**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

Ảnh có chứa văn bản, mẫu họa

Mô tả được tạo tự động**KHOA CÔNG NGHỆ THÔNG TIN**

**NGUYỄN TRÍ DŨNG: 20133029**

**NGUYỄN KHOA QUANG THẮNG: 20133090**

Đề tài:

**XÂY DỰNG NỀN TẢNG THỐNG KÊ**

**VÀ DỰ ĐOÁN CÁC TRẬN ĐẤU BÓNG ĐÁ**

**TIỂU LUẬN CHUYÊN NGÀNH KTDL**

**GIÁO VIÊN HƯỚNG DẪN**

**ThS. LÊ MINH TÂN**

**KHÓA 2020 – 2024**

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |

# PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1 :........................ MSSV 1:

Họ và tên Sinh viên 2 :........................ MSSV 2:

Ngành: Công nghệ Thông tin

Tên đề tài:

Họ và tên Giáo viên hướng dẫn:

## *NHẬN XÉT*

## *Về nội dung đề tài & khối lượng thực hiện:*

1. Ưu điểm:

1. Khuyết điểm

1. Đề nghị cho bảo vệ hay không ?
2. Đánh giá loại :
3. Điểm :

Tp*. Hồ Chí Minh, ngày tháng năm 201*

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |

# PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1: ........................ MSSV 1:

Họ và tên Sinh viên 2: ........................ MSSV 2:

Ngành: Công nghệ Thông tin

Tên đề tài:

Họ và tên Giáo viên phản biện:

## *NHẬN XÉT*

## *Về nội dung đề tài & khối lượng thực hiện:*

1. Ưu điểm:

1. Khuyết điểm

1. Đề nghị cho bảo vệ hay không ?
2. Đánh giá loại :
3. Điểm :

Tp*. Hồ Chí Minh, ngày tháng năm 201*

Giáo viên phản biện

*(Ký & ghi rõ họ tên)*

# LỜI CAM ĐOAN

Tiểu luận này là công trình nghiên cứu của chúng em, được thực hiện dưới sự hướng dẫn khoa học của thầy Lê Minh Tân. Các số liệu, những kết luận nghiên cứu và sản phẩm được tạo ra bởi chúng em được trình bày trong khóa luận này là trung thực.

Chúng em xin hoàn toàn chịu trách nhiệm về lời cam đoan này.

|  |  |
| --- | --- |
| Sinh viên thực hiện  (Ký và ghi rõ họ tên) | Sinh viên thực hiện  (Ký và ghi rõ họ tên) |
| **Nguyễn Trí Dũng** | **Nguyễn Khoa Quang Thắng** |

Lời cảm ơn

*Lời đầu tiên, nhóm xin được gửi lời cảm ơn đặc biệt đến Thầy -Th.S Lê Minh Tân - Giảng viên hướng dẫn đề tài Tiểu luận Chuyên ngành – trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.*

*Trong thời gian nhóm làm đề tài, nhóm đã nhận được nhiều sự giúp đỡ từ thầy. Thầy đã cung cấp đầy đủ kiến thức, chỉ bảo và đóng góp những ý kiến quý báu giúp nhóm có thể hoàn thành được đề tài theo một cách hoàn thiện nhất.*

*Xuất phát từ mục đích học tập, tìm hiểu kỹ hơn các kiến thức về hệ thống thông tin, cũng như tìm hiểu sâu hơn về quy trình nghiệp vụ của lên ý tưởng, xây dựng mô hình dự đoán. Nhóm đã thực hiện đề tài “Xây dựng nền tảng thống kê và dự đoán các trận đấu bóng đá”. Trong quá trình thực hiện đề tài, dựa trên những kiến thức được thầy truyền đạt, kết hợp với việc tự tìm hiểu những công nghệ và những kiến thức mới, nhóm đã cố gắng thực hiện đề tài một cách tốt nhất. Không thể tránh khỏi những thiếu sót trong quá trình thực hiện cũng như kết quả thực hiện có thể chưa đạt mức tối ưu cao nhất, đó là thách thức mà nhóm phải thực hiện.*

*Nhóm rất mong nhận được sự góp ý từ thầy nhằm rút ra những kinh nghiệm quý báu và hoàn thiện vốn kiến thức để nhóm có thể hoàn thành những đề tài tốt hơn trong tương lai.*

*Nhóm xin chân thành cảm ơn thầy!*

**Trường Đại học Sư Phạm Kỹ Thuật TP.HCM**

**Khoa : Công nghệ thông tin**

**ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH**

Họ và tên Sinh viên 1: Nguyễn Trí Dũng MSSV 1: **20133029**

Họ và tên Sinh viên 2: Nguyễn Khoa Quang Thắng MSSV 2: **20133090**

Chuyên ngành: **Kỹ thuật dữ liệu**

Tên đề tài: Xây dựng nền tảng thống kê và dự đoán các trận đấu bóng đá

Họ và tên giáo viên hướng dẫn: **ThS. Lê Minh Tân**

**Nhiệm vụ của tiểu luận chuyên ngành:**

* Tìm hiểu các mô hình học máy, các phương pháp đánh giá mô hình.
* Tìm hiểu các công cụ và thư viện để xây dựng nên một hệ thống truyền dữ liệu.
* Tìm hiểu các thư viện hỗ trợ trong  bài toán để cung cấp các hàm và mô hình để sử dụng trong bài toán
* Cài đặt demo ứng dụng
* Đánh giá bằng thực nghiệm

# Kế hoạch thực hiện

|  |  |  |  |
| --- | --- | --- | --- |
|  | Kế hoạch trong tuần | Ngày thực hiện | Công việc |
| Tuần 1 | Lên kế hoạch thực hiện đề tài TLCN | 25/8/2023 | Lựa chọn mục tiêu, các công nghệ sử dụng cho đề tài |
| Tuần 2 | Tìm hiểu, tham khảo các open source code, tìm hiểu các data có sẵn | 30/8/2023 | Tìm hiểu các data bóng đá có sẵn trên github, kaggle |
|  |  | 31/8/2023 | Tìm hiểu thuật toán dự đoán kết quả |
| Tuần 3 | Thực hiện cào dữ liệu từ FBref | 9/9/2023 | Cài đặt Python Scrapy, demo các tính năng |
| Tuần 4 | Tiếp tục cào dữ liệu từ FBref, tìm hiểu các thuộc tính | 15/6/2023 | Cào các thuộc tính với trang web tĩnh |
|  |  | 16/6/2023 | Cào với điều kiện trang web có JavaScripts |
| Tuần 5 | Đánh trọng số từ các thuộc tính trận đấu | 20/9/2023 | Đánh trọng số từ 1-5 cho các thuộc tính có ảnh hưởng đến kết quả trận đấu |
| Tuần 6 | Cào các thông tin và thuộc tính các cầu thủ trong Sofifa | 26/9/2023 | Cào các cầu thủ trong cùng một phiên bản |
|  |  | 27/9/2023 | Cào các cầu thủ trong cùng một giải đấu, có phân trang |
| Tuần 7 | Tiếp tục cào tất cả các cầu thủ. Bước đầu phân tích dữ liệu về các trận đấu | 3/10/2023 | Treo máy cào các cầu thủ (mất khoảng 3 ngày) |
|  |  | 5/10/2023 | Làm sạch dữ liệu các trận đấu: đổi tên, đổi kiểu dữ liệu. |
| Tuần 8 | Phân tích dữ liệu các trận đấu và cầu thủ | 14/10/2023 | Kết hợp các cầu thủ trong đội hình với các cầu thủ trong Sofifa |
|  |  | 15/10/2023 | Thêm các giá trị bổ sung cho các cầu thủ trong FBref |
| Tuần 9 | Tạo các thông số các trận đấu dùng cho mô hình dự đoán | 20/10/2023 | Đổi thông tin về các thông số diễn biến trận đấu với thông số của 2 trận trước |
|  |  | 21/10/2023 | Kết hợp các thuộc tính từ nhiều bảng thành một bảng duy nhất với các input, output dùng cho mô hình dự đoán |
| Tuần 10 | Xây dựng mô hình dự đoán sử dụng các thuật toán phân loại | 3/11/2023 | Phân tích dữ liệu, tạo các train, test data, sử dụng các phương pháp chuẩn hóa dữ liệu |
|  |  | 4/11/2023 | Xây dựng mô hình dự đoán với 3 thuật toán Logistic Regression, NaiveBayes, Decision Tree, Random Forest |
| Tuần 11 | Đánh giá các mô hình với Accuracy, F1-Score, Precision, Recall. Chọn mô hình tốt nhất | 5/11/2023 | Xây dựng mô hình huấn luyện và đánh giá qua nhiều lần thử nghiệm |
| Tuần 12 | Xây dựng mô hình tối ưu hóa đội hình | 11/12/2023 | Sử dụng mô hình hồi qui tuyến tính nguyên hỗn hợp kết hợp dữ liệu Sofifa và FBref |
| Tuần 13 | Cào các trận đấu mới. Truyền dẫn cho Kafka | 18/11/2023 | Cào các trận đấu mới với Sportgambler với đội hình dự kiến |
|  |  | 19/11/2023 | Đẩy thông tin cho Kafka, tạo consumer lấy và xử lý dữ liệu |
| Tuần 14 | Tạo các schema cho MySQL, thực hiện đẩy dữ liệu vào | 25/11/2023 | Tạo các schema |
|  |  | 26/11/2023 | Đẩy dữ liệu từ các file csv vào hệ quản trị cơ sở dữ liệu MySQL |
| Tuần 15 | Xây dựng dashboard | 2/12/2023 | Tạo các API Django, kết nối MySQL trả về JSON |
|  |  | 3/12/2023 | Xây dựng giao diện với AngularJS |
| Tuần 16 | Tiếp tục hoàn thiện các thành phần hệ thống thông tin | 10/12/2023 | Xử lý dữ liệu từ tầng thu thập đến tầng truyền tin cho đến tầng lưu trữ, tầng truy cập |
| Tuần 17 | Tiếp tục hoàn thiện các giao diện cho người dùng | 16/12/2023 | Hoàn thiện giao diện lịch sử các trận đấu, chi tiết một trận đấu, các thông tin về các trận đấu sắp diễn ra |
| Tuần 18 | Hoàn thiện báo cáo, source code | 19/12/2023 | Hoàn thiện các quy trình đi từ dữ liệu các trận đấu thô cho đến kết quả đã được phân tích cho người dùng |
| Tuần 18 | Hoàn thiện báo cáo, source code | 20/12/2023 | Chỉnh sửa, bổ sung, hoàn thiện báo cáo |

**MỤC LỤC**

[PHẦN 1 – GIỚI THIỆU CHUNG 1](#_Toc154267046)

[1.1. Tính cấp thiết của đề tài 1](#_Toc154267047)

[1.1.1. Mục đích nghiên cứu………………………………………………………….1](#_Toc154267048)

[1.1.2. Đối tượng và phạm vi nghiên cứu…………………………………………….3](#_Toc154267049)

[1.2. Ý nghĩa khoa học và thực tiễn 4](#_Toc154267050)

[1.3. Phương pháp tiếp cận 4](#_Toc154267051)

[PHẦN 2: CƠ SỞ LÝ THUYẾT 7](#_Toc154267052)

[2.1. Học máy – Machine learning: 7](#_Toc154267053)

[2.1.1. Các thuật toán phân loại – Classification:…………………………………….7](#_Toc154267054)

[2.2. Mô hình lập trình tuyến tính nguyên hỗn hợp – Mixed-Integer Linear Programming Model 13](#_Toc154267055)

[2.3. Các thuật toán chuẩn hoá – Regularization Algorithms 13](#_Toc154267056)

[2.3.1. Hồi quy Ridge…………………………………………………………….....14](#_Toc154267057)

[2.3.2. Hồi quy Lasso……………………………………………………………….14](#_Toc154267058)

[2.3.3. Hồi quy ElasticNet………………………………………………………….15](#_Toc154267059)

[PHẦN 3 – GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG 16](#_Toc154267060)

[3.1. Các công nghệ sử dụng 16](#_Toc154267061)

[3.1.1. Ngôn ngữ lập trình, công cụ hỗ trợ…………………………………………16](#_Toc154267062)

[3.1.2. Các hệ thống, framework, platform…………………………………………17](#_Toc154267063)

[3.2. Môi trường thực nghiệm 26](#_Toc154267064)

[3.2.1. Cấu hình chung……………………………………………………………...26](#_Toc154267065)

[3.2.2. Hoạt động các ứng dụng…………………………………………………….26](#_Toc154267066)

[PHẦN 4 – MÔ TẢ TẬP DỮ LIỆU 28](#_Toc154267067)

[4.1. Nguồn thu thập 28](#_Toc154267068)

[4.2. Mô tả chi tiết tập dữ liệu 30](#_Toc154267069)

[4.2.1. Trận đấu……………………………………………………………………..30](#_Toc154267070)

[4.2.2. Cầu thủ………………………………………………………………………36](#_Toc154267071)

[PHẦN 5: XÂY DỰNG HỆ THỐNG 41](#_Toc154267072)

[STREAMING DATA 41](#_Toc154267073)

[5.1. Tầng thu thập – Collection Layer: 41](#_Toc154267074)

[5.1.1. Thu thập các trận đấu với crawlMatches.py: ……………………………….41](#_Toc154267075)

[5.1.2. Thu thập các cầu thủ với crawlPlayers.py: …………………………………49](#_Toc154267076)

[5.1.3. Thu thập các trận đấu tương lai với crawlUpComingMatch.py:……………55](#_Toc154267077)

[5.2. Tầng hàng đợi thông tin - Message Queuing Layer với Apache Kafka: 56](#_Toc154267078)

[5.2.1. Sử dụng Apache Kafka để thu thập trận đấu mới:…………………………..56](#_Toc154267079)

[5.3. Tầng phân tích – Analysis Layer với Apache Spark và Pandas: 58](#_Toc154267080)

[5.3.1. Chuẩn bị dữ liệu: ……………………………………………………………58](#_Toc154267081)

[5.3.2. Phân tích các thông số cần dùng cho thống kê:……………………………..63](#_Toc154267082)

[5.3.3. Kết hợp các thông số sử dụng cho model:…………………………………..71](#_Toc154267083)

[5.4. Tầng lưu trữ – Storage Layer với MySQL: 81](#_Toc154267084)

[5.5. Tầng truy cập – Data Access Layer với Python Django và AngularJS: 83](#_Toc154267085)

[5.5.1. Xây dựng nền tảng truy cập dữ liệu với Python Django:……………………83](#_Toc154267086)

[5.5.2. Xây dựng trang theo dõi và thống kê với AngularJS:……………………….86](#_Toc154267087)

[PHẦN 6: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN 93](#_Toc154267088)

[CÁC TRẬN ĐẤU VÀ MÔ HÌNH TỐI ƯU HÓA 93](#_Toc154267089)

[ĐỘI HÌNH 93](#_Toc154267090)

[6.1. Mô hình dự đoán các trận đấu 93](#_Toc154267091)

[6.1.1. Phân tích tập dữ liệu………………………………………………………...93](#_Toc154267092)

[6.1.2. Sử dụng các thuật toán phân loại trong Spark Machine Learning:…………94](#_Toc154267093)

[6.2. Mô hình tối ưu hóa đội hình 100](#_Toc154267094)

[6.2.1. Phân tích tập dữ liệu………………………………………………………100](#_Toc154267095)

[6.2.2. Xây dựng mô hình tối ưu hóa đội hình dựa vào các thống số cầu thủ:…….102](#_Toc154267096)

[PHẦN 7: ĐÁNH GIÁ MÔ HÌNH BẰNG THỰC NGHIỆM 106](#_Toc154267097)

[7.1. Các độ đo đánh giá mô hình 106](#_Toc154267098)

[7.1.1. Độ chính xác - Accuracy…………………………………………………106](#_Toc154267099)

[7.1.2. Độ đúng đắn - Precision……………………………………………………106](#_Toc154267100)

[7.1.3. Recall………………………………………………………………………106](#_Toc154267101)

[7.1.4. F1-Score……………………………………………………………………106](#_Toc154267102)

[7.1.5. Weighted recall: …………………………………………………………..107](#_Toc154267103)

[7.2. Các phương pháp đánh giá mô hình 107](#_Toc154267104)

[7.2.1. Kiểm chứng chéo – Cross Validation: ……………………………………107](#_Toc154267105)

[7.3. Các trường hợp thực nghiệm 108](#_Toc154267106)

[7.3.1. Đánh giá, nhận xét và so sánh sau khi sử dụng nhiều mô hình……………108](#_Toc154267107)

[7.3.2. Đánh giá và nhận xét mô hình khi sử dụng phương pháp kiểm chứng chéo108](#_Toc154267108)

[7.3.3. Đánh giá độ ảnh hưởng của các siêu tham số sau khi tối ưu mô hình:…….108](#_Toc154267109)

[7.4. Kết quả thực nghiệm: 109](#_Toc154267110)

[7.4.1. Đánh giá, nhận xét và so sánh sau khi sử dụng nhiều mô hình kết hợp sử dụng phương pháp kiểm chứng chéo…………………………………………………...109](#_Toc154267111)

[7.4.3. Đánh giá, nhận xét độ ảnh hưởng của các siêu tham số sau khi tối ưu mô hình: ……………………………………………………………………………...113](#_Toc154267112)

[PHẦN 8: KẾT LUẬN 114](#_Toc154267113)

[8.1. Kết quả 114](#_Toc154267114)

[8.1.1. Về phần thu thập dữ liệu…………………………………………………...114](#_Toc154267115)

[8.1.2. Về phần mô hình dự đoán………………………………………………….114](#_Toc154267116)

[8.2. Hạn chế, khó khăn 114](#_Toc154267117)

[8.1.1. Về phần thu thập dữ liệu…………………………………………………...114](#_Toc154267118)

[8.1.2. Về phần mô hình dự đoán………………………………………………….115](#_Toc154267119)

[8.3. Hướng phát triển 115](#_Toc154267120)

[8.3.1. Về tương lai gần……………………………………………………………115](#_Toc154267121)

[8.3.2. Về tương lai xa……………………………………………………………..115](#_Toc154267122)

[TÀI LIỆU THAM KHẢO, PHỤ LỤC 116](#_Toc154267123)

# Danh mục bảng biểu

Biểu đồ 1.1: Quy trình chung hệ thống streaming data

Bàng 4.1: Mô tả MatchInfos trong tập dữ liệu

Bàng 4.2: Mô tả MatchSquad trong tập dữ liệu

Bàng 4.3: Mô tả MatchGoals trong tập dữ liệu

Bàng 4.4: Mô tả MatchStats trong tập dữ liệu

Bàng 4.5: Mô tả SofifaInfos trong tập dữ liệu

Bàng 4.6: Mô tả SofifaAttr trong tập dữ liệu

# Danh mục hình minh họa

Hình 1.1: Trang báo VnExpress tạo cuộc bình chọn dự đoán tỷ số cho một trận đấu và đưa ra thống kê, phong độ.

