dự đoán chính xác so với tổng số dự đoán. Được tính bằng cách chia số lượng dự đoán đúng cho tổng số lượng dự đoán.
- **macro avg:** Đây là trung bình đơn giản của các chỉ số (precision, recall, f1-score) trên tất cả các lớp, không xem xét số lượng mẫu trong mỗi lớp. Điều này có thể hữu ích khi bạn có các lớp không cân đối và muốn đánh giá mô hình một cách công bằng trên tất cả các lớp.
- **weighted avg:** Đây là trung bình có trọng số của các chỉ số trên tất cả các lớp, với trọng số là số lượng mẫu trong mỗi lớp. Điều này có thể hữu ích khi bạn có các lớp không cân đối và muốn đánh giá mô hình dựa trên tầm quan trọng của mỗi lớp.

3 Experiments and Results

3.1 Experiments procedure

3.1.1 Data collecting

Để phục vụ cho quá trình thu thập dữ liệu, nhóm chúng tôi đã xây dựng một script Python có tên *data_collection*² để tự động hóa quá trình này. Script này được thiết kế để thu thập dữ liệu bóng đá chi tiết từ trang web FBref từ mùa 2018 đến mùa 2024 bằng cách lấy dữ liệu trong từng năm sau đó truy cập vào trang thống kê của từng đội bóng trong giải Ngoại hạng Anh và thu thập các thông tin theo yêu cầu, cụ thể được thể hiện trong Hình 10. Các thư viện chính được sử dụng trong Script này bao gồm:

- **requests:** Để tải nội dung trang web.
- **BeautifulSoup4:** Để phân tích cú pháp HTML và trích xuất thông tin.
- **pandas:** Để xử lý và phân tích dữ liệu dạng bảng.

²<https://colab.research.google.com/drive/1QPLYgIp-oULtn24axekAuh2lscWJ1bM8?usp=sharing>

- **numpy**: Để thực hiện các phép toán số học (ví dụ: thêm cột NaN).
- **time**: Để tạo các khoảng dừng giữa các requests được gửi đến trang web.
- **random**: Để tạo các khoảng dừng ngẫu nhiên.

Đối với mỗi đội bóng trong giải Ngoại hạng Anh, script này không chỉ dừng lại ở việc thu thập thông tin cơ bản như tên đội và mùa giải, mà còn tiến sâu hơn để thu thập dữ liệu chi tiết về hai khía cạnh quan trọng trong bóng đá:

- **Thông tin trận đấu ("Scores & Fixtures")**: Đây là phần dữ liệu ghi lại kết quả của từng trận đấu mà đội bóng tham gia trong mùa giải. Nó bao gồm các thông tin quan trọng như ngày diễn ra trận đấu, đối thủ, tỷ số, địa điểm thi đấu (sân nhà hay sân khách), và có thể cả các sự kiện khác trong trận đấu như thẻ vàng, thẻ đỏ, hay các tình huống thay người.
- **Số liệu sút bóng ("Shooting")**: Đây là phần dữ liệu chuyên sâu hơn, tập trung vào khả năng tấn công của đội bóng. Nó bao gồm số lần sút (Sh), số lần sút trúng đích (SoT), khoảng cách trung bình của các cú sút (Dist), số lần sút phạt trực tiếp (FK), ...

```
links = [l.get("href") for l in standings_table.find_all('a')]
links = [l for l in links if '/squads/' in l]
team_urls = [f"https://fbref.com{l}" for l in links]

previous_season = soup.select("a.prev")[0].get("href")
standings_url = f"https://fbref.com{previous_season}"

for team_url in team_urls:
    team_name = team_url.split("/")[-1].replace("-Stats", "").replace("-", " ")
    data = requests.get(team_url)
    matches = pd.read_html(data.text, match="Scores & Fixtures")[0]
    soup = BeautifulSoup(data.text)
    links = [l.get("href") for l in soup.find_all('a')]
    links = [l for l in links if l and 'all_comps/shooting/' in l]
    data = requests.get(f"https://fbref.com{links[0]}")
    shooting = pd.read_html(data.text, match="Shooting")[0]
```