Hình 1.2: Trang báo VnExpress đưa ra thống kê phần trăm dự đoán sau khi người dùng bầu chọn.

Hình 1.3: Quy trình chung Hệ thống Streaming Data về bóng đá

Hình 2.1: Hàm mất mát của Ridge Regression

Hình 2.2: Hàm mất mát của Hồi quy Lasso

Hình 4.1: Một góc các tính năng có trong FBRef bao gồm players, clubs, competitions, countries, matches,…

Hình 4.2: Một góc các tính năng trong Sofifa bao gồm player, teams, squads qua các mốc thời gian và phiên bản

Hình 4.3: Sportsgambler.com đưa ra đội hình và các cầu thủ xuất phát của một trận đấu chưa diễn ra

Hình 5.1: Đối tượng Player và các hàm hỗ trợ

Hình 5.2: Cấu hình Scrapy của tệp crawlMatches.py

Hình 5.3: Hàm giúp Scrapy truy cập tới Fbref

Hình 5.4: Hàm giúp Scrapy truy cập tới Fbref theo mùa giải

Hình 5.5: Hàm giúp Scrapy truy cập tới Fbref theo các trận đấu

Hình 5.6: Hàm giúp Scrapy khởi tạo các đối tượng

Hình 5.7: Đoạn mã giúp Scrapy lấy match\_id

Hình 5.8: Đoạn mã giúp Scrapy lấy đường dẫn và mùa giải

Hình 5.9: Đoạn mã giúp Scrapy lấy tên của mùa giải

Hình 5.10: Đoạn mã giúp Scrapy lấy ngày trận đấu diễn ra

Hình 5.11: Đoạn mã giúp Scrapy lấy tuần trận đấu diễn ra

Hình 5.12: Đoạn mã giúp Scrapy lấy tên của hai đội nhà và đội khách

Hình 5.13: Đoạn mã giúp Scrapy phân biệt đội nhà và đội khách

Hình 5.14: Đoạn mã giúp Scrapy lấy tỉ số

Hình 5.15: Đoạn mã giúp Scrapy lấy tên của huấn luyện viên của hai đội

Hình 5.16: Đoạn mã giúp Scrapy lấy tên của đội trưởng của hai đội

Hình 5.17: Đoạn mã giúp Scrapy lấy các thông tin giờ diễn ra trận đấu, số lượng khán giả, …

Hình 5.18: Đoạn mã giúp Scrapy lấy đội hình của cả hai đội

Hình 5.19: Đoạn mã giúp Scrapy lấy các thông tin trong trận đấu của cả hai đội

Hình 5.20: Đoạn mã giúp Scrapy trả về kết quả

Hình 5.21: Đoạn mã giúp Scrapy lấy tên, số áo, và biến xác định có phải là cầu thủ dự bị của các cầu thủ

Hình 5.22: Đoạn mã giúp Scrapy lấy thông tin bàn thắng

Hình 5.23: Đoạn mã giúp Scrapy lấy

Hình 5.24: Đoạn mã nhập thư viện sử dụng trong tệp crawlPlayers.py

Hình 5.25: Cấu hình Scrapy cơ bản

Hình 5.26: Đoạn mã giúp Scrapy truy cập tới Sofifa

Hình 5.27: Đoạn mã giúp Scrapy truy cập Sofifa theo mùa giải

Hình 5.28: Đoạn mã giúp Scrapy truy cập theo các phiên bản và các bản cập nhật Fifa

Hình 5.29: Đoạn mã giúp Scrapy truy cập vào các trang danh sách cầu thủ

Hình 5.30: Đoạn mã giúp Scrapy truy cập vào các trang danh sách cầu thủ

Hình 5.31: Đoạn mã giúp Scrapy lấy ngày cập nhật và các vị trí của cầu thủ

Hình 5.32: Đoạn mã giúp Scrapy lấy thông tin cá nhân của cầu thủ

Hình 5.33: Đoạn mã giúp Scrapy lấy đặc điểm riêng của cầu thủ

Hình 5.34: Đoạn mã giúp Scrapy lấy câu lạc bộ hoặc hoặc đội tuyển của cầu thủ

Hình 5.35: Đoạn mã giúp Scrapy lấy chỉ số tấn công

Hình 5.36: Đoạn mã giúp Scrapy lấy chỉ số kĩ năng

Hình 5.37: Đoạn mã giúp Scrapy lấy chỉ số chuyển động

Hình 5.38: Đoạn mã giúp Scrapy lấy chỉ số sức mạnh

Hình 5.39: Đoạn mã giúp Scrapy lấy chỉ số phòng thủ

Hình 5.40: Đoạn mã giúp Scrapy lấy chỉ số thủ môn

Hình 5.41: Đoạn mã giúp Scrapy lấy kĩ năng

Hình 5.42: Các thư viện sử dụng cho tệp crawlUpComingMatch.py

Hình 5.43: Đoạn mã giúp Scrapy lấy các thông tin của trận đấu dự kiến

Hình 5.44: Gửi dữ liệu cho Kafka

Hình 5.45: Nhận dữ liệu từ Kafka

Hình 5.46: Xuất dữ liệu từ Kafka ra CSV

Hình 5.47: Đường dẫn của tệp prepare.ipynb

Hình 5.48: Nhập thư viện cho tệp prepare.ipynb

Hình 5.49: Các biến dataframe

Hình 5.50: Thay đổi tên cột cho fbref\_matchinfos

Hình 5.51: Chọn ra các thuộc tính cần thiết cho fbref\_matchinfos

Hình 5.52: Ép kiểu cho fbref\_matchinfos

Hình 5.53: Fbref\_Infos sau quá trình chuẩn bị dữ liệu

Hình 5.54: Tạo dataframe mới

Hình 5.55: Hình ảnh sau khi chuẩn hoá dữ liệu

Hình 5.56: Thay đổi tên cột cho sofifa\_players\_infos

Hình 5.57: Dữ liệu sau khi được chuẩn hoá

Hình 5.58: Chọn ra các thuộc tính cần thiết

Hình 5.59: Dữ liệu sau khi được chọn lọc và sắp xếp

Hình 5.60: Ép kiểu các thuộc tính trong sofifa\_players\_infos

Hình 5.61: Thay đổi tên thuộc tính sofifa\_players\_attr

Hình 5.62: Hình ảnh sau khi thay đổi tên thuộc tính

Hình 5.63: Ép kiểu thuộc tính sofifa\_players\_attr

Hình 5.64: Xuất kết quả ra CSV

Hình 5.65: Kết quả các tệp CSV liên quan tới Fbref

Hình 5.66: Kết quả các tệp CSV liên quan tới Sofifa

Hình 5.67: Nhập thư viện

Hình 5.68: Tạo dataframe mới

Hình 5.69: Đoạn mã giúp tìm ra các biến chưa khớp

Hình 5.70: Thay thế các biến chưa khớp

Hình 5.71: Sắp xếp theo thời gian

Hình 5.72: Hàm tìm ra ngày cập nhật gần nhất

Hình 5.73: Đoạn mã giúp tìm cầu thủ tương ứng

Hình 5.74: Hàm xử lí những cầu thủ chưa khớp

Hình 5.75: Tạo ra các danh sách và dataframe để lưu trữ

Hình 5.76: Đoạn mã giúp tìm cầu thủ tương ứng

Hình 5.77: Hiển thị kết quả

Hình 5.78: Đoạn mã tìm các cầu thủ trùng tên trong đội

Hình 5.79: Tạo DataFrame chứa cầu thủ bị thiếu thông tin.

Hình 5.80: Bổ sung các cầu thủ bị thiếu chỉ số

Hình 5.81: Hiển thị kết quả sau khi xử lí

Hình 5.82: Bổ sung các giá trị bị thiếu của cầu thủ

Hình 5.83: Xuất kết quả ra CSV

Hình 5.84: Hình ảnh tệp kết quả CSV

Hình 5.85: Kết hợp các thông tin cầu thủ trong Sofifa với các cầu thủ mỗi trận đấu trong FBref.

Hình 5.86: Nhập thư viện

Hình 5.87: Tạo dataframe

Hình 5.88: FBref\_MatchStats bao gồm các thông số dạng numeric mô tả thông tin tổng kết của hai đội bóng trong mỗi trận đấu.

Hình 5.89: Tách dữ liệu cho sân nhà và sân khách

Hình 5.90: Hình ảnh sau khi tách ra sân nhà

Hình 5.91: Hình ảnh sau khi tách sân khách

Hình 5.92: Đặt tên lại các thuộc tính

Hình 5.93: Liên kết tập dữ liệu đội nhà và đội khách

Hình 5.94: Sắp xếp các trận đấu theo thời gian giảm dần

Hình 5.95: Tính trung bình cộng của thông số của 2 trận đấu trước

Hình 5.96: Loại bỏ 2 trận đấu cũ nhất của mỗi đội

Hình 5.97: Loại bỏ ra khỏi tập dữ liệu

Hình 5.98: Đoạn mã giúp chia vị trí các cầu thủ

Hình 5.99: Tính sức mạnh các cầu thủ theo vị trí

Hình 5.100: Hình ảnh sau khi tính toán

Hình 5.101: Xuất ra kết quả

Hình 5.102: Tệp CSV kết quả

Hình 5.103: Thư viện sqlalchemy tạo engine kết nối MySQL và ThreadPoolExecutor tạo các luồng

Hình 5.104: Tạo kết nối MySQL với hostname, database, user và password

Hình 5.105: Lưu các đường dẫn các file csv

Hình 5.106: Hàm đọc các file csv và nhập vào các schema

Hình 5.107: Tạo các luồng thực thi

Hình 5.108: Tạo kết nối với hệ cơ sở dữ liệu MySQL

Hình 5.109: Các module (thành phần) trong settings.py

Hình 5.110: 2 ứng dụng chính của Django

Hình 5.111: Xử lý các yêu cầu cho các trận đấu trong view.py

Hình 5.112: Mô tả một model MatchGoals trong models.py

Hình 5.113: Thực hiện lấy thông tin của trận đấu

Hình 5.114: Các thành phần cấu chức năng

Hình 5.115: Đoạn mã JS gửi request yêu cầu Django trả về tất cả các trận đấu JSON

Hình 5.116: Đoạn mã xử lý dữ liệu JSON và xuất ra HTML

Hình 5.117: Giao diện trang chủ

Hình 5.118: Giao diện các giải đấu

Hình 5.119: Giao diện các trận đấu sắp diễn ra

Hình 5.119: Giao diện các trận đấu sắp diễn ra

Hình 5.120: Giao diện các trận đấu lịch sử

Hình 5.121: Giao diện các thông tin chi tiết của trận đấu lịch sử

Hình 5.122: Giao diện thống kê một trận đấu sắp diễn ra và đưa ra dự đoán cùng đội hình tối ưu

Hình 6.1: Mã thực thi đọc file csv

Hình 6.2: Thêm thuộc tính Result

Hình 6.3: Biểu đồ trực quan hóa chỉ số Attack của đội bóng

Hình 6.4: Các thư viện của PySpark dùng cho xử lý dữ liệu và Machine Learning

Hình 6.5: Mã thực thi phiên làm việc trên Spark

Hình 6.6: Lưu trữ các thuộc tính không cần dùng cho xây dựng mô hình dự đoán

Hình 6.7: Biến đổi các thuộc tính dạng string sang numeric

Hình 6.8: Mã thực thi chuyển đổi các thuộc tính theo phương pháp OneHotEncoding

Hình 6.9: Mã thực thi chuyển đổi biến dự đoán

Hình 6.10: Hàm loại bỏ 10 trận đấu mới nhất

Hình 6.11: Hàm trả về tập train và test

Hình 6.12: Hàm tính toán các phương pháp đánh giá mô hình

Hình 6.13: Hàm xây dựng mô hình dự đoán theo thuật toán Logistic Regression

Hình 6.14: Độ đo Accuracy sau mỗi lần huấn luyện, giảm tập dữ liệu

Hình 6.15: Hàm xây dựng mô hình dự đoán theo thuật toán Decision Tree

Hình 6.16: Hàm xây dựng mô hình dự đoán theo thuật toán NaiveBayes

Hình 6.17: Hàm xây dựng mô hình dự đoán theo thuật toán Random Forest

Hình 6.18: Các độ đo đánh giá mô hình theo các thuật toán

Hình 6.19: Thư viện pulp sử dụng cho bài toán tuyến tính nguyên hỗn hợp

Hình 6.20: Dữ liệu về các cầu thủ trong Sofifa

Hình 6.21: Lọc ra những cầu thủ trong cùng một phiên bản mới nhất

Hình 6.22: Xóa những cầu thủ nghỉ thi đấu hoặc chấn thương

Hình 6.23: Đổi giá trị Value cho cầu thủ

Hình 6.24: Chia các vị trí cầu thủ trên sân thành 4 nhóm

Hình 6.25: Lấy thông số Potential và Value của cầu thủ

Hình 6.26: Xây dựng mô hình tuyến tính nguyên hỗn hợp

Hình 6.27: Lấy dữ liệu các trận đấu trong mùa giải mới nhất

Hình 6.28: Thực thi tối ưu hóa các cầu thủ cho mỗi team

Hình 6.29: Kết quả các cấu thủ tối ưu cho mỗi đội bóng

Hình 6.30: Kết quả đội hình tối ưu cho mỗi đội bóng

Hình 7.1: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình hồi quy Logistic.

Hình 7.2: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình Cây quyết định.

Hình 7.3: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình Naïve Bayes.

Hình 7.4: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình Rừng ngẫu nhiên.

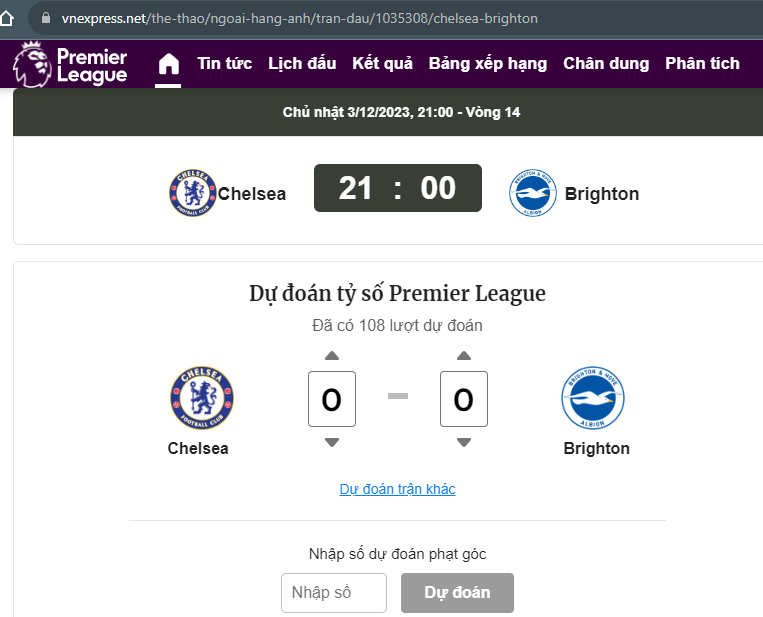
# PHẦN 1 – GIỚI THIỆU CHUNG

## 1.1. Tính cấp thiết của đề tài

### 1.1.1. Mục đích nghiên cứu

Hiện nay có rất nhiều nền tảng thống kê các tập dữ liệu ứng dụng cho thực tế ví dụ như tập dữ liệu về orders, bookings, cryptos, chứng khoán,… Việc thu thập các tập dữ liệu ấy rất hữu ích cho việc phân tích, nhằm đưa ra những định hướng về sau cho doanh nghiệp. Những gì chúng ta thường xuyên theo dõi ngày nay, ví dụ như giá vàng hay một biểu đồ thống kê về doanh số bán hàng của một công ty nào đó,… có thể gọi chung là một “dashboard”, nó là kết quả cuối cùng của một quy trình xử lý dữ liệu, thông tin. Một hệ thống thông tin như vậy tất nhiên cần phải đảm bảo tập dữ liệu được thu thập không được thiếu sót, tính bảo mật cho thông tin, tốc độ truyền tải tốt nhất tới cho người dùng,…

Các nền tảng thống kê tập dữ liệu về bóng đá cũng như vậy. Bóng đá là một môn thể thao rất phát triển do đó tập dữ liệu về bóng đá là rất lớn như các thông tin về giải đấu, trận đấu, cầu thủ,… Người dùng xem bóng đá thường xuyên theo dõi lịch thi đấu, thông số các trận đấu, chỉ số cầu thủ và thường có thói quen dự đoán kết quả của các trận đấu tiếp theo dựa theo tình hình hiện tại và phong độ gần đây của các đội bóng. Như VnExpress, một trang báo nổi bật và nhiều người xem nhất ở Việt Nam, ở danh mục Thể Thao, hàng tuần trang báo này đều đưa ra các cuộc bầu chọn cho người đọc dự đoán tỷ số các trận đấu sắp diễn ra – hình 10, bên cạnh đó là các thống kê, phong độ của 2 đội bóng đối đầu và ngoài ra, sau khi dự đoán thì người dùng có thể tham khảo thống kê phần trăm dự đoán của mọi người dùng khác – hình 11. Người dùng cũng dựa nhiều vào thông tin đội hình ra sân của mỗi đội để dự đoán các trận đấu. Nắm được những nhu cầu đó, nhóm muốn xây dựng một nền tảng dành cho người dùng có thể theo dõi, tìm kiếm những thông tin về thống kê tập dữ liệu bóng đá, đưa ra những kết quả phân tích về dự doán kết quả các trận đấu tương lai cùng với đội hình các cầu thủ tối ưu giúp cho người dùng có thể coi đó như là một công cụ tham khảo để đưa ra dự đoán.







Hình 1.1: Trang báo VnExpress tạo cuộc bình chọn dự đoán tỷ số cho một trận đấu và đưa ra thống kê, phong độ.



Hình 1.2: Trang báo VnExpress đưa ra thống kê phần trăm dự đoán sau khi người dùng bầu chọn.

Trước hết phải tìm hiểu, nghiên cứu các công nghệ sử dụng để xây dựng hệ thống thông tin, cùng với đó là áp dụng các thuật toán trong Machine Learning cũng như các thuật toán tối ưu hóa để xây dựng mô hình dự đoán. Đó chính là mục đích nghiên cứu chính của nhóm trong đề tài này.

### 1.1.2. Đối tượng và phạm vi nghiên cứu

Hệ thống streaming data, hệ thống gồm các quy trình thu thập thông tin, truyền dẫn thông tin, phân tích thông tin, lưu trữ và truy xuất thông tin. Là hệ thống phân tán hiện đại ứng dụng cho việc truyền dẫn thông tin, đặc biệt đáp ứng cho các thông tin trong thời gian thực.

Machine Learning – Học máy là phương pháp học hỏi dựa trên tập dữ liệu được huấn luyện, sử dụng các thuật toán và đưa ra kết quả dựa trên quá trình học hỏi đó. Việc áp dụng Machine Learning vào tập dữ liệu về bóng đá nhằm đưa ra dự đoán kết quả cho các trận đấu trong tương lai dựa vào các thông số của những trận đấu trong quá khứ.

Tập dữ liệu về bóng đá được thu thập từ các nguồn phân tích nghiệp vụ bóng đá, nhóm sẽ đề cập chi tiết hơn ở phần 3 – Mô tả tập dữ liệu.

## 1.2. Ý nghĩa khoa học và thực tiễn

Về mặt khoa học, đề tài sử dụng các công nghệ ứng dụng cho việc truyển tải thông tin phổ biến và thông dụng ngày nay như Apache Kafka, Apache Spark,… nên có thể triển khai cho nhiều hệ thống. Các thuật toán Machine Learning được sử dụng cho xây dựng các model cũng rất ứng dụng cho nhiều lĩnh vực phân tích dữ liệu nhằm đưa ra xu hướng trong tương lai.

Về mặt thực tiễn, triển khai các ứng dụng trên nền tảng điện toán đám mây – Cloud Computing đang ngày cảng mở rộng và phát triển. Nắm bắt được xu hướng này nên các ứng dụng trong đề tài đều có thể triển khai trên nền tảng điện toán đám mây. Các kỹ thuật Machine Learning, đặc biệt là Regression, rất khả thi và phù hợp cho tập dữ liệu về bóng đá do nó sở hữu rất nhiều thông số dạng numeric, kết quả trận đấu bị tác động nhiều hay ít vì các biến giải thích đó.

## 1.3. Phương pháp tiếp cận

Đầu tiên ta cần thu thập dữ liệu cần thiết dùng cho cả việc thống kê và xây dựng mô hình dự đoán. Một trong những công nghệ dùng để thu thập dữ liệu, nói cách khác là cào – Crawler, phổ biến là Web Crawler, một kỹ thuật dùng để chỉ quá trình truy cập website một cách tự động và lấy dữ liệu thông qua một công cụ hay chương trình phần mềm. Trong đề tài sử dụng thư viện Scrapy trong Python, sẽ được đề cập đến ở phần sau.

Tiếp đến ta cần một ứng dụng thiết lập các kênh để truyền tải thông tin giữa các ứng dụng khác, cần đưa những dữ liệu thu thập ban đầu – dữ liệu thô đến các tầng để có thể xử lý và phân tích chúng. Apache Kafka được sử dụng để đảm bảo điều đó cùng với những ưu điểm, hạn chế cúa nó sẽ được làm rõ ở phần 2 – Giới thiệu công nghệ sử dụng.

Dữ liệu, tất nhiên cần được xử lý và phân tích để đạt được kết quả mà ta mong muốn. Điều ta cần là dữ liệu cần được xử lý nhanh, phân tích được trực quan. Một phương pháp hữu dụng ngày nay là một công cụ xử lý dữ liệu phân tán, Apache Spark là rất phù hợp cho điều đó.

Và ta cũng cần một nơi để lưu trữ dữ liệu về lâu về dài, DBMS – Database Management System (Hệ quản trị cơ sở dữ liệu) hay File Systems (Hệ thống lưu trữ tập tin) đều thông dụng cho các hệ thống ngày nay.

Cuối cùng, ta đưa chúng đến cho người dùng thông qua các công nghệ bao gồm cả Back-end và Front-end.

Một hệ thống streaming data như trên, cụ thể cho đề tài là Hệ thống Streaming Data về bóng đá có thể biểu diễn bằng mô hình 5 layers như sau:

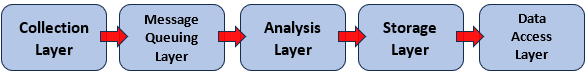
Collection Layer – Tầng thu thập: thư viện Scrapy sẽ tiến hành thu thập dữ liệu về các trận đấu, thông số cầu thủ từ các trang web phân tích nghiệp vụ bóng đá, dữ liệu sau khi được thu thập sẽ tiến hành gửi cho Apache Kafka.

Message Queuing Layer – Tầng hàng đợi thông tin: đây có thể coi là tầng trung gian có nhiệm vụ thiết lập các kênh để truyền tải thông tin giữa các tầng, sau khi nhận được thông tin mà Collection Layer đã gửi, Analysis Layer sẽ tiến hành lấy thông tin cần thiết thông qua Message Queuing Layer.

Analysis Layer – Tầng phân tích: dữ liệu thô về các trận đấu, cầu thủ cần được xử lý nhằm cho mục đích thống kê một cách trực quan và mục tiêu xây dựng mô hình dự đoán dựa trên các thông số.

Storage Layer – Tầng lưu trữ: dữ liệu sau khi được xử lý sẽ được lưu trữ ngắn hạn hoặc dài hạn, đối với dữ liệu về các trận đấu, cầu thủ, chúng cần được ưu tiên lưu trữ dài hạn bởi dữ liệu lịch sử còn rất hữu ích để sử dụng cho phân tích tương lai.

Data Access Layer – Tầng truy cập: tầng hiển thị thông tin đã được lưu trong Storage Layer cho người dùng.



Hình 1.3: Quy trình chung Hệ thống Streaming Data về bóng đá

# PHẦN 2: CƠ SỞ LÝ THUYẾT

## 2.1. Học máy – Machine learning:

Trong phần này, nhóm sẽ liệt kê và mô tả tóm tắt các thuật toán học máy được sử dụng trong đề tài với mục đích “Xây dựng mô hình dự đoán các trận đấu tương lai”, bao gồm các thuật toán nằm trong Supervised Machine Learning – Học máy có giám sát, và tất nhiên, chúng đều được hỗ trợ trong Spark Machine Learning đã được đề cập trong phần 2.1.2.3 – Apache Spark.

Trước hết ta nói qua về Học có giám sát, tiếng anh là Supervised Learning, là thuật toán dự đoán đầu ra của một dữ liệu mới dựa trên các cặp input, outcome đã biết từ trước [1]. Về cơ bản, dữ liệu đầu vào là dữ liệu đã qua đào tạo học máy bằng cách sử dụng dữ liệu đã được gán nhãn hoặc cho ra kết quả rõ ràng. Học có giám sát được phân thành 2 loại thuật toán:

- Phân loại (Classification): các nhãn (kết quả) của dữ liệu đầu vào được chia thành một số hữu hạn nhóm, đây là loại thuật toán mà nhóm sẽ sử dụng với 3 nhóm kết quả: Win, Lose, Draw.

- Hồi quy (Regression): đầu ra là một giá trị thực cụ thể, chẳng hạn như đơn bị tiền tệ hay khối lượng,…

### 2.1.1. Các thuật toán phân loại – Classification:

#### 2.1.1.1. Hồi quy logistic

Hồi quy logistic là một kỹ thuật phân tích dữ liệu sử dụng toán học để tìm ra mối quan hệ giữa hai yếu tố dữ liệu. Mô hình hồi quy logistic được ứng dụng trong dự đoán đối tượng thuộc vào một trong hai nhóm. Hồi quy logistic sử dụng nguyên lí của hàm sigmoid. [2]

Hàm sigmoid là một hàm phi tuyến biến đổi đầu vào thành xác suất tương ứng với một trong hai lớp nhị phân. Hàm sigmoid được diễn giải sau đây:

Với z bất kì, hàm sigmoid trả về đầu ra là một giá trị xác suất nằm trong khoảng [0;1]. Tương tự, ta có nếu đầu vào là một ma trận dữ liệu X và trọng số . [2]

Hàm mất mát của bài toán hồi quy logistic là hàm mất mát Cross-Entropy. Hàm mất mát này được định nghĩa như sau: [2]

Với:

* : Số lượng mẫu từ tập dữ liệu dùng để huấn luyện mô hình dự đoán.
* : Giá trị thực tế từ mô hình của đầu ra thứ i.
* : Với đầu vào thứ i, là xác suất dự đoán đầu vào này thuộc lớp 1.

Ưu điểm:

* Hồi quy logistic là phương pháp dễ dàng sử dụng: Để tạo ra được mô hình máy học thì cần huấn luyện và thử nghiệm mô hình, và mô hình hồi quy logistic dễ dàng trong việc đào tạo và triển khai.
* Hoạt động tốt với các dữ liệu có phân tách tuyến tính: Tập dữ liệu phân tách tuyến tính được hiểu như là có một đường thẳng ở giữa phân tách tập dữ liệu làm hai phần riêng biệt.

Nhược điểm:

* Hồi quy logistic giả định tuyến tính giữa biến dự đoán và biến dự báo: Trong hiện thực, rất ít có trường hợp mà chúng ta có thể thu thập được dữ liệu phân tách tuyến tính. Do vậy, đây vừa là điểm mạnh nhưng vừa là điểm yếu của mô hình hồi quy logistic.
* Tính chính xác thấp nếu như kích thước để huấn luyện mô hình quá nhỏ: Điều này có thể dẫn tới trường hợp mô hình bị quá khớp. Do đó, để dự đoán các kết quả mới sẽ dường như là bất khả thi với trường hợp này.

Trong đề tài này nhóm quyết định sử dụng thuật toán Hồi quy logistic để dự đoán kết quả của trận đấu vì có thể dễ dàng triển khai và vì đầu ra của thuật toán là biến phân loại.

#### 2.1.1.2. Cây quyết định - Decision Tree:

Cây quyết định là phương pháp phân lớp các đối tượng, là một cây phân cấp có cấu trúc và có các luật khác nhau. Các kiểu dữ liệu được sử dụng trong cây quyết định có thể đa dạng như biến số lượng, biến định danh, biến thứ tự, biến nhị phân. [3]

Thuật toán ID3 của cây quyết định là phương pháp tìm kiếm một cách tham lam từ trên xuống dưới và có thể không có quay lui. Thuật toán ID3 sử dụng Entropy và Information Gain để xây dựng nên một cây quyết định. [3]

Đối với phân phối xác suất của biến rời rạc x có thể nhận n giá trị khác nhau, x1, x2, ..., xn­­. Từ đó, ta sẽ có giá trị xác suất khác nhau để x có thể nhận x1, x2, ..., xn ­­, gọi là p1, p2, ..., pn­. Công thức tính p là pi = p(x = xi)­. Từ đó, Entropy của phân phối này được tính theo công thức sau: [3]

Entropy sẽ đạt tối đa khi xác suất xảy ra của hai lớp ngang bằng với nhau. Từ đó phân loại p thành hai loại, tinh khiết và vẩn đục: [3]

* P tinh khiết: pi = 0 hoặc pi = 1
* P vẩn đục: pi = 0.5, khi đó hàm Entropy đạt đỉnh cao nhất

Information gain dựa vào sự giảm của hàm Entropy khi bắt đầu sự phân chia của một thuộc tính. Để xây dựng quyết định thì luôn luôn phải sử dụng các thuộc tính có information gain cao nhất. [3]

Information gain được tính toán tại mỗi nút trong cây quyết định như sau:

* Bước 1: Gọi S là biến mục tiêu có tổng số phần tử là N, NC là số phần tử thuộc lớp C cho trước.
* Bước 2: Tính Entropy, tại mỗi thuộc tính x, dữ liệu trong S sẽ được phân chia thành các k lớp các nốt con gọi là S1, S2, …, Sk với lượng dữ liệu trong mỗi nốt con là m1, m2, …, mk. Ta có:
* Bước 3: Chỉ số Gain Information được tính theo công thức:

Ưu điểm:

* Mô hình cây quyết định là một mô hình có thể diễn giải được, dễ dàng trong việc tìm hiểu và áp dụng.
* Có thể làm việc với nhiều loại biến khác nhau.
* Có khả năng làm việc với dữ liệu lớn.

Nhược điểm:

* Mô hình này phần lớn lệ thuộc vào dữ liệu huấn luyện, một sự thay đổi nhỏ sẽ khiến cả cây quyết định thay đổi.[3]

Mô hình cây quyết định được nhóm sử dụng trong đề tài vì đầu vào của dữ liệu là nhiều loại biến khác nhau, và do mô hình này có khả năng làm việc với dữ liệu lớn.

#### 2.1.1.3. Naive Bayes

Naive Bayes là một kỹ thuật phân loại dựa trên định lý Bayes, mô hình này là một trong những mô hình tiêu biểu sử dụng hướng xác suất thống kê để phân loại, mô hình này giả định mọi biến dự đoán là độc lập với nhau. Phân loại Naïve Bayes là một mô hình học có giám sát và tính xác suất của một giả thuyết dựa trên dữ liệu cho trước [4]. Công thức của định lý Bayes:

Trong đó:

* : Xác suất sau, là xác suất của mục tiêu A với đặc trưng B.
* : Xác suất likelihood, xác suất của đặc trưng B khi biết mục tiêu A.
* : Xác suất trước của mục tiêu A.
* : Xác suất trước của đặc trưng B.

Với công thức này, áp dụng với X là các vector điều kiện đặc trưng,

thì Naïve Bayes sẽ trở thành:

Một số kiểu mô hình Naïve Bayes: [4]

* Naïve Bayes đa thức: Mô hình đa phần được sử dụng trong phân loại văn bản. Đặc trưng đầu vào ở đây chính là tần suất xuất hiện của từ trong văn bản đó.
* Bernoulli Naive Bayes: Bằng cách sử dụng phân phối Bernoulli, áp dụng khi các đặc trưng đầu vào chỉ nhận giá trị nhị phân 0 hoặc 1.
* Gaussian Naive Bayes: Khi đặc trưng có giá trị liên tục, với giả định rằng các biến đều có phân phối Gaussian, ta sẽ tính toán được likelihood theo công thức:

Nhóm quyết định sử dụng mô hình Naïve Bayes trong đề tài này vì để dự đoán thắng thua của các trận bóng đá cần mô hình có đầu ra là biến phân loại, và mô hình Naïve Bayes đáp ứng được nhu cầu.

#### 2.1.1.4. Rừng ngẫu nhiên - Random Forest

Rừng ngẫu nhiên là thuật toán học máy có giám sát, thuật toán này có thể giải quyết bài toán hồi quy lẫn phân loại. [5]

Rừng ngẫu nhiên là một bộ phân loại chứa một số cây quyết định trên các tập con khác nhau của tập dữ liệu đã cho và lấy giá trị trung bình để cải thiện độ chính xác dự đoán của tập dữ liệu đó. Nói cách khác, thuật toán Rừng ngẫu nhiên xây dựng nên nhiều cây quyết định và tập hợp lại, sau đó lấy kết quả theo số đông các cây bằng cách biểu quyết. [5]

Giả định cho thuật toán rừng ngẫu nhiên là các dự đoán từ mỗi cây phải có mối tương quan rất thấp, vì nếu tương quan cao thì không có ý nghĩa gì khi khởi tạo thuật toán rừng ngẫu nhiên. [5]

Ưu điểm: [5]

* Thuật toán mất ít thời gian đào tạo hơn so với các thuật toán khác.
* Thuật toán dự đoán đầu ra với độ chính xác cao, hiệu quả ngay cả đối với tập dữ liệu lớn.
* Khi dữ liệu bị thiếu thì thuật toán vẫn có thể chạy được với độ chính xác ổn định.
* Rừng ngẫu nhiên có thể xử lí cả bài toán phân loại và hồi quy.

Nhược điểm: [5]

* Rừng ngẫu nhiên có độ phức tạp cao.
* Việc xây dựng nên rừng ngẫu nhiên tốn thời gian hơn so với xây dựng cây quyết định.
* Rừng ngẫu nhiên đòi hỏi nặng về mặt tính toán.

Mô hình Rừng ngẫu nhiên được sử dụng trong đề tài này vì đây là mô hình nâng cao hơn của mô hình Cây quyết định, mô hình này linh hoạt trong việc sử dụng nhiều loại biến khác nhau, dễ dàng triển khai, …

## 2.2. Mô hình lập trình tuyến tính nguyên hỗn hợp – Mixed-Integer Linear Programming Model

Mô hình lập trình tuyến tính hỗn hợp số nguyên (MILP) có những vấn đề cần giải quyết sau: [6]

* Hàm mục tiêu tuyến tính, fTx, trong đó c là vectơ cột của các hằng số và x là vectơ cột ẩn số.
* Giới hạn và ràng buộc tuyến tính, nhưng không có ràng buộc phi tuyến tính.
* Một số giới hạn cho thành phần x để có giá trị nguyên.

Trong lập trình toán học, có một số các trường hợp đặc biệt của vấn đề bao gồm các chương trình tuyến tính hoặc LP có thể được viết dưới dạng: [6]

Với:

* f, lb, và ub là các vectors.
* 2 vector tương ứng b and beq.
* Các ma trận A và Aeq.
* Vector cần giải quyết là vector x.