Hình 10: Script thu thập thông tin về từng đội bóng trong mùa giải đang xét

Việc tránh giới hạn truy cập (rate limiting) trong quá trình thu thập dữ liệu từ các trang web đóng vai trò quan trọng để đảm bảo việc truy cập thông tin diễn ra suôn sẻ và không gây quá tải cho máy chủ. Cụ thể để làm việc này, sau mỗi lần lấy dữ liệu về một đội bóng, đoạn mã sẽ chủ động tạm dừng một khoảng thời gian ngẫu nhiên từ 1.2 đến 3.8 giây, cụ thể trong Hình 11. Khoảng dừng này có tác dụng như một "khoảng nghỉ" giữa các yêu cầu liên

tiếp, giúp tránh việc gửi quá nhiều yêu cầu trong một khoảng thời gian ngắn, điều có thể khiến FBref hiểu nhầm là hành vi spam hoặc tấn công. Việc chủ động giới hạn tốc độ này đảm bảo rằng quá trình thu thập dữ liệu diễn ra ổn định, liên tục và không bị gián đoạn do các biện pháp hạn chế từ phía FBref. Cuối cùng, tất cả dữ liệu thu được sẽ được tổng hợp và lưu vào tệp *matches.csv*

```
sleep_time = random.randint(10, 19)/5  
time.sleep(sleep_time)
```

Hình 11: Đoạn mã tạo khoảng nghỉ mỗi lần lấy dữ liệu nhằm tránh rate limiting

3.1.2 Data Exploration

Để phục vụ cho việc khám phá dữ liệu cũng như áp dụng các mô hình cho việc dự đoán, nhóm đã xây dựng thêm một script Python có tên *Prediction*³ cho việc này. Đầu tiên, script sẽ đọc file “matches.csv” đã thu được ở phần trước và lưu dưới dạng “pandas dataframe”. Sau khi nhóm phân tích dữ liệu, các trường thông tin “comp”, “match report”, “formation” và “notes” được nhận xét là không cần thiết cho việc biến đổi về sau cũng như việc dự đoán cho nên các cột này sẽ bị loại bỏ bằng đoạn mã trong Hình 12. Sau bước này, dữ liệu ban đầu thu thập được sẽ gồm 5272 hàng và 23 cột như đã đề cập trong Hình 1.

```
matches = matches.drop("comp", axis=1)  
matches = matches.drop("match report", axis=1)  
matches = matches.drop("formation", axis=1)  
matches = matches.drop("notes", axis=1)  
matches
```

Hình 12: Đoạn mã bỏ các trường dữ liệu không cần thiết

Nhóm đã phân tích dữ liệu ban đầu và có một số nhận xét như sau:

- Có thể thấy trong Hình 13, có sự chênh lệch về số lượng dữ liệu, lý do cho việc này là các đội bóng hàng đầu như Liverpool, Manchester City, Manchester United, Arsenal, Everton, ... hầu như chưa bao giờ bị xuống hạng. Ngược lại, với các đội bóng khác, việc số lượng dữ liệu thấp là do họ không được tham dự EPL ở một số mùa (do xuống hạng hoặc các đội chỉ mới lên hàng mùa gần đây). Do dữ liệu được cập nhật khá gần so với thực tế và hiện tại thì giải đấu vẫn chưa kết thúc cho nên dữ liệu cũng có thể chênh lệch khi ta xét dựa trên các vòng đấu.