## 2.3. Các thuật toán chuẩn hoá – Regularization Algorithms

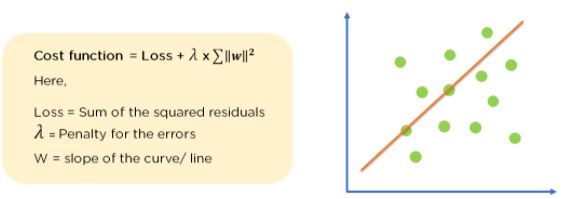
Các thuật toán chuẩn hoá là các kỹ thuật được sử dụng để điều chỉnh các mô hình học máy nhằm giảm thiểu hàm mất mát đã được điều chỉnh và ngăn chặn hiện tượng quá khớp hoặc không khớp. [7]

Có ba thuật toán chính trong Regularization là:

* Hồi quy Ridge.
* Hồi quy LASSO.
* Hồi quy Elastic-Net.

### 2.3.1. Hồi quy Ridge

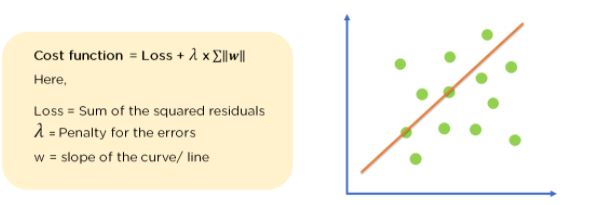
Hồi quy Ridge còn có tên gọi khác là Chuẩn hoá Ridge, thuật toán này điều chỉnh sự quá khớp hoặc không khớp của mô hình bằng cách thêm những hình phạt vào tổng bình phương độ lớn của các hệ số. [7]



*Hình 2.1: Hàm mất mát của Ridge Regression* [7]

Ở trong hàm mất mát, độ lớn của hình phạt được thể hiện bởi Lambda. Mức độ phạt càng cao thì độ lớn của hệ số sẽ giảm. Đối với bài toán Hồi quy Logistic, việc chuẩn hoá sẽ làm hệ số, làm cho giảm độ nhạy của các thuộc tính đối với kết quả dự đoán. [7]

### 2.3.2. Hồi quy Lasso



*Hình 2.2: Hàm mất mát của Hồi quy Lasso* [7]

Hồi quy Lasso được dùng để điều chỉnh việc mô hình bị quá khớp hoặc không khớp tương tự như Hồi quy Ridge, điều khác biệt duy nhất ở đây là thay vì chúng ta bình phương ở tổng hệ số, chúng ta sẽ lấy giá trị tuyệt đối. [7]

Hồi quy Lasso có thể loại bỏ những thuộc tính không cần thiết ra khỏi mô hình bằng cách thay đổi Lambda – tức tăng mức phạt – sẽ khiến cho các hệ số của các thuộc tính không quan trọng tiến về 0 – tức loại bỏ hoàn toàn sự ảnh hưởng của thuộc tính đó khỏi đối với kết quả dự đoán. Trong khi Hồi quy Ridge không thể loại bỏ hoàn toàn các biến, nhưng việc bình phương các trọng số sẽ khiến hệ số của các thuộc tính quan trọng tăng lên đáng kể, và cũng có thể giảm các hệ số của các thuộc tính không quan trọng rất nhiều, nhưng không thể loại bỏ hoàn toàn các biến không quan trọng.Do vậy, Hồi quy Ridge sẽ hoạt động tốt nếu đa phần các thuộc tính là có ý nghĩa đối với kết quả dự đoán, còn Hồi quy Lambda sẽ phù hợp hơn với tập dữ liệu mà các thuộc tính không có ý nghĩa chiếm đa số.

### 2.3.3. Hồi quy ElasticNet

Hồi quy ElasticNet là tổng hợp của hai phương pháp trên, có công thức như sau: [7]

# PHẦN 3 – GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG

## 3.1. Các công nghệ sử dụng

### 3.1.1. Ngôn ngữ lập trình, công cụ hỗ trợ

*Python* là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển. [8]

Python được sử dụng và sở hữu một số ưu điểm:

* Có nhiều thư viện tiêu chuẩn lớn cho các nhà phát triển
* Tương thích tốt với các hệ điều hành Windows, Linux, macOS
* Cộng đồng lớn, nhiều tài liệu, project mã nguồn mở
* Đặc biệt ứng dụng cho phát triển phần mềm, khoa học dữ liệu và

máy học

Do tính tương thích cao cũng như có thể tương tác với nhiều ứng dụng khác dùng trong đề tài nên Python, cụ thể là phiên bản Python 1.11 là ngôn ngữ lập trình phù hợp nhất, một số đặc trưng của Python được áp dụng trong đề tài:

* Scrapy – một Python Framework có thể chạy standalone
* Sau khi Scrapy hoàn thành thu thập dữ liệu, Python có một thư việc khác hỗ trợ kết nối với Apache Kafka, giúp đưa dữ liệu đến cho Apache Kafka
* Python hỗ trợ thư viện PySpark giúp cho các nhà phát triển có thể sử dụng Apache Spark với ngôn ngữ Python, nó như một cổng giao tiếp kết nối 2 ứng dụng lại với nhau
* Như đã đề cập, Python có khả năng tương tác, giao tiếp với các ứng dụng khác với cả các hệ thống lưu trữ, hệ quản trị cơ sở dữ liệu,… Python dễ dàng truy xuất với HDFS, MySQL,…
* Python tất nhiên là công cụ phát triển phần mềm hoạt động rất tốt, phát triển webapp về cả hai mảng Back-end và Front-end. Do đó trong đề tài sử dụng Python Django – một Python Framework phát triển ứng dụng web như là một giao diện cho người dùng.

*Visual Studio Code* là ứng dụng cho phép biên tập, trình soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế phần mềm một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code. Trình soạn thảo này vận hành mượt mà trên mọi nền tảng Windows, macOS, Linux. Hơn thế nữa, VS Code còn cho khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng. [9]

Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép các lập trình viên thay đổi Theme, phím tắt, và đa dạng các tùy chọn khác. VSCode là một trong những Code Editor mạnh mẽ và phổ biến nhất dành cho lập trình viên. Nhờ hỗ trợ nhiều ngôn ngữ lập trình phổ biến, tích hợp đầy đủ các tính năng và khả năng mở rộng, nên VSCode trở nên cực kì thân thuộc với bất kì lập trình viên nào. [9]

Nhóm sử dụng VSCode do sự phát triển mạnh mẽ của nó.

*Jupyter* là một nền tảng tính toán khoa học mã nguồn mở, với khả năng nổi bật cho phép tương tác trực tiếp với từng dòng code (interactive), hỗ trợ hơn 40 ngôn ngữ lập trình, trong đó tập trung vào 3 ngôn ngữ là Julia, Python và R. [10]

Trình làm việc của Jupyter được tổ chức như là một markdown giúp các nhà phát triển có thể dánh dấu, lập chỉ mục giúp cho nội dung các code trở nên rõ ràng, tường minh, trực quan.

### 3.1.2. Các hệ thống, framework, platform

#### 3.1.2.1. Scrapy framework và Selenium

Web Scraping là ứng dụng thu thập dữ liệu được sử dụng để trích xuất dữ liệu từ các trang web. Phần mềm Web Scaping có thể truy cập trực tiếp vào World Wide Web bằng Giao thức HTTP hoặc thông qua trình duyệt web. Mặc dù người dùng phần mềm có thể thực hiện việc quét web theo cách thủ công, nhưng thuật ngữ này thường đề cập đến các quy trình tự động được thực hiện bằng cách sử dụng bot hoặc trình thu thập thông tin web. Đây là một hình thức sao chép, trong đó dữ liệu cụ thể được thu thập và sao chép từ web, thường vào cơ sở dữ liệu cục bộ trung tâm hoặc bảng tính, để truy xuất hoặc phân tích sau này. [11]

*Scrapy* là một Python Framework được phát triển bởi Cambuslang mã nguồn mở [12] hỗ trợ cho việc Crawling hay Scraping dữ liệu từ các trang web thông qua việc download HTML của website và giải nén – extract data từ chúng.

Scrapy được xây dựng xung quanh các “spiders” – các trình thu thập độc lập kèm theo các hướng dẫn. Điều đó giúp cho việc xây dựng và mở rộng quy mô các project thu thập thông tin lớn.

Trong đề tài, Scrapy sẽ tiến hành thu thập dữ liệu về bóng đá như các trận đấu, cầu thủ từ các nguồn chuyên phân tích nghiệp vụ về thể thao bóng đá, nhóm sẽ đề cập chi tiết ở phần sau.

*Selenium* là thư viện phát triển web nguồn mở được sử dụng để tự động hóa các chức năng duyệt web. Nó được phát triển vào năm 2004 và chủ yếu được sử dụng để tự động kiểm tra các trang web và ứng dụng trên nhiều trình duyệt khác nhau, nhưng giờ đây nó đã trở thành một công cụ phổ biến để cào web. Selenium có thể được sử dụng với nhiều ngôn ngữ lập trình, bao gồm Python, Java và C#. Nó cung cấp các API mạnh mẽ để tương tác trên trang web, bao gồm điều hướng, nhấp chuột, gõ và cuộn.[13]

Selenium cơ bản bao gồm 4 thành phần: Selenium Integreted Development Enviroment, Selenium Remote Control, WebDriver, và Selenium Grid. Selenium Integreted Development Enviroment có thể sử dụng để ghi lại hoặc thực hiện lại các thao tác theo một quy trình hay một quy tắc nhất định. Selenium Remote Control có thể khởi chạy máy chủ riêng và tương tác với trình duyệt web. WebDriver giúp gửi lệnh khởi chạy và tương tác với các trình duyệt có sẵn được cài đặt cục bộ ở máy tính thay vì cần phải khởi chạy máy chủ. Và cuối cùng, Selenium Grid giúp khởi chạy các thử nghiệm thông qua nhiều máy và sử dụng đa dạng nhiều trình duyệt tại cùng một thời điểm.[13]

#### 3.1.2.2. Apache Kafka

Apache Kafka là một nền tảng theo kiến trúc phân tán cho phép lưu trữ sự kiện và xử lý dữ liệu luồng (streaming) mã nguồn mở được phát triển bởi Apache Software Foundation được viết bằng Java và Scala. [14]

Ban đầu Kafka được phát triển bởi LinkedIn và từ năm 2011, nó đã trở thành một dự án mã nguồn mở. Hiện nay, Kafka được sử dụng rộng rãi trong các hệ thống xử lý dữ liệu lớn và các hệ thống ứng dụng thời gian thực. Về mặt kiến trúc, Kafka lưu trữ các gói tin dưới dạng cặp key-value và các gói tin này được gửi từ nhiều tiến trình gọi là producer. Kafka sử dụng các partition để phân chia dữ liệu vào các chủ đề (topic). Trong mỗi partition, các gói tin được sắp xếp một cách nghiêm ngặt và được lưu trữ với chỉ mục và đánh dấu thời gian. Các consumer là các tiến trình có thể đọc các gói tin từ các partition. Kafka cung cấp Stream API để xử lý dữ liệu luồng, cho phép các ứng dụng Java (như Apache Spark) sử dụng dữ liệu từ Kafka và ghi kết quả trở lại Kafka. Các hệ thống xử lý dữ liệu luồng khác như Apache Apex, Apache Flink, Apache Storm, Apache Spark,… cũng tương thích với Kafka. [15]

Kafka chạy trên một cụm (cluster) có thể bao gồm một hoặc nhiều máy chủ được gọi là broker. Các partition của tất cả các topic được phân bổ trên các node trong cụm máy chủ đó và các partition được sao chép cho nhiều broker. Điều này cho phép Kafka truyền tải dữ liệu lớn mà vẫn có khả năng chịu tải cao và kháng lỗi. [15]

Có 5 API chính trong Kafka:

* Producer API: xuất bản các luồng dữ liệu ghi
* Consumer API: cấp quyền cho các ứng dụng ngoài đăng ký các topic và ghi dữ liệu
* Connector API: liên kết các topic với các ứng dụng ngoài, thực hiện cả producer và consumer
* Streams API: quá trình chuyển đổi dữ liệu từ input thành output
* Admin API: quản lý các topic, broker và các đối tượng khác

Giao thức truyền tin cơ bản của Kafka được viết dưới dạng nhị phân binary nên các nhà phát triển có thể viết các ứng dụng consumer và producer tùy thích trên mọi ngôn ngữ lập trình.

Ngày nay thì Kafka đã phát triển trở thành nền tảng stream dữ liệu phân tán được sử dụng rộng rãi nhất bởi khả năng xử lý hàng nghìn tỷ bản ghi mà không có bất kỳ đột trễ về hiệu suất, các tổ chức Microsoft, AirBnB và Netflix đều sử dụng Kafka nhằm mục đích trải nhiệm theo thời gian thực, hướng dữ liệu theo khách hàng. [15]

Một số ưu điểm nổi trội của Kafka:

* Thông lượng cao – High-throughput: xử lý lượng lớn dữ liệu một các liên tục gần như là không có thời gian chờ
* Khả năng mở rộng – Scalability: dễ dàng mở rộng các partition, topic mà không cần thời gian chờ
* Cộng đồng người dùng đông đảo, có các diễn đàn hỗ trợ, hỏi đáp

Nhược điểm lớn nhất của Kafka có lẽ là chưa có một bộ công cụ giám sát hoàn chỉnh, tuy vẫn có các tool quản lý, hỗ trợ message như GUI tool, Kafka tool nhưng hầu như mỗi tool chỉ quản lý một vấn đề nghiệp vụ.

Trong đề tài, dữ liệu về các trận đấu hay cầu thủ có thể được biểu diễn dưới dạng luồng để được xử lý và phân tích thông qua hệ thống processing data Apache Spark, với những lợi thế mà Apache Kafka đem lại, sẽ là hợp lý cho việc sử dụng Kafka cho Collection Layer – Tầng thu thập.

#### 3.1.2.3. Apache Spark

Apache Spark là một hệ thống xử lý và phân tích dữ liệu quy mô lớn mã nguồn mở - open-source. Spark cung cấp giao diện cho các ứng dụng khác có thể giao tiếp, đặc trưng với tính kháng lỗi cao. Ban đầu được phát triển bởi AMPLab thuộc trường Đại học California, Berkeley, sau đó được trao tặng cho Apache Software Foundation – quỹ phần mềm Apache và được Apache duy trì hoạt động cho đến ngày nay. [16]

Apache Spark được xây dựng dựa trên kiến trúc Resilient Distributed Dataset (RDD) – bộ dữ liệu phân tán linh hoạt, nó là tập hợp dữ liệu chỉ đọc – read-only được phân bổ trên các cụm máy và được duy trì theo hướng có khả năng chịu lỗi. Một số dạng trừu tượng trong RDD có thể kể đến như Dataframe API, Dataset API. Các phiên bản Spark trước version 2 sử dụng RDD API, về sau được thay thế bởi Dataset API nhưng nhìn chung RDD vẫn là nền tảng cho Dataset API. [16]

Spark được phát triển để kế thừa và phát huy nhưng ưu điểm, hạn chế trong mô hình MapReduce triển khai theo cụm. Hoạt động của nó có thể tóm tắt qua các quy trình đọc dữ liệu, ánh xạ dữ liệu, giảm kết quả và lưu trữ dữ liệu. [16]

Spark có thể truy vấn cơ sở dữ liệu nhưng có thể lặp đi lặp lại điều đó và độ trễ cũng giảm đi một cách đáng kể. Spark yêu cầu phải có một Cluster Manager – trình quản lý cụm và hệ thống lưu trữ phân tán – Distributed Storage. Đối với việc quản lý cụm, các nhà phát triển có thể tiến hành khởi chạy một cụm Spark theo cách thủ công, sử dụng các tập lệnh khởi chạy có sẵn từ nhà cung cấp hoặc có thể khởi chạy Spark trên các cụm phân tán như Hadoop Yarn, Kubernetes,… Các hệ thống lưu trữ phân tán mà Spark có thể giao tiếp như Hadoop Distributed File System (HDFS), Cassandra,… và với cả Amazon S3 – distributed storage trên dịch vụ điện toán đám mây của Amazon. Với mục đích chỉ phát triển hoặc thử nghiệm, Spark hỗ trợ chế độ cục bộ giả phân phối – pseudo-distributed local mode giúp người dùng có thể tiến hành khởi chạy Spark mà không cần Storage Distributed. [16]

Apache Spark có một số framework, abtraction sử dụng cho nhiều mục đích khác nhau, trong đề tài nhóm sử dụng Spark Streaming, Spark SQL và Spark Mllib, chúng là những thành phần cốt lõi cho Tầng phân tích – Analysis Layer. [16]

Trước hết ta nói qua về Spark Core, nền tảng chung cho tổng thể project. Nó điều phối, lập lịch các thao tác I/O cơ bản, được thể hiện thông qua các API cho Java, Scala, Python,… dựa trên tính trừu tượng của RDD. Các hoạt động trong Spark Core như map, filter, reduce, join gọi là các hàm trên Spark, các hàm này sau đó sẽ được lên lịch dể thực hiện song song trên cụm. Giải thích thêm, RDD có thể chứa bất kỳ loại đối tượng nào như Python, .NET, Scala,… [16]

Spark Streaming áp dụng khả năng lập lịch tiến trình nhanh của Spare Core để thực hiện phân tích streaming. Spark Streaming nhập dữ liệu theo từng lô nhỏ - batch và thực hiện các phép biến đổi RDD trên các lô nhỏ đó. Điều đó giúp cho Spark Streaming có thể phân tích dữ liệu hàng loạt nhưng đi cùng với hạn chế lớn nhất chính là độ trễ. Spark Streaming như là một consumer, có khả năng nhập dữ liệu từ các nguồn TCP/IP, Twitter, Kinesis,.. và Apache Kafka – đây là một trong những lý do chính mà nhóm sử dụng Apache Spark như một kênh consumer. [16]

Spark SQL là một thành phần dựa trên Spark Core, một trừu tượng hóa dữ liệu gọi là Dataframe hỗ trợ dữ liệu có cấu trúc và bán cấu trúc. Spark SQL cung cấp ngôn ngữ dành riêng cho thao tác với Dataframe trong Python, Scala,… và cả ngôn ngữ truy vấn SQL. Tập dữ liệu trong đề tài có định dạng là dữ liệu có cấu trúc nên Spark SQL sẽ hỗ trợ rất tốt cho việc truy vấn. [16]

Spark MLlib – Machine Learning Library, một distributed machine-learning framework trên Spark Core, bao gồm nhiều thư viện gồm các thuật toán thống kê và học máy phổ biến giúp triển khai các quy trình học máy trên quy mô lớn dựa vào Spark Core. Các thư viện có rất nhiều chức năng có thể kể đến như thống kê, tóm tắt, kiểm định giả thuyết, các bài toán phân loại và hồi quy (logicstic regression, linear regression, decision tree,…), phân tích cụm, tối ưu hóa gradient,… Để xây dựng một mô hình học máy thì Spark MLlib sẽ hỗ trợ rất tốt cho đề tài. [16]

Tóm lại, Apache Spark là sự lựa chọn hoàn chỉnh cho nghiệp vụ đề tài, rất thích hợp cho Tầng phân tích - Analysis Layer như đã đề cập ở phần trước. Về ý nghĩa khoa học, các module, framework của Spark sẽ cung cấp những tiện ích, công cụ tốt nhất cho việc xử lý và phân tích tập dữ liệu. Về mặt thực tiễn, các nhà phát triển có thể triển khai Spark trên cả các dịch vụ điện toán đám mây.

#### 3.1.2.4. MySQL

MySQL được phát triển, phát hành và hỗ trợ bởi tổ chức Oracle. MySQL được phát hành phiên bản chính thức đầu tiên vào năm 1994 và sau này trở thành một trong những hệ quản trị cơ sở dữ liệu mã nguồn mở nổi tiếng trên thế giới. Ngoài ra, MySQL sử dụng mô hình máy khách – máy chủ. MySQL hoàn toàn có thể được sử dụng mà không cần phải mất phí.

Cơ chế hoạt động của MySQL:

* MySQL đang tạo ra bảng để có thể lưu trữ dữ liệu và định nghĩa về sự liên quan giữa những bảng đó
* Client sẽ trực tiếp gửi yêu cầu SQL bằng 1 lệnh đặc biệt có trên MySQL.
* Ứng dụng tại server sẽ tiến hành phản hồi thông tin cũng như trả về những kết quả trên máy client.

HDFS là lựa chọn cần thiết cho Tầng lưu trữ - Storage Layer trong đề tài bởi khả năng tương thích với các ứng dụng khác.

#### 3.1.2.5. Python Django

Django là một framework miễn phí và mã nguồn mở dành cho việc lập trình web. Được viết hoàn toàn bằng ngôn ngữ Python, Django có đầy đủ các module phục vụ cho các nhà phát triển web. Mục tiêu chính của Django là đơn giản hoá trong công cuộc tạo nên các trang web phức tạp mà có sử dụng cơ sở dữ liệu. [20]

Django sử dụng mô hình Model-View-Template.Mô hình này gồm 3 phần gồm Model, View và Template. Model là phần xử lý logic và tương tác với cơ sở dữ liệu. View là thành phần đảm nhiệm hiển thị cho người dùng, view lấy dữ liệu từ thành phần model. Thành phần cuối cùng là template, template là khung để tạo ra các giao diện của trang web. Template bao gồm có thành phần cố định và các thành phần động. [20]

Django được đánh giá bởi các lập trình viên là nhanh trong quá trình xây dựng web, vì chính bản thân Django đã được xây dựng sẵn những gói tiện ích đến từ cộng đồng người dùng với hơn 4000 gói hỗ trợ cho việc kiểm thử, gỡ lỗi, … [20]

Django được đánh giá xây dựng nên trang web nhanh chóng do luôn tuân thủ hai quy tắc là “DRY – Don’t repeat yourself”, có nghĩa là giữ cho việc phát triển mã nguồn tránh bị lặp lại. Quy tắc thứ hai là “KISS - Keep it short and simple”, có nghĩa là càng ngắn gọn càng tốt. [20]

Django có hỗ trợ một vài lỗ hổng bảo mật cơ bản như bảo vệ khỏi tấn công kịch bản chéo trang, giả mạo yêu cầu chéo trang web, tiêm SQL, clickjacking, … [20]

Django hoàn toàn có thể được mở rộng trong tương lai cả theo chiều dọc và theo chiều ngang. Đối với mở rộng theo chiều ngang, Django hỗ trợ bộ nhớ đệm (Bộ nhớ đệm bộ nhớ hoặc Redis). [20]

Ưu điểm:

* Django hỗ trợ xây dựng lên trang web nhanh chóng.
* Django có khả năng mở rộng theo chiều ngang và theo chiều dọc.
* Connection pool được hỗ trợ khi kết nối tới cơ sở dữ liệu trong Django.

Nhược điểm:

* Để làm chủ được Django sẽ khó khăn hơn các framework khác vì có nhiều cấu hình và cài đặt.
* Các dự án sử dụng Django phải là nguyên khối, nếu các nhà phát triển web muốn truy cập và sử dụng một tệp bất kì nằm ngoài thư mục dự án của Django sẽ khó khăn hơn.
* Django không hỗ trợ xử lí nhiều request đồng thời nên các nhà phát triển phải tìm hướng đi khác nếu muốn tiếp cận hướng đi này.

#### 3.1.2.6. AngularJS

Định nghĩa chính thức về AngularJS được trình bày như sau: “*AngularJS là một framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Nó xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.*” [21]

AngularJS giúp các lập trình viên có thể xây dựng nên một ứng dụng nhiều công dụng và có thể lựa chọn viết ứng dụng đó thuần về phía khách sử dụng JavaScript và dựa trên MVC. Framework này có thể được chạy đa dạng trên nhiều trình duyệt. Và hơn nữa, đây là một framework mã nguồn mở và hoàn toàn miễn phí, có cộng đồng sử dụng rộng lớn và được hỗ trợ từ cộng đồng các lập trình viên sử dụng AngularJS. [22]

Các nhóm tính năng chính của AngularJS bao gồm: [22]

* Ràng buộc dữ liệu − Tự động đồng bộ data giữa model và view components, trong cấu trúc Model-View-Presenter (Model-View-Presenter bao gồm Model là lớp mô tả dữ liệu và chứa dữ liệu, View là thành phần tương tác với người dùng, Presenter sẽ nhận đầu vào của người dùng thông qua View, Presenter kiểm soát Model và cập nhật vào View)
* Scope− Danh sách các đối tượng tham chiếu tới model. Kết nối giữa controller và view
* Controller – Là các hàm Javascript được ràng buộc phạm vi cụ thể
* Services − Xử lý logic của các ứng dụng.
* Filters – Được sử dụng để lọc mảng, từ nguồn ban đầu tạo ra một mảng mới
* Directives − Là các đối tượng trên cây DOM (các phần tử, thuộc tính, CSS), dùng để tạo các thẻ HTML tuỳ chỉnh. Angular thì dùng ngBind, ngModel, …
* Templates – Template có nhiệm vụ render ra view, thường là một tệp duy nhất kiểu như index.html hoặc nhiều views kết hợp với nhau.
* Routing – Routing giúp chuyển đổi các view với nhau
* Model View Whatever − MVW chia ứng dụng thành các phần như Model, View và Controller, mỗi một thành phần chịu trách nhiệm riêng.
* Deep Linking − Deep linking cho phép mã hoá trạng thái hiện tại của ứng dụng. Sau này từ URL, ứng dụng có thể khôi phục lại trạng thái.

## 3.2. Môi trường thực nghiệm

### 3.2.1. Cấu hình chung

Hệ điều hành: Windows

Memory: 16GB

CPU: Core(TM) i5-7200U CPU @ 2.50Hz

### 3.2.2. Hoạt động các ứng dụng

|  |  |  |  |
| --- | --- | --- | --- |
|  | RAM | Disk space | Process time |
| Scrapy | ~ 2 GB | 117MB (bao gồm tệp code thực thi, file csv) | 48h37p (để generate ra 5 file CSV) |
| Apache Kafka | ~ 1 GB | ~ 67KB (bao gồm code thực thi) | <1s |
| Pandas Python Runtime | ~ 2.5 GB | ~ 95MB (bao gồm tệp code thực thi, file csv) | ~40p (kết hợp FBref và Sofifa)  ~17p (xây dựng tập data cho mô hình dự đoán) |
| Apache Spark | ~ 2.5 GB | ~ 2.5MB (bao gồm tệp code thực thi, file csv) | ~9p (cho việc huấn luận, đánh giá cho mỗi thuật toán) |
| Python Django | ~ 1 GB | ~ 1.5MB (bao gồm tệp code thực thi, file csv) | ~10s (mỗi lần khởi động server) |
| AngularJS | ~2 GB | ~ 190MB (bao gồm tệp code thực thi, hình ảnh) | ~5p (cho mỗi lần khởi động server) |

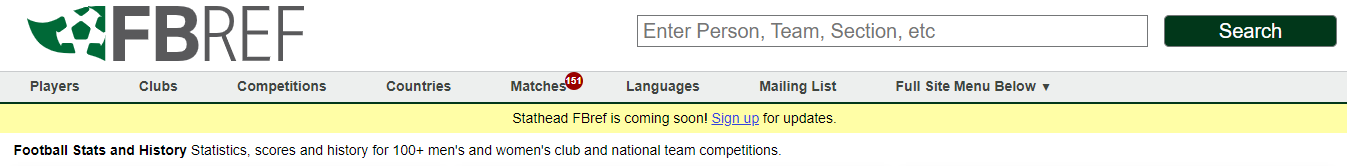
# PHẦN 4 – MÔ TẢ TẬP DỮ LIỆU

## 4.1. Nguồn thu thập

Tập dữ liệu về bóng đá trong đề tài có thể được chia thành 2 thành phần: trận đấu và cầu thủ. Với công nghệ thu thập dữ liệu Scrapy như đã đề cập ở Phần 2 thì nguồn dữ liệu sẽ được trích xuất từ các website phân tích nghiệp vụ về môn thể thao bóng đá.

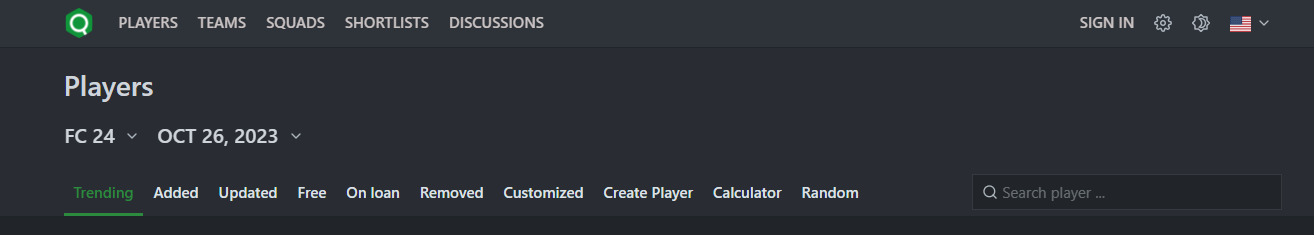
Đầu tiên là website *FBref.com*, một trang web chuyên về theo dõi các số liệu thống kê về các đội bóng và cầu thủ trên toàn thế giới qua các trận đấu, bao gồm các môn bóng đá nam và nữ. Được tạo bởi nhóm Sports Reference, nhóm đứng đằng sau các trang web thống kê phổ biến khác như Baseball-Reference (thống kê số liệu về giải bóng chày Major League của Mỹ) và Basketball-Reference (các số liệu về giải bóng rổ NBA của Mỹ). Ngoài ra *FBref.com* còn có các thông tin về bộ môn khúc khôn cầu và các môn thể thao ở các trường Đại học. *FBref.com* ra mắt vào tháng 6 năm 2018 với phạm vi phủ sóng 6 giải đấu hàng đầu thế giới: Anh, Pháp, Ý, Đức, Tây Ban Nha, Mỹ và hiện nay đã mở rộng tới hơn 45 quốc gia và hơn 140 giải đấu khác. FBRef bao gồm lịch sử thi đấu, các công cụ để phân tích chuyên sâu về nghiệp vụ trong bộ môn bóng đá. [23]

Các thông tin về các trận đấu bóng đá nhóm sẽ tiến hành thu thập ở website này.



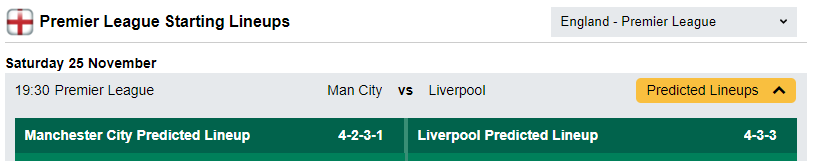
*Hình 4.1: Một góc các tính năng có trong FBRef bao gồm players, clubs, competitions, countries, matches,…*

Tiếp theo là *sofifa.com*, một website chuyên tính toán, phân tích để đưa ra các số liệu cá nhân của các cầu thủ, tập thể của các đội bóng dựa vào phong độ hiện tại trên thực thế và áp dụng các thông số đó vào trong tựa game FIFA – được xuất bản bởi EA Sport. Nói qua một chút về video game FIFA thì nó là một trong những thương hiệu trò chơi điện tử bán chạy nhất trên thế giới với nội dung game liên quan đến bộ môn thể thao bóng đá. Có thể xem *sofifa.com* là một công cụ đánh giá chỉ số cá nhân các cầu thủ và tập thể các đội bóng hoàn chỉnh và thực tế nhất. Ngoài ra *sofifa.com* luôn cập nhật các thông tin một các đầy đủ chỉ trong vài ngày tùy thuộc vào phong độ thực tế. Điều đó là cơ sở chính mà nhóm sẽ dùng sofifa để thu thập thông tin các cầu thủ một cách khách quan nhất.



*Hình 4.2: Một góc các tính năng trong Sofifa bao gồm player, teams, squads qua các mốc thời gian và phiên bản*

Nguồn thu thập cuối cùng là Sportsgambler.com, một nền tảng cung cấp cho người dùng các số liệu thống kê, tin tức đội bóng, xem trước các thông tin trận đấu. Sportsgambler.com được cung cấp bởi các chuyên gia hàng đầu trong lĩnh vực thể thao, không chỉ tập trung vào các giải bóng đá nổi tiếng nhu Premier League hay Champions League mà còn là các giải khác như giải quần vợt ATP, WTA, giải bóng rổ NBA, bóng chày MLB,… Vì các thông tin trước trận đấu về đội hình và cầu thủ ra sân mà nền tảng này đem lại rất nhanh chóng và kịp thời nên Sportsgambler.com là một nguồn thu thập tốt cho các trận đấu trong tương lai.



Hình 4.3: Sportsgambler.com đưa ra đội hình và các cầu thủ xuất phát của một trận đấu chưa diễn ra

## 4.2. Mô tả chi tiết tập dữ liệu

Với trọng tâm của đề tài “Xây dựng nền tảng thống kế và dự đoán các trận đấu bóng đá”, nhóm tiến hành chia tập dữ liệu về các trận đấu thành 2 nhóm:

* *Tập dữ liệu lịch sử*: bao gồm các trận đấu đã xảy ra tới thời điểm hiện tại.
* *Tập dữ liệu tương lai*: bao gồm các trận đấu chưa diễn ra.

Một trận đấu được đưa vào *Tập dữ liệu lịch sử* khi nguồn thu thập *FBref.com* cập nhật đầy đủ các thông tin của trận đấu đó. Với các trận đấu chưa được kết thúc hay kết thúc mà *FBref.com* chưa cập nhật đầy đủ thông tin (thường sẽ được cập nhật đầy đủ sau vài giờ kết thúc trận đấu) và đã được đưa ra đội hình ra sân dự kiến từ *Sportgambler.com* thì chúng được đưa vào nhóm *Tập dữ liệu tương lai*.

Xuyên suốt phần 4 – Xây dựng hệ thống Streaming Data, với mỗi tầng đều có sự tương tác, xử lý hay phân tích 2 nhóm tập dữ liệu ở trên.

### 4.2.1. Trận đấu

Các trận đấu trong tập dữ liệu sẽ có phạm vi bao gồm tất cả các trận đấu bóng đá nam thuộc 5 giải bóng đá châu Âu: Premier League – Anh, La Liga – Tây Ban Nha, Serie A – Ý, Bundesliga – Đức và Ligue 1 – Pháp. Với 7 mùa giải gần nhất (từ 2017-2018 đến mùa giải hiện tại 2023-2024).

Một giải đấu Premier League, La Liga và Seria A trong một mùa giải sẽ có 20 đội bóng trong khu vực tham gia, đối với Bundesliga và Ligue 1 trong một mùa giải sẽ có ít hơn với 18 đội tham gia. Với một trong ba giải đấu Premier League, La Liga và Seria A thì sẽ có tổng cộng 380 trận/ 1 mùa, với một trong hai giải đấu Bundesliga và Ligue 1 sẽ có tổng cộng 306 trận/ 1 mùa. Mùa giải hiện tại 2023-2024 vẫn còn đang diễn ra, nên nếu tính 6 mùa giải trước thì ta có tổng cộng (380\*3 + 306\*2) \* = 10512 trận đấu (cho cả 5 giải đấu trong 6 mùa giải) trong tập dữ liệu cho đề tài.

Thông tin về các trận đấu nhóm tiến hành chia thành 5 thành phần mô tả đầy đủ các thuộc tính của một trận đấu nằm trong nhóm *Tập dữ liệu lịch sử* : MatchInfos, MatchGoals, MatchPlayerStats, MatchSquad và MatchStats. Mỗi thành phần sẽ tương ứng với một table sau quá trình tiền xử lý dữ liệu – sẽ được mô tả ở phần sau. Mọi thuộc tính – attribute trong các table đều được sử dụng cho mục đích thống kê và một số thuộc tính được kết hợp từ nhiều table được dùng cho xây dựng mô hình dự đoán.