³<https://colab.research.google.com/drive/112UKtXNFIE42n11etIMGkFO2raZmbn4l>

team	
Arsenal	264
Everton	264
Liverpool	264
West Ham United	264
Tottenham Hotspur	263
Newcastle United	263
Chelsea	263
Brighton and Hove Albion	263
Manchester City	263
Crystal Palace	263
Manchester United	262
Southampton	228
Leicester City	228
Burnley	226
Wolverhampton Wanderers	226
Bournemouth	188
Aston Villa	188
Watford	152
Fulham	150
Leeds United	114
Brentford	112
Sheffield United	112
Norwich City	76
West Bromwich Albion	76
Huddersfield Town	76
Nottingham Forest	74
Cardiff City	38
Swansea City	38
Stoke City	38
Luton Town	36

Hình 13: Tổng số trận của từng đội bóng trong khoảng thời gian 2028 - 2024

- Hầu hết kiểu dữ liệu của các thuộc tính trong dữ liệu ban đầu đều ở dạng số (float64, int64). Tuy nhiên, vẫn còn một số thuộc tính có dạng object cho nên để có thể đưa vào mô hình học máy, ta cần có một chút biến đổi. Ví dụ như các thuộc tính: “venue” thể hiện sân nhà hay sân khách, “team” và “opponent” là tên của đội nhà và đội khách hay “time” và “date” sẽ được tạo thành các trường thuộc tính mới với giá trị là số tương trưng cho mỗi giá trị ban đầu. Cụ thể được thực hiện như đoạn mã trong Hình 14. Ngoài ra, mục tiêu dự đoán của chúng ta đó là kết quả Thắng/Hòa/Thua cũng được đưa về giá trị số lần lượt là 2, 1 và 0.

```
[257] matches["venue_code"] = matches["venue"].astype("category").cat.codes

[258] matches["opp_code"] = matches["opponent"].astype("category").cat.codes

[259] matches["team_code"] = matches["team"].astype("category").cat.codes

[260] matches["hour"] = matches["time"].str.replace(":.+", "", regex=True).astype("int")

[261] matches["day_code"] = matches["date"].dt.dayofweek
```

Hình 14: Mã hóa các giá trị object trong dữ liệu ban đầu

Sau khi thử sử dụng các thuộc tính đã có để thực hiện việc dự đoán, nhóm nhận thấy

kết quả có được không cao. Vì vậy tạo ra các trường dữ liệu liên quan đến phong độ (form) dựa trên trung bình thông số từ các trận đấu trước. Việc này được thực hiện thông qua hàm “rolling_averages” do nhóm xây dựng như trong Hình 15.

```
def rolling_averages(group, cols, new_cols):
    group = group.sort_values("date")
    rolling_stats = group[cols].rolling(3, closed='left').mean()
    group[new_cols] = rolling_stats
    group = group.dropna(subset=new_cols)
    return group
```

Hình 15: Mã nguồn hàm rolling_averages

Sau các bước xử lý đã nêu trên, dữ liệu mới sẽ có dạng như trong Hình 16. Bao gồm 5272 dòng và 39 cột tương ứng với 38 trường dữ liệu.

	date	time	round	day	venue	result	gf	ga	opponent	xg	...	gf_form	ga_form	sh_form	sot_form	dist_form	fk_form	pk_form	pkatt_form	xg_form	xga_form
0	2017-09-09	15:00	Matchweek 4	Sat	Home	W	3	0	Bournemouth	2.2	...	1.333333	2.666667	17.666667	5.333333	18.133333	0.000000	0.000000	0.000000	1.533333	1.766667
1	2017-09-17	13:30	Matchweek 5	Sun	Away	D	0	0	Chelsea	1.4	...	1.000000	1.666667	14.333333	5.000000	16.766667	0.333333	0.000000	0.000000	1.433333	1.466667
2	2017-09-25	20:00	Matchweek 6	Mon	Home	W	2	0	West Brom	2.2	...	1.000000	1.333333	12.000000	3.666667	16.566667	0.333333	0.000000	0.000000	1.400000	1.500000
3	2017-10-01	12:00	Matchweek 7	Sun	Home	W	2	0	Brighton	2.4	...	1.666667	0.000000	14.333333	5.333333	17.400000	1.333333	0.333333	0.333333	1.933333	0.766667
4	2017-10-14	17:30	Matchweek 8	Sat	Away	L	1	2	Watford	1.0	...	1.333333	0.000000	17.000000	5.000000	18.333333	1.666667	0.333333	0.333333	2.000000	0.700000
...
5168	2024-04-13	15:00	Matchweek 33	Sat	Away	D	2	2	Nottingham Forest	1.0	...	0.666667	1.666667	11.333333	3.666667	17.466667	0.666667	0.333333	0.333333	1.300000	1.166667
5169	2024-04-20	19:30	Matchweek 34	Sat	Home	L	0	2	Arsenal	0.2	...	1.333333	1.666667	10.666667	3.666667	17.966667	0.333333	0.333333	0.333333	1.133333	1.466667
5170	2024-04-24	19:45	Matchweek 29	Wed	Home	L	0	1	Bournemouth	0.5	...	1.000000	2.000000	9.666667	4.000000	20.533333	0.333333	0.333333	0.333333	0.900000	1.433333
5171	2024-04-27	15:00	Matchweek 35	Sat	Home	W	2	1	Luton Town	1.2	...	0.666667	1.666667	10.333333	3.333333	18.766667	0.000000	0.000000	0.000000	0.566667	1.666667
5172	2024-05-04	17:30	Matchweek 36	Sat	Away	L	1	5	Manchester City	0.3	...	0.666667	1.333333	11.000000	4.000000	19.666667	0.000000	0.000000	0.000000	0.633333	1.233333

Hình 16: Bảng dữ liệu mới sau quá trình xử lý dữ liệu ban đầu

3.1.3 Training Model

Nhóm chúng tôi đã áp dụng các mô hình được nêu ở Mục 2.5 cho việc dự đoán kết quả trận đấu bóng đá. Cụ thể mã nguồn của từng mô hình được trình bày trong Hình 17.

3.2 Results

Chúng tôi đã sử dụng các mô hình đã nêu ở trên để thực nghiệm việc dự đoán dựa trên tập dữ liệu train là từ mùa 2018 đến hết mùa 2023 và tập dữ liệu để test sẽ là mùa 2024. Cụ thể kết quả được lưu như Hình 18.

Như có thể thấy trong hình, mô hình Gradient Boosting Classifier cho ra kết quả cao nhất với độ chính xác là 52%. Các mô hình khác cho kết quả dao động từ 47% đến 50%. Chúng tôi đã sử dụng phương thức classification_report để lấy chi tiết kết quả dự đoán của các mô hình trên và cho ra kết quả như Hình. Có thể thấy, các mô hình khác ngoại trừ Gradient Boosting Classifier đều có tỷ lệ đoán đúng các trận hòa là 0%, điều này dẫn đến kết quả thấp của các mô hình này.

```

def RandomForestClassifierPredictions(data, predictors, train, test):
    rf.fit(train[predictors], train["target"])
    preds = rf.predict(test[predictors])
    combined = pd.DataFrame(dict(actual=test["target"], predicted=preds), index=test.index)
    accuracy = classification_report(test["target"], preds)
    return combined, accuracy

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
model = LogisticRegression(random_state=100, max_iter=1000)
def LogisticRegressionPredictions(data, predictors, train, test):
    model.fit(train[predictors], train["target"])
    preds = model.predict(test[predictors])
    accuracy = classification_report(test["target"], preds)
    combined = pd.DataFrame(dict(actual=test["target"], predicted=preds), index=test.index)
    return combined, accuracy

from sklearn.ensemble import GradientBoostingClassifier
GBC_model = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1, random_state=0)
def GradientBoostingClassifierPredictions(data, predictors, train, test):
    GBC_model.fit(train[predictors], train["target"])
    preds = GBC_model.predict(test[predictors])
    accuracy = classification_report(test["target"], preds)
    combined = pd.DataFrame(dict(actual=test["target"], predicted=preds), index=test.index)
    return combined, accuracy

from sklearn.svm import SVC
SVC_model = SVC(kernel = 'rbf', random_state = 0)
def SVCPredictions(data, predictors, train, test):
    SVC_model.fit(train[predictors], train["target"])
    preds = SVC_model.predict(test[predictors])
    accuracy = classification_report(test["target"], preds)
    combined = pd.DataFrame(dict(actual=test["target"], predicted=preds), index=test.index)
    return combined, accuracy