Đầu tiên là MatchInfos table, mô tả các thông tin ban đầu, thông tin chung, ngoài lề của một trận đấu:

|  |  |  |
| --- | --- | --- |
| *Attribute* | *Mô tả* | *Ví dụ* |
| Match\_Id | Key phân biệt | db4d7668 |
| League | Tên giải đấu | Premier League |
| Season | Mùa giải | 2023/2024 |
| Match\_Week | Tuần thi đấu | Matchweek 10 |
| Home\_Team | Đội sân nhà | Manchester United |
| Away\_Team | Đội sân khách | Manchester City |
| Match\_Date | Ngày thi đấu (tại sân) | 10/29/2023 |
| Venue\_Time | Thời gian bắt đầu (tại sân) | 15:30 |
| Attendance | Số lượng khán giả | 20067 |
| Stadium | Sân tổ chức | Old Trafford, Manchester |
| Officials | Tổ trọng tài | Stuart Attwell (Referee), Adam Nunn (AR1),  Simon Long (AR2),  Gavin Ward (4th),  Simon Hooper (VAR) |

*Bảng 4.1: Mô tả table MatchInfos trong tập dữ liệu*

Tiếp theo là MatchSquad liệt kê các cầu thủ trong đội hình cả đá chính lẫn dự bị của cả hai đội trong trận đấu:

|  |  |  |
| --- | --- | --- |
| *Attribute* | *Mô tả* | *Ví dụ* |
| Match\_Id | Key phân biệt | db4d7668 |
| Team | Tên đội | Manchester United |
| Is\_Home\_Team | Sân nhà hoặc sân khách | Yes |
| Player\_Name | Tên cầu thủ | Antony |
| Player\_Kitnum | Số áo cầu thủ | 7 |
| Is\_Sub | Dự bị | No |

*Bảng 4.2: Mô tả table MatchSquad trong tập dữ liệu*

Kế tiếp là MatchGoals mô tả các thời điểm ghi bàn trong một trận đấu ứng với số phút, cầu thủ ghi bàn và kiểu ghi bàn (normal hoặc penalty):

|  |  |  |
| --- | --- | --- |
| *Attribute* | *Mô tả* | *Ví dụ* |
| Match\_Id | Key phân biệt | db4d7668 |
| Team | Tên đội | Manchester United |
| Minute | Số phút ghi bàn | 89 |
| Player\_Name | Tên cầu thủ | Antony |
| Type\_Of\_Goal | Số áo cầu thủ | Normal |
| Is\_Home\_Team | Sân nhà hoặc sân khách | Yes |

*Bảng 4.3: Mô tả table MatchGoals trong tập dữ liệu*

MatchStats bao gồm các thông số của cả mỗi đội trong mỗi trận đấu, các thông số đó sẽ được dùng cho việc xây dựng mô hình dự đoán, MatchPlayersStat cũng sử dụng các thông số đó với mô tả chi tiết cho từng cầu thủ:

|  |  |  |
| --- | --- | --- |
| *Attribute* | *Mô tả* | *Ví dụ* |
| Match\_Id | Key phân biệt | db4d7668 |
| Team | Tên đội | Manchester United |
| Is\_Home\_Team | Sân nhà hoặc sân khách | Yes |
| Manager | Huấn luyện viên | Carlo Ancelotti |
| Captain | Đội trưởng | Harry Maguire |
| Formation | Kiểu đội hình | 4222 |
| Posession | Tỉ lệ kiểm soát | 55 |
| Fouls | Số lần phạm lỗi | 10 |
| Corners | Số phạt góc | 7 |
| Crosses | Số đường chọc khe | 21 |
| Aerials\_Won | Số lần thắng không chiến | 35 |
| Clearances | Số cơ hội tạo ra | 30 |
| Offsides | Số lần việt vị | 4 |
| Goal\_Kicks | Số cú sút | 10 |
| Throw\_Ins | Số lần ném biên | 20 |
| Long\_Balls | Số đường chuyền dài | 70 |
| Minutes | Tổng số phút đã chơi  cả đội | 990 |
| Gls | Số bàn thắng | 1 |
| Ast | Số kiến tạo | 1 |
| PK | Số penalty hoàn thành | 0 |
| PK\_Att | Số lần sút penalty | 1 |
| Sh | Số cú sút | 20 |
| SoT | Số cú sút trúng đích | 15 |
| CrdY | Số thẻ vàng | 4 |
| CrdR | Số thẻ đỏ | 0 |
| Touches | Số lần chạm bóng | 580 |
| Tkl | Số lần tắc bóng | 20 |
| Int | Số lần can thiệp | 10 |
| Blocks | Số lần đánh chặn | 15 |
| xG | Số bàn thắng kỳ vọng | 2.1 |
| npxG | Số bàn thắng kỳ vọng (không kể penalty) | 2.1 |
| xAG | Số kiến tạo kỳ vọng | 1.7 |
| SCA | Số cơ hội tạo ra cú sút | 30 |
| GCA | Số cơ hội tạo ra bàn thắng | 4 |
| Cmp\_Passes | Số đường chuyền  thành công | 565 |
| Att\_Passes | Số đường chuyền | 660 |
| Cmp\_percent\_Passes | Tỉ lệ chuyền thành công | 85.45 |
| PrgP\_Passes | Số đường chuyền tới | 71 |
| Carries\_Carries | Thời gian cầm bóng | 483 |
| PrgC\_Carries | Thời gian cầm bóng tiến tới | 24 |
| Att\_Take\_Ons | Số lần vượt qua  hậu vệ đối phương | 26 |
| Succ\_Take\_Ons | Số lần vượt qua hậu vệ đối phương thành công | 9 |

*Bảng 4.4: Mô tả table MatchStats trong tập dữ liệu*

### 4.2.2. Cầu thủ

Các cầu thủ trong Sofifa có phạm vi bao gồm tất cả các phiên bản từ FIFA 13 – năm 2013 đến FC 24 – năm 2024 (phiên bản ở thời điểm hiện tại) trong giới hạn các đội bóng của 5 giải bóng đá châu Âu: Premier League – Anh, La Liga – Tây Ban Nha, Serie A – Ý, Bundesliga – Đức và Ligue 1 – Pháp. Tập dữ liệu các cầu thủ được sắp xếp theo thứ tự các phiên bản với phiên bản đầu tiên của mỗi tháng. Do đó với các trận đấu trong một tháng thì các chỉ số về cầu thủ sẽ được lấy từ phiên bản đầu tiên của mỗi tháng đó. Với mỗi cầu thủ sẽ có Sofifa\_Id – khóa chính để phân biệt cho dù có nhiều chỉ số sẽ bị thay đổi qua mỗi phiên bản. Ngoài Sofifa\_Id thì tên riêng định danh cho mỗi cầu thủ cũng không bị thay đổi qua các phiên bản.

Một cầu thủ bao gồm rất nhiều chỉ số cá nhân để đánh giá, nhưng với thuộc tính Overall\_Rating – chỉ số được tổng hợp từ tất cả chỉ số khác, có thể coi nó là điểm số cuối cùng để đánh giá một cầu thủ, Do đó, tổng hợp các Overall\_Rating của các cầu thủ trong một đội bóng cũng được dùng trong việc xây dựng mô hình dự đoán. Tất cả các thông tin của một cầu thủ nhóm tiến hành chia thành 2 thành phần: PlayerÌnfos và PlayerAttrs.

PlayerInfos mô tả các thông tin chung của một cầu thủ:

|  |  |  |
| --- | --- | --- |
| *Attribute* | *Mô tả* | *Ví dụ* |
| Player\_Id | Key cầu thủ | 192985 |
| Player\_Name | Tên cầu thủ | K. De Bruyne |
| Player\_Full\_Name | Tên đầy đủ cầu thủ | Kevin De Bruyne |
| Birthdate | Ngày sinh | 1991/6/28 |
| Age | Tuổi | 32 |
| Value | Giá trị chuyển nhượng | 103M |
| Wage | Lương theo tuần | 350K |
| Update\_Date | Ngày cập nhật, phiên bản hiện tại | 2023/9/22 |
| Team | CLB và đội tuyển quốc gia | Manchester United |
| Team\_Contract | Năm hết hạn hợp đồng | 2027 |
| Team\_Joined | Ngày ký hợp đồng | 2022/8/3 |
| Team\_Kitnum | Số áo | 10 |
| Team\_Loaned\_From | CLB cho mượn | Manchester City |
| Preferred\_Foot | Chân thuận | Right |
| Skill\_Moves | Điểm kỹ năng  di chuyền | 4 |
| Weak\_Foot | Điểm dùng chân  không thuận | 5 |
| Reputation | Điểm danh tiếng | 5 |
| Work\_Rate | Khả năng hoạt động | High |
| Specialities | Kỹ năng đặc biệt | #Dribbler, #Playmaker |

*Bảng 4.5: Mô tả table SofifaInfos trong tập dữ liệu*

PlayerAttrs bao gồm tất cả các thuộc tính đánh giá chỉ số cá nhân của cầu thủ, với 2 thuộc tính Overall\_Rating và All\_Possitions được dùng cho việc xây dựng mô hình dự đoán:

|  |  |  |
| --- | --- | --- |
| *Attribute* | *Mô tả* | *Ví dụ* |
| Crossing | Chọc khe | 95 |
| Finishing | Dứt điểm | 85 |
| Heading\_Accuracy | Đánh đầu chính xác | 55 |
| Short\_Passing | Chuyền ngắn | 94 |
| Volleys | Dứt điểm một chạm | 83 |
| Dribbling | Rê dắt bóng | 86 |
| Curve | Cứa lòng | 92 |
| FK Accuracy | Sút phạt chính xác | 83 |
| Long\_Passing | Chuyền dài | 94 |
| Ball\_Control | Kiểm soát bóng | 92 |
| Acceleration | Tăng tốc | 72 |
| Sprint\_Speed | Chạy nước rút | 72 |
| Agility | Nhanh nhẹn | 74 |
| Reactions | Phản xạ | 92 |
| Balance | Thăng bằng | 78 |
| Shot\_Power | Lực sút | 92 |
| Jumping | Bật nhảy | 72 |
| Stamina | Thể lực | 88 |
| Strength | Sức mạnh | 74 |
| Long\_Shots | Sút xa | 92 |
| Aggression | Xông xáo | 75 |
| Interceptions | Can thiệp | 66 |
| Vision | Tầm nhìn | 95 |
| Penalties | Khả năng sút penalty | 83 |
| Composure | Bình tĩnh | 88 |
| Defensive\_Awareness | Nhận thức phòng thủ | 66 |
| Standing\_Tackle | Tắc bóng đứng | 70 |
| Sliding\_Tackle | Xoạc bóng | 53 |
| Marking | Theo kèm | 50 |
| GK\_Diving | Nhào bắt bóng | 15 |
| GK\_Handling | Bắt bóng | 13 |
| GK\_Kicking | Phát bóng | 5 |
| GK\_Positioning | Giữ vị trí | 10 |
| GK\_Reflexes | Phản xạ thủ môn | 13 |
| Height | Chiều cao | 185 |
| Weight | Cân nặng | 80 |
| All\_Positions | Tất cả các vị trí  có thể chơi | CM, CDM, CAM |
| Overall\_Rating | Chỉ số tổng thể | 91 |

*Bảng 4.1: Mô tả table SofifaAttr trong tập dữ liệu*

# PHẦN 5: XÂY DỰNG HỆ THỐNG

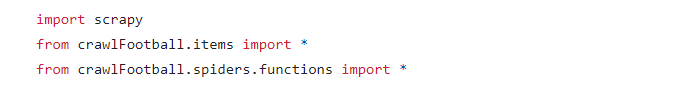
# STREAMING DATA

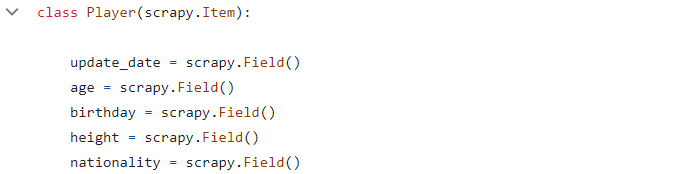
## 5.1. Tầng thu thập – Collection Layer:

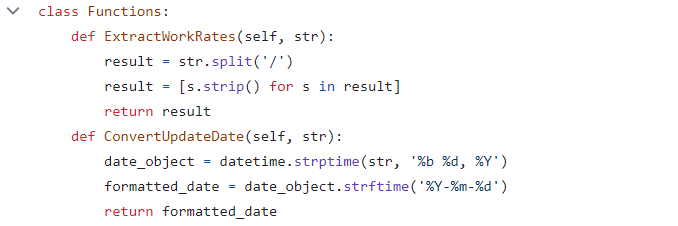
Với thư viện scrapy như đã đề cập ở phần 2, nhóm tiến hành thu thập dữ liệu như sau:

### 5.1.1. Thu thập các trận đấu với crawlMatches.py:

Tệp crawlMatches.py chịu trách nhiệm trong việc lấy dữ liệu của trận đấu trong giải bóng đá Ngoại hạng Anh từ mùa giải 2018-2019 đến mùa giải hiện tại 2023-2024.



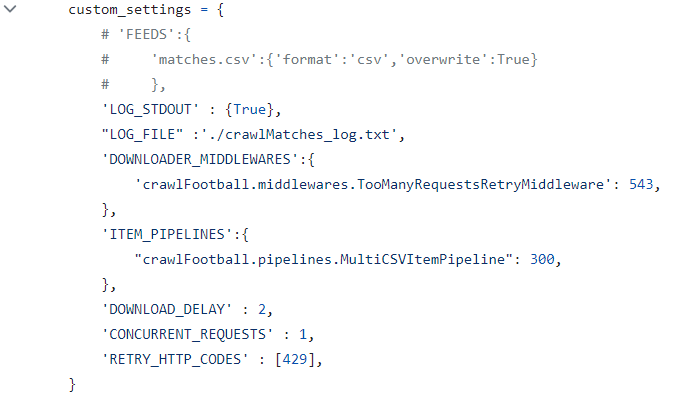




*Hình 5.1: Đối tượng Player và các hàm hỗ trợ*

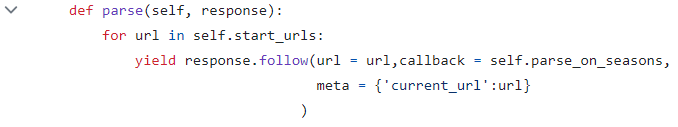
Sử dụng thư viện scrapy, cùng với tệp items.py và functions.py chứa các hàm thực thi trong cùng project. Tệp items.py giúp tạo ra các class lưu trữ thông tin khi cào từ trang web về, ví dụ như một class Player có các thuộc tính của một số thuộc tính của cầu thủ. Class Funtion cung cấp một số chức năng giúp xử lý dữ liệu thô như tách chuỗi, chuyển đổi kiểu dữ liệu sang DateTime,…





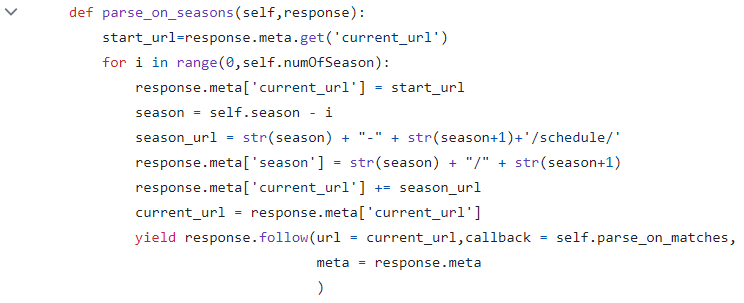
*Hình 5.2: Cấu hình Scrapy của tệp crawlMatches.py*

Với mỗi spider trong scrapy sẽ là một class, biến name giúp Scrapy xác định được spider, allowed\_domains giúp Scrapy không tải những phần tử khác nằm ở ngoài trang, start\_urls là mảng chứa những đường dẫn cần cào, custom\_settings là những cài đặt cần thiết cho spiders, ví dụ như đặt tên cho tệp xuất ra kết quả và định dạng, in ra nhật ký để theo dõi, cài đặt middleware và pipeline,…



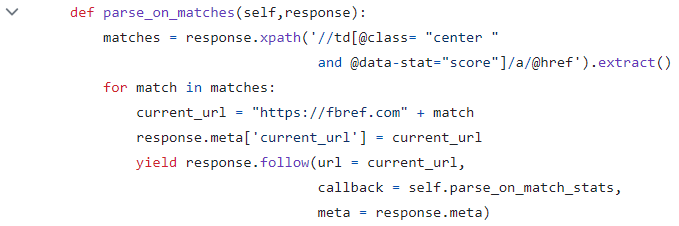
*Hình 5.3: Hàm giúp Scrapy truy cập tới Fbref*

Hàm parse bắt đầu cào trang web với mỗi đường dẫn nằm trong biến start\_urls và gọi tới hàm parse\_on\_seasons.



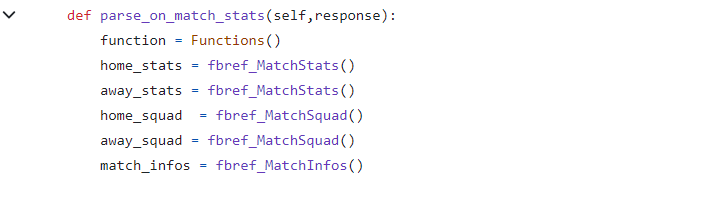
*Hình 5.4: Hàm giúp Scrapy truy cập tới Fbref theo mùa giải*

Hàm parse\_on\_seasons thực hiện chỉnh sửa đường dẫn theo từng mùa giải, sau đó gọi tới parse\_on\_matches.



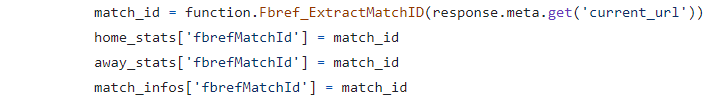
*Hình 5.5: Hàm giúp Scrapy truy cập tới Fbref theo các trận đấu*

Hàm parse\_on\_matches thực hiện lấy đường dẫn của từng trận đấu theo xpath, sau đó sẽ truy cập vào đường dẫn của từng trận đấu bằng cách gọi hàm parse\_on\_match\_stats.



*Hình 5.6: Hàm giúp Scrapy khởi tạo các đối tượng*

Hàm parse\_on\_match\_stats sẽ lấy toàn bộ thông tin của trận đấu. Ban đầu sẽ khởi tạo các đối tượng functions, home\_stats, away\_stats, home\_squad, away\_squad, match\_infos.



*Hình 5.7: Đoạn mã giúp Scrapy lấy match\_id*

Thực hiện lấy match\_id từ đường dẫn mà spider đang truy cập và gắn vào home\_stats, away\_stats, match\_infos.