```

Hình 17: Mã nguồn của các mô hình dự đoán đã được xây dựng

Do đó, để cải thiện độ chính xác cho các mô hình, có thể chỉnh sửa lại mục tiêu dự đoán là thắng hoặc không thắng tương ứng với các giá trị 2 và 0. Có thể thấy, độ chính xác của các mô hình đều được cải thiện với accuracy đều trên mức 0.6, tuy nhiên mô hình Gradient Boost vẫn đạt mức cao nhất là 0.67. Vì vậy kể từ phần này, nhóm sẽ tập trung chủ yếu vào mô hình này. Chi tiết được chỉ ra trong Hình 20.

Một điều đáng chú ý khác đó là khi áp dụng mô hình học cho từng đội (với target là Thắng/Hòa/Thua thì kết quả vẫn được cải thiện. Dưới đây là một số đánh giá cho kết quả của từng đội:

```

RandomForestClassifierPredictions:
0.5091678420310296
LogisticRegressionPredictions:
0.4936530324400564
GradientBoostingClassifierPredictions:
0.5289139633286318
SVCPredictions:
0.4781382228490832

```

Hình 18: Kết quả dự đoán ban đầu

RandomForestClassifierPredictions:					GradientBoostingClassifierPredictions:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.51	0.66	0.58	273	0	0.55	0.67	0.60	273
1	0.00	0.00	0.00	160	1	0.18	0.03	0.04	160
2	0.51	0.66	0.57	276	2	0.53	0.68	0.60	276
accuracy			0.51	709	accuracy			0.53	709
macro avg	0.34	0.44	0.38	709	macro avg	0.42	0.46	0.41	709
weighted avg	0.39	0.51	0.44	709	weighted avg	0.46	0.53	0.47	709

LogisticRegressionPredictions:					SVCPredictions:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.49	0.61	0.55	273	0	0.46	0.66	0.54	273
1	0.00	0.00	0.00	160	1	0.00	0.00	0.00	160
2	0.49	0.66	0.57	276	2	0.50	0.58	0.53	276
accuracy			0.49	709	accuracy			0.48	709
macro avg	0.33	0.42	0.37	709	macro avg	0.32	0.41	0.36	709
weighted avg	0.38	0.49	0.43	709	weighted avg	0.37	0.48	0.42	709

Hình 19: Chi tiết kết quả ban đầu

RandomForestClassifierPredictions:					GradientBoostingClassifierPredictions:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.67	0.87	0.76	433	0	0.69	0.83	0.75	433
2	0.62	0.34	0.44	276	2	0.61	0.42	0.49	276
accuracy			0.66	709	accuracy			0.67	709
macro avg	0.65	0.60	0.60	709	macro avg	0.65	0.62	0.62	709
weighted avg	0.65	0.66	0.63	709	weighted avg	0.66	0.67	0.65	709

LogisticRegressionPredictions:					SVCPredictions:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.68	0.86	0.76	433	0	0.63	0.93	0.75	433
2	0.61	0.36	0.45	276	2	0.59	0.15	0.24	276
accuracy			0.66	709	accuracy			0.63	709
macro avg	0.64	0.61	0.60	709	macro avg	0.61	0.54	0.50	709
weighted avg	0.65	0.66	0.64	709	weighted avg	0.61	0.63	0.55	709

Hình 20: Kết quả khi chỉ dự đoán thắng hoặc không thắng

Arsenal:					Liverpool:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.00	0.00	0.00	5	0	0.00	0.00	0.00	4
1	0.00	0.00	0.00	5	1	0.50	0.11	0.18	9
2	0.74	0.96	0.83	26	2	0.68	0.91	0.78	23
accuracy			0.69	36	accuracy			0.61	36
macro avg	0.25	0.32	0.28	36	macro avg	0.39	0.34	0.32	36
weighted avg	0.53	0.69	0.60	36	weighted avg	0.56	0.61	0.54	36
Manchester City:					Sheffield United:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.00	0.00	0.00	3	0	0.73	0.92	0.81	26
1	0.00	0.00	0.00	7	1	0.00	0.00	0.00	7
2	0.68	0.84	0.75	25	2	0.00	0.00	0.00	3
accuracy			0.60	35	accuracy			0.67	36
macro avg	0.23	0.28	0.25	35	macro avg	0.24	0.31	0.27	36
weighted avg	0.48	0.60	0.54	35	weighted avg	0.53	0.67	0.59	36
Aston Villa:					Manchester United:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.40	0.67	0.50	9	0	0.38	0.25	0.30	12
1	1.00	0.14	0.25	7	1	0.00	0.00	0.00	6
2	0.65	0.65	0.65	20	2	0.54	0.88	0.67	16
accuracy			0.56	36	accuracy			0.50	34
macro avg	0.68	0.49	0.47	36	macro avg	0.30	0.38	0.32	34
weighted avg	0.66	0.56	0.53	36	weighted avg	0.39	0.50	0.42	34

Hình 21: Kết quả khi dự đoán cho từng đội bóng

	actual	predicted	date	team	opponent	result
3108	2	2	2023-08-11	Manchester City	Burnley	W
3109	2	2	2023-08-19	Manchester City	Newcastle Utd	W
3110	2	2	2023-08-27	Manchester City	Sheffield Utd	W
3111	2	2	2023-09-02	Manchester City	Fulham	W
3112	2	2	2023-09-16	Manchester City	West Ham	W
3113	2	2	2023-09-23	Manchester City	Nott'ham Forest	W
3114	0	2	2023-09-30	Manchester City	Wolves	L
3115	0	2	2023-10-08	Manchester City	Arsenal	L
3116	2	2	2023-10-21	Manchester City	Brighton	W
3117	2	2	2023-10-29	Manchester City	Manchester Utd	W
3118	2	2	2023-11-04	Manchester City	Bournemouth	W
3119	1	2	2023-11-12	Manchester City	Chelsea	D
3120	1	2	2023-11-25	Manchester City	Liverpool	D
3121	1	2	2023-12-03	Manchester City	Tottenham	D
3122	0	2	2023-12-06	Manchester City	Aston Villa	L
3123	2	0	2023-12-10	Manchester City	Luton Town	W
3124	1	2	2023-12-16	Manchester City	Crystal Palace	D
3125	2	2	2023-12-27	Manchester City	Everton	W
3126	2	2	2023-12-30	Manchester City	Sheffield Utd	W
3127	2	2	2024-01-13	Manchester City	Newcastle Utd	W
3128	2	2	2024-01-31	Manchester City	Burnley	W
3129	2	2	2024-02-05	Manchester City	Brentford	W
3130	2	2	2024-02-10	Manchester City	Everton	W
3131	1	2	2024-02-17	Manchester City	Chelsea	D
3132	2	1	2024-02-20	Manchester City	Brentford	W

Hình 22: Kết quả khi dự đoán cho từng đội bóng

	actual	predicted	date	team	opponent	result
225	2	2	2023-08-12	Arsenal	Nott'ham Forest	W
226	2	2	2023-08-21	Arsenal	Crystal Palace	W
227	1	2	2023-08-26	Arsenal	Fulham	D
228	2	2	2023-09-03	Arsenal	Manchester Utd	W
229	2	2	2023-09-17	Arsenal	Everton	W
...
5168	1	0	2024-04-13	Wolverhampton Wanderers	Nott'ham Forest	D
5169	0	0	2024-04-20	Wolverhampton Wanderers	Arsenal	L
5170	0	2	2024-04-24	Wolverhampton Wanderers	Bournemouth	L
5171	2	0	2024-04-27	Wolverhampton Wanderers	Luton Town	W
5172	0	0	2024-05-04	Wolverhampton Wanderers	Manchester City	L

709 rows × 6 columns

Hình 23: Kết quả khi dự đoán cho từng đội bóng