*Hình 5.8: Đoạn mã giúp Scrapy lấy đường dẫn và mùa giải*

Thực hiện lấy đường dẫn và mùa giải vào đối tượng match\_infos.



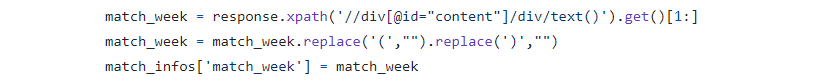
*Hình 5.9: Đoạn mã giúp Scrapy lấy tên của mùa giải*

Thực hiện lấy tên của giải đấu vào đối tượng match\_infos.



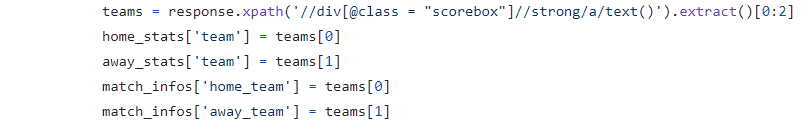
*Hình 5.10: Đoạn mã giúp Scrapy lấy ngày trận đấu diễn ra*

Thực hiện lấy thời gian diễn ra trận đấu vào đối tượng match\_infos.



*Hình 5.11: Đoạn mã giúp Scrapy lấy tuần trận đấu diễn ra*

Thực hiện lấy tuần diễn ra trận đấu của mùa giải đó vào đối tượng match\_week.



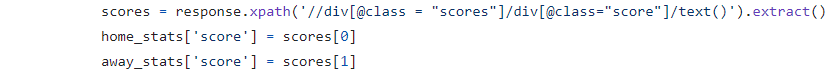
*Hình 5.12: Đoạn mã giúp Scrapy lấy tên của hai đội nhà và đội khách*

Thực hiện lấy tên của 2 đội vào đối tượng home\_stats, away\_stats và match\_infos.



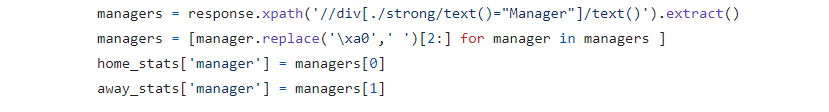
*Hình 5.13: Đoạn mã giúp Scrapy phân biệt đội nhà và đội khách*

Với đối tượng home\_stats, đặt is\_home\_team với giá trị là Yes, và với đối tượng away\_stats, đặt với giá trị là No



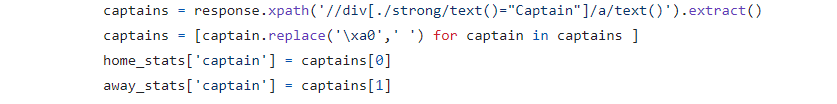
*Hình 5.14: Đoạn mã giúp Scrapy lấy tỉ số*

Với đối tượng home\_stats và away\_stats, đặt biến score theo số bàn thắng ghi được đối với đội nhà và đội khách.



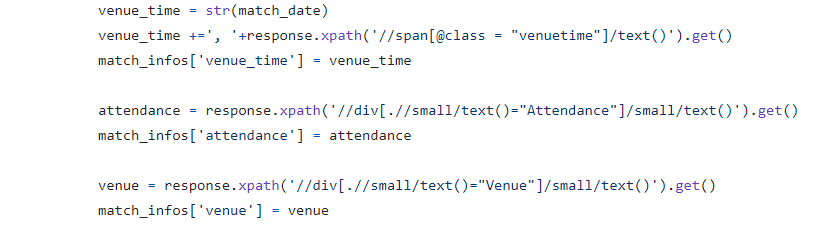
*Hình 5.15: Đoạn mã giúp Scrapy lấy tên của huấn luyện viên của hai đội*

Lấy tên huấn luyện viên của cả 2 đội.



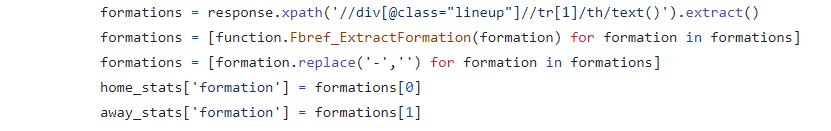
*Hình 5.16: Đoạn mã giúp Scrapy lấy tên của đội trưởng của hai đội*

Lấy tên đội trưởng của cả 2 đội.



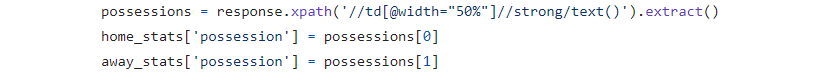
*Hình 5.17: Đoạn mã giúp Scrapy lấy các thông tin giờ diễn ra trận đấu, số lượng khán giả, …*

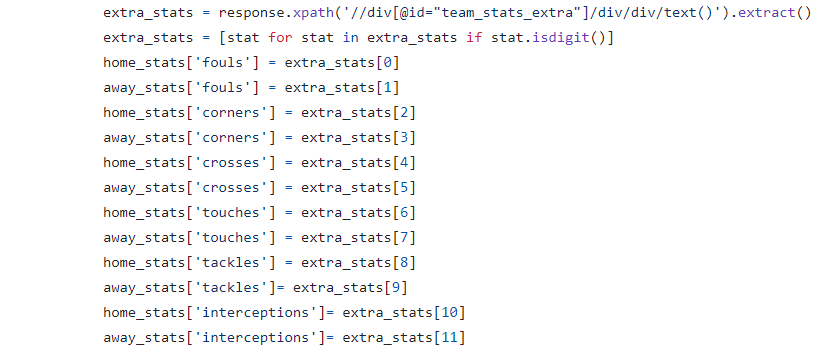
Với đối tượng match\_infos, lấy thời gian diễn ra trận đấu tại khu vực đó, số lượng người tham dự, địa điểm diễn ra trận đấu và tổ trọng tài.



*Hình 5.18: Đoạn mã giúp Scrapy lấy đội hình của cả hai đội*

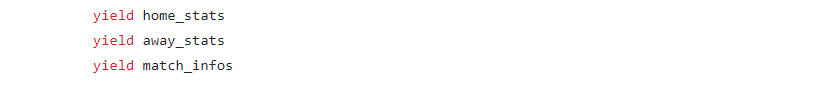
Đội hình của cả 2 đội được gán vào biến formation của đối tượng home\_stats và away\_stats.





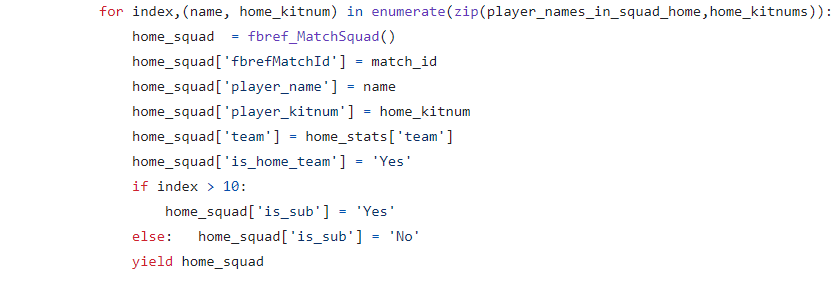
*Hình 5.19: Đoạn mã giúp Scrapy lấy các thông tin trong trận đấu của cả hai đội*

Thông số của 2 đội trong trận đấu được gắn vào home\_stats và away\_stats



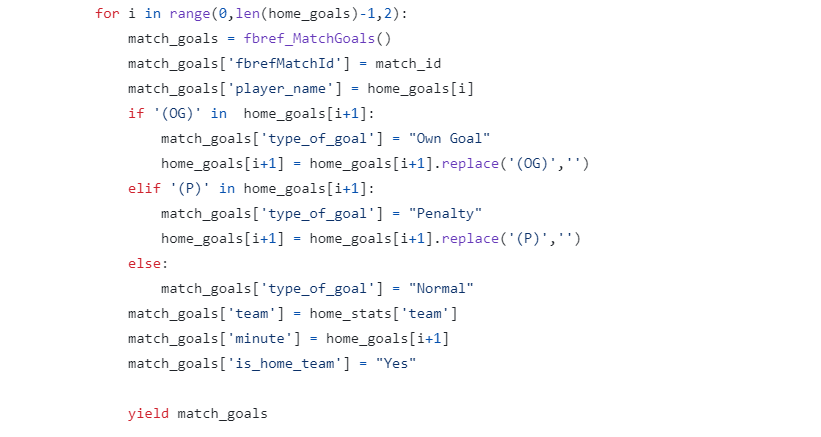
*Hình 5.20: Đoạn mã giúp Scrapy trả về kết quả*

Xuất kết quả ra tệp fbref\_MatchInfos.csv và fbref\_MatchStats.csv sau khi kết thúc lấy các thông số.



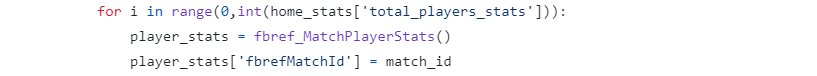
*Hình 5.21: Đoạn mã giúp Scrapy lấy tên, số áo, và biến xác định có phải là cầu thủ dự bị của các cầu thủ*

Thực hiện lấy tên và số áo mỗi cầu thủ của 2 đội. Tiếp tục phân loại ra cầu thủ đá chính và dự bị. Cuối cùng xuất ra tệp fbref\_MatchSquad.csv.



*Hình 5.22: Đoạn mã giúp Scrapy lấy thông tin bàn thắng*

Thực hiện việc lấy thông tin bàn thắng của đội nhà, tương tự đối với đội khách.



*Hình 5.23: Đoạn mã giúp Scrapy lấy*

Thực hiện việc lấy chỉ số của mỗi cầu thủ của đội nhà trong trận đấu, tương tự với mỗi cầu thủ đội sân khách.

### 5.1.2. Thu thập các cầu thủ với crawlPlayers.py:

Tệp tin crawlPlayers.py chịu trách nhiệm trong việc cào dữ liệu thông số của các cầu thủ trong trò chơi Fifa từ trang web Sofifa.



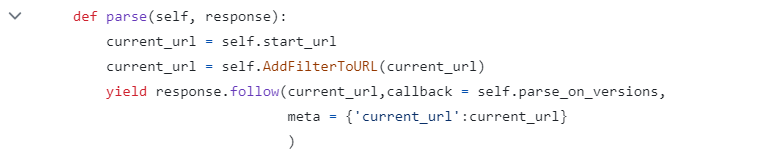
*Hình 5.24: Đoạn mã nhập thư viện sử dụng trong tệp crawlPlayers.py*

Sử dụng thư viện Scrapy và Pandas, tệp items.py giúp tạo ra các class lưu trữ thông tin khi cào từ trang web về.



*Hình 5.25: Cấu hình Scrapy cơ bản*

Khởi tạo các biến của spider như name, allowed\_domain,… leauges là số mùa giải của trang web Sofifa chỉ định, numOfVersions là số phiên bản của Sofifa.



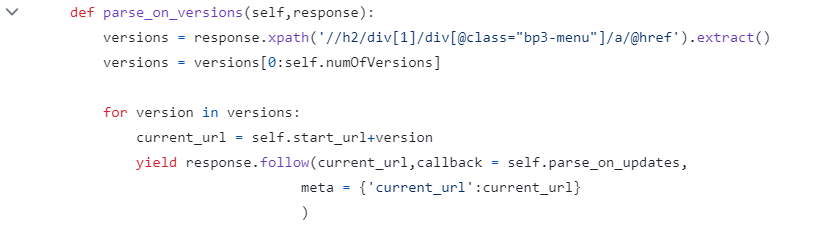
*Hình 5.26: Đoạn mã giúp Scrapy truy cập tới Sofifa*

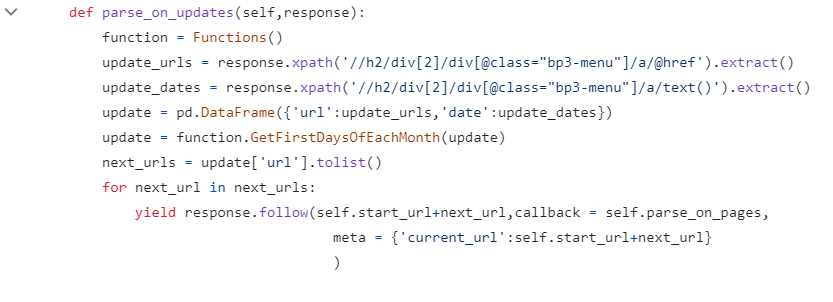
Hàm parse bắt đầu cào trang web, hàm này sẽ gọi AddFilterToURL điều chỉnh đường dẫn cho phù hợp với mùa giải.



*Hình 5.27: Đoạn mã giúp Scrapy truy cập Sofifa theo mùa giải*

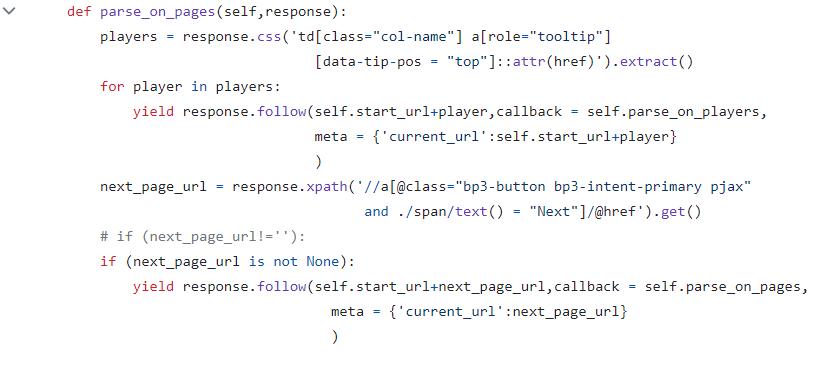
Hàm AddFilterToURL dùng để truyền biến là giải bóng đá thông qua đường dẫn do người dùng chỉ định.





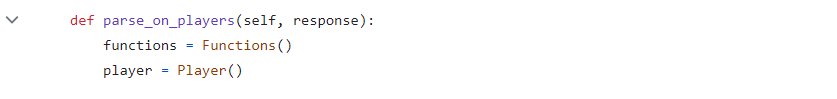
*Hình 5.28: Đoạn mã giúp Scrapy truy cập theo các phiên bản và các bản cập nhật Fifa*

Với mỗi phiên bản của Fifa và mỗi bản cập nhật, spider sẽ đều được đưa tới.



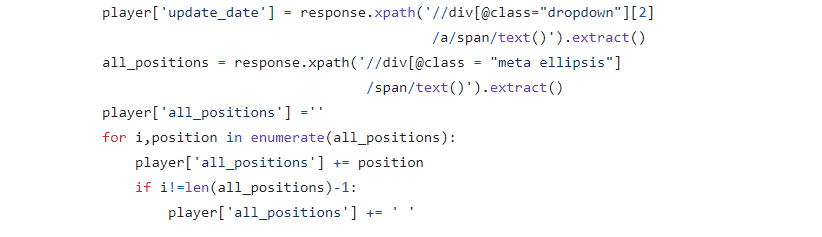
*Hình 5.29: Đoạn mã giúp Scrapy truy cập vào các trang danh sách cầu thủ*

Mỗi 60 cầu thủ sẽ nằm trên 1 trang nên parse\_on\_pages sẽ giúp spider duyệt qua tất cả các trang đó.



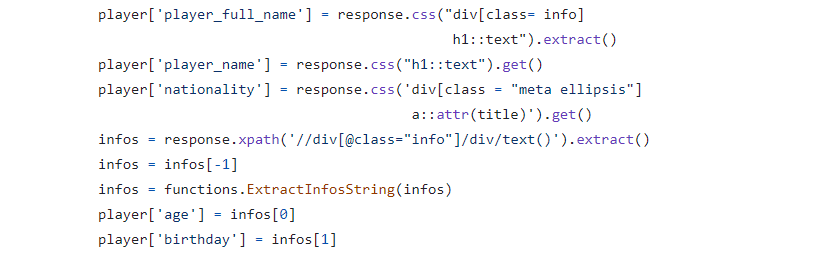
*Hình 5.30: Đoạn mã giúp Scrapy truy cập vào các trang danh sách cầu thủ*

Hàm parse\_on\_players sẽ khởi tạo đối tượng functions và player.



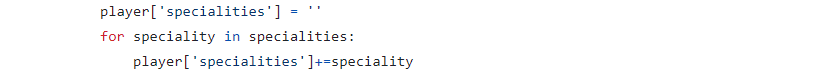
*Hình 5.31: Đoạn mã giúp Scrapy lấy ngày cập nhật và các vị trí của cầu thủ*

Thực hiện lấy ngày của bản cập nhật và tất cả vị trí của cầu thủ đó có thể chơi.



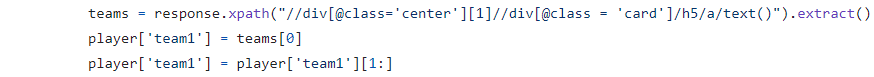
*Hình 5.32: Đoạn mã giúp Scrapy lấy thông tin cá nhân của cầu thủ*

Thực hiện lấy những thông tin cơ bản của cầu thủ như tên, tuổi, ngày sinh,…



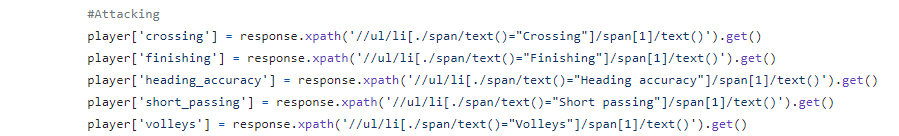
*Hình 5.33: Đoạn mã giúp Scrapy lấy đặc điểm riêng của cầu thủ*

Thực hiện lấy những đặc điểm riêng của cầu thủ.



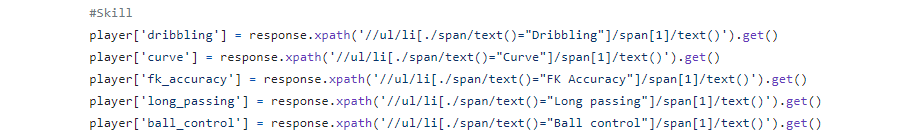
*Hình 5.34: Đoạn mã giúp Scrapy lấy câu lạc bộ hoặc hoặc đội tuyển của cầu thủ*

Thực hiện lấy tên câu lạc bộ hoặc đội tuyển mà cầu thủ đó đang thi đấu.



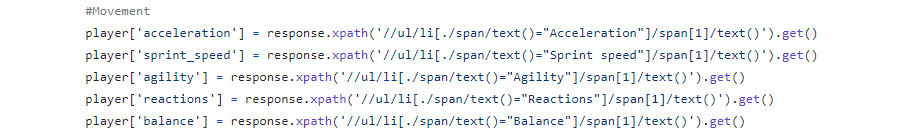
*Hình 5.35: Đoạn mã giúp Scrapy lấy chỉ số tấn công*

Thực hiện lấy chỉ số tấn công của cầu thủ.



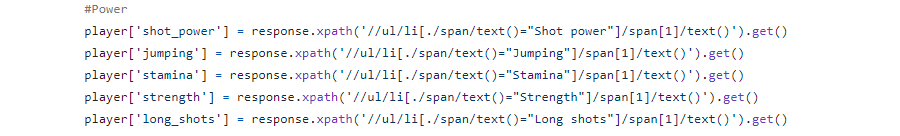
*Hình 5.36: Đoạn mã giúp Scrapy lấy chỉ số kĩ năng*

Thực hiện lấy chỉ số kĩ năng của cầu thủ.



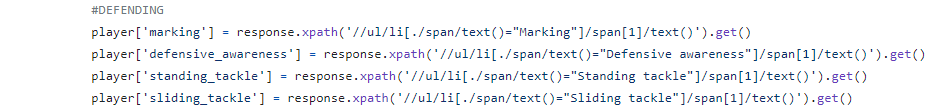
*Hình 5.37: Đoạn mã giúp Scrapy lấy chỉ số chuyển động*

Thực hiện lấy chỉ số chuyển động của cầu thủ.



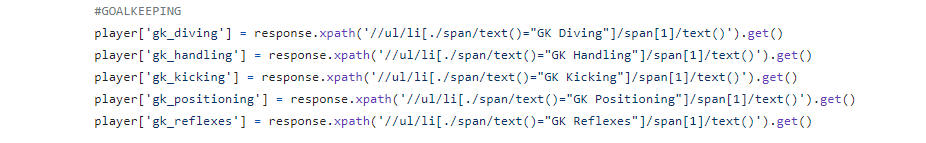
*Hình 5.38: Đoạn mã giúp Scrapy lấy chỉ số sức mạnh*

Thực hiện lấy chỉ số sức mạnh của cầu thủ.



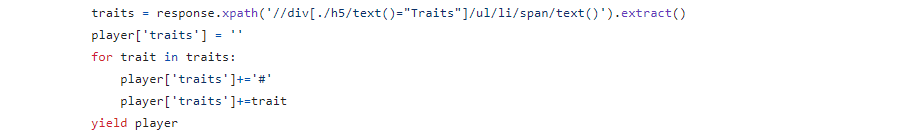
*Hình 5.39: Đoạn mã giúp Scrapy lấy chỉ số phòng thủ*

Thực hiện lấy chỉ số phòng ngự của cầu thủ



*Hình 5.40: Đoạn mã giúp Scrapy lấy chỉ số thủ môn*

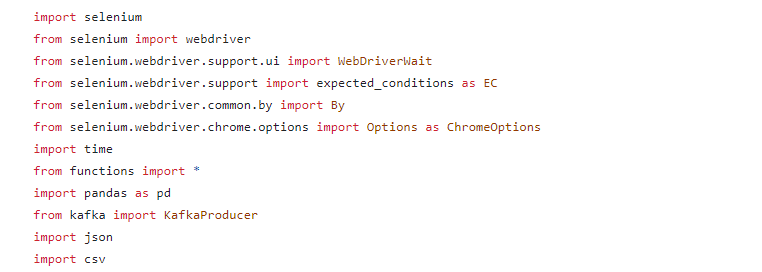
Thực hiện lấy chỉ số thủ môn cho cầu thủ là thủ môn



*Hình 5.41: Đoạn mã giúp Scrapy lấy kĩ năng*

Thực hiện lấy kĩ năng của cầu thủ.

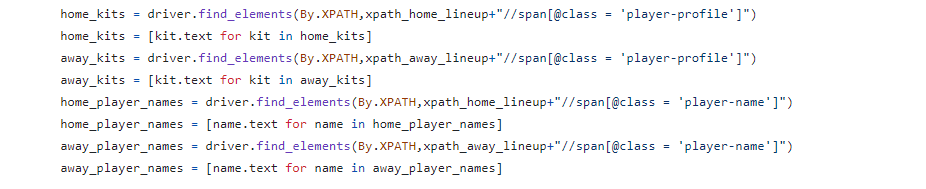
### 5.1.3. Thu thập các trận đấu tương lai với crawlUpComingMatch.py:



*Hình 5.42: Các thư viện sử dụng cho tệp crawlUpComingMatch.py*

Đối với tệp crawlUpComingMatch.py, nhóm sử dụng thư viện selenium để tiến hành cào trang web *Sportsgambler.com*, sau đó kết quả sẽ được xuất ra tệp CSV và cuối cùng gửi sang Kafka.





*Hình 5.43: Đoạn mã giúp Scrapy lấy các thông tin của trận đấu dự kiến*

Với mỗi ngày và với mỗi trận đấu, selenium sẽ lấy đội hình xuất phát ra sân dự kiến của cả 2 đội, bao gồm tên cầu thủ và số áo.

## 5.2. Tầng hàng đợi thông tin - Message Queuing Layer với Apache Kafka:

### 5.2.1. Sử dụng Apache Kafka để thu thập trận đấu mới:

Tầng hàng đợi thông tin sẽ được áp dụng để thu thập các trận đấu trong tương lai, tại tập tin crawlUpComingMatch.py, sẽ có hàm kafka\_producer để truyền dữ liệu từ tệp tin data.csv vào Kafka với chủ đề là UpComingMatch\_Topic và sẽ có tệp consumer.py tiến hành xử lí.

A screen shot of a computer code

Description automatically generated

*Hình 5.44: Gửi dữ liệu cho Kafka*

Consumer.py nhận dữ liệu từ Kafka qua cổng 9092 với chủ đề là UpComingMatch\_Topic, sau đó sẽ ép kiểu dữ liệu đó về lại dạng CSV.

A computer screen shot of text

Description automatically generated

*Hình 5.45: Nhận dữ liệu từ Kafka*

Consumer.py sẽ xuất ra định dạng CSV sau mỗi 3 giây nếu có dữ liệu.

A screen shot of a computer code

Description automatically generated

*Hình 5.46: Xuất dữ liệu từ Kafka ra CSV*

## 5.3. Tầng phân tích – Analysis Layer với Apache Spark và Pandas:

Dữ liệu về các trận đấu và cầu thủ được xử lý và phân tích thông qua 3 phần được trình bày ở bên dưới với 2 mục đích dùng cho thống kê và xây dựng mô hình dự đoán.

### 5.3.1. Chuẩn bị dữ liệu:

Ở phần này, dữ liệu về các trận đấu FBref và cầu thủ Sofifa sẽ được định dạng, sắp xếp, làm sạch, các bước xử lý được trình bày trong prepare.ipynb:



*Hình 5.47: Đường dẫn của tệp prepare.ipynb*



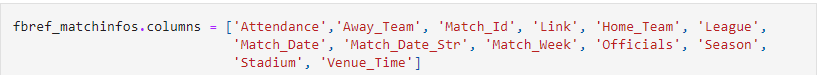
*Hình 5.48: Nhập thư viện cho tệp prepare.ipynb*

Thư viện pandas của Python được sử dụng để xử lý các file .csv, đưa về lớp DataFrame.



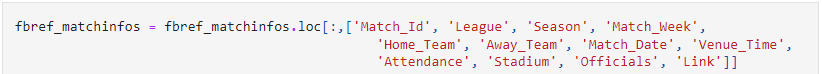
*Hình 5.49: Các biến dataframe*

Dữ liệu FBref sau khi được đọc trải qua các bước xử lý: đổi tên các trường, sắp xếp các trường, định dạng dữ liệu các trường, thay đổi giá trị các trường như ví dụ bên dưới:



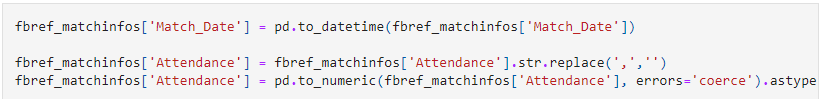
*Hình 5.50: Thay đổi tên cột cho fbref\_matchinfos*

Các trường được viết hoa chữ cái đầu tiên của mỗi tiếng, các tiếng cách nhau bởi dấu gạch dưới “\_”.



*Hình 5.51: Chọn ra các thuộc tính cần thiết cho fbref\_matchinfos*

Sử dụng hàm .loc để sắp xếp cũng như chọn lọc những trường cần thiết.



*Hình 5.52: Ép kiểu cho fbref\_matchinfos*

Đối với các trường có định dạng Date, chuyển về định dạng DateTime mặc định là mm/dd/yyyy. Một số trường được đưa về dạng numeric ví dụ như Attendance: 46,890 (số lượng khán giả) bằng cách bỏ đi dấu phẩy “,”.



*Hình 5.53: Fbref\_Infos sau quá trình chuẩn bị dữ liệu*

 *Hình 5.54: Tạo dataframe mới*

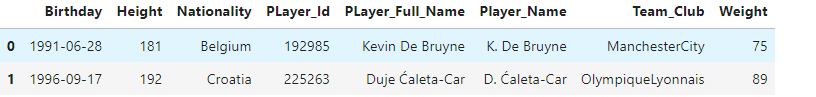
Tiếp đến là dữ liệu các cầu thủ Sofifa cũng được trải qua các bước xử lý tương tự như dữ liệu các trận đấu FBref:



*Hình 5.55: Hình ảnh sau khi chuẩn hoá dữ liệu*

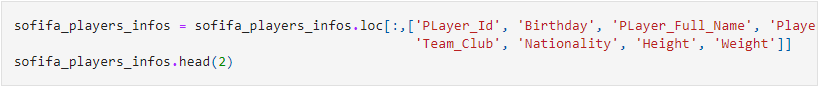


*Hình 5.56: Thay đổi tên cột cho sofifa\_players\_infos*

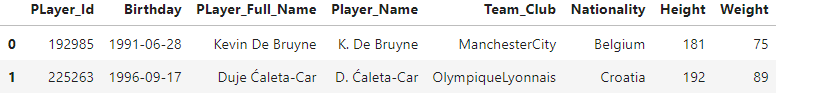


*Hình 5.57: Dữ liệu sau khi được chuẩn hoá*

Các trường được viết hoa chữ cái đầu tiên và cách nhau bởi dấu gạch dưới.

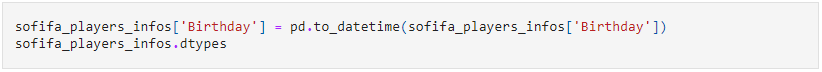


*Hình 5.58: Chọn ra các thuộc tính cần thiết*



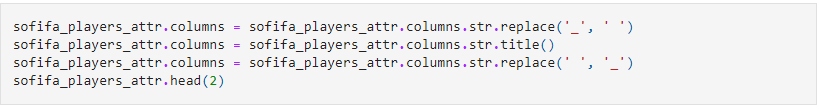
*Hình 5.59: Dữ liệu sau khi được chọn lọc và sắp xếp*

Sắp xếp các trường với Player\_Id đầu tiên.

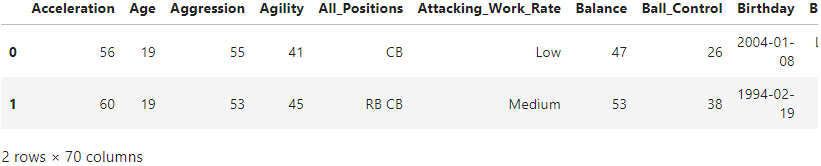


*Hình 5.60: Ép kiểu các thuộc tính trong sofifa\_players\_infos*

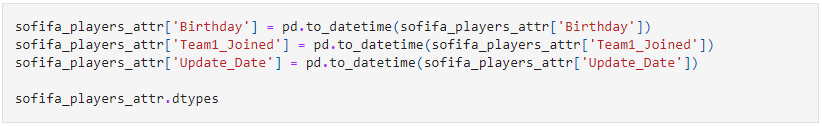
Chuyển đổi kiểu dữ liệu DateTime cho trường Birthday (ngày sinh).



*Hình 5.61: Thay đổi tên thuộc tính sofifa\_players\_attr*

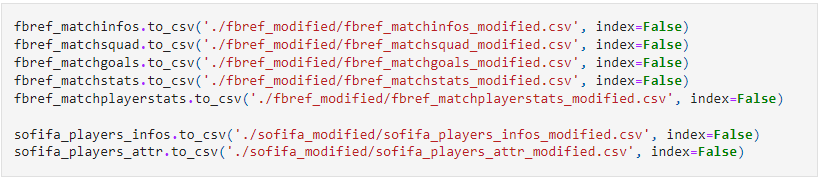


*Hình 5.62: Hình ảnh sau khi thay đổi tên thuộc tính*

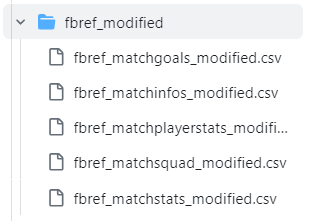


*Hình 5.63: Ép kiểu thuộc tính sofifa\_players\_attr*

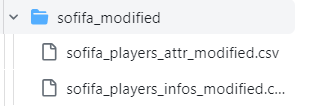
Với tập player\_attr.csv có tới 70 trường cũng được định dạng với chữ cái đầu tiên được viết hoa và có 3 trường cần được chuyển đổi DateTime là Birthday (ngày sinh), Team1\_Joined (ngày tham gia CLB) và Update\_Date (phiên bản cầu thủ).



*Hình 5.64: Xuất kết quả ra CSV*



*Hình 5.65: Kết quả các tệp CSV liên quan tới Fbref*



*Hình 5.66: Kết quả các tệp CSV liên quan tới Sofifa*

Ứng với 5 tệp csv các trận đấu FBref và 2 tệp các cầu thủ Sofifa được đọc, sau quá trình chuẩn bị dữ liệu ta có 7 tệp tương ứng và được lưu trong thư mục fbref\_modified và sofifa\_modified.

### 5.3.2. Phân tích các thông số cần dùng cho thống kê:

Phần này sẽ kết hợp các cầu thủ có trong đội hình mỗi trận đấu FBref với các thông tin chi tiết hơn của cầu thủ trong Sofifa, quá trình xử lý được thể hiện trong analysis.ipynb:





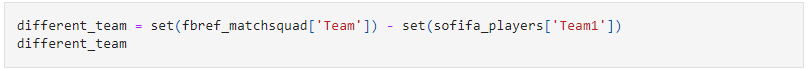
*Hình 5.67: Nhập thư viện*

Ngoài thư viện pandas được dùng để xử lý DataFrame, numpy được sử dụng để kiểm tra dữ liệu bị NaN (bị thiếu).



*Hình 5.68: Tạo dataframe mới*

Fbref\_matchsquad chứa các thông tin về các cầu thủ cả đá chính và dự bị trong trận đấu (đã được đề cập ở phần 3 – Mô tả tập dữ liệu) được bổ sung thêm Match\_Date (ngày diễn ra trận đấu) từ Fbref\_matchinfos. Và cuối cùng được kết hợp với Sofifa để lấy thêm chi tiết thông tin từng cầu thủ.



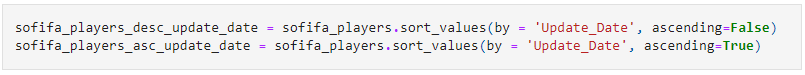


*Hình 5.69: Đoạn mã giúp tìm ra các biến chưa khớp*



*Hình 5.70: Thay thế các biến chưa khớp*

Có một sự khác biệt về tên câu lạc bộ giữa tập dữ liệu FBref và Sofifa là “Bournemouth”, ta tiến hành thay thế tên câu lạc bộ đó.



*Hình 5.71: Sắp xếp theo thời gian*

Các cầu thủ Sofifa được sắp xếp theo thứ tự tăng và giảm dần các phiên bản tùy vào mục đích sử dụng.



*Hình 5.72: Hàm tìm ra ngày cập nhật gần nhất*

Hàm ClosestUpdateDate() trả về phiên bản gần nhất các cầu thủ trong đội hình, ví dụ trận đấu diễn ra vào ngày 20/11/2023 thì phiên bản các cầu thủ gần nhất là 04/11/2023. Hàm SofifaPlayersEqualUpdateDate() trả về tất cả các cầu thủ trong một phiên bản, ví dụ tất cả các cầu thủ trong phiên bản 04/11/2023.



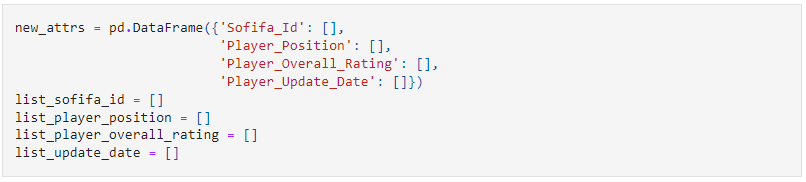
*Hình 5.73: Đoạn mã giúp tìm cầu thủ tương ứng*

Nhóm tiến hành tìm cầu thủ theo 2 phương pháp: tìm theo số áo và câu lạc bộ với hàm SofifaPlayer() và tìm theo tên cầu thủ SofifaPlayer\_ByName().



*Hình 5.74: Hàm xử lí những cầu thủ chưa khớp*

Hàm PlayerNotFound() tiến hành tìm cầu thủ trong trường hợp không tìm thấy (thực tế là cầu thủ đó chưa gia nhập câu lạc bộ).



*Hình 5.75: Tạo ra các danh sách và dataframe để lưu trữ*

Với thông tin chi tiết của một cầu thủ trong Sofifa, ta cần quan tâm đến 4 thuộc tính: Sofifa\_Id (Id cầu thủ Sofifa), Player\_Position (vị trí cầu thủ), Player\_Overall\_Rating (chỉ số đánh giá), Player\_Update\_Date (phiên bản cầu thủ).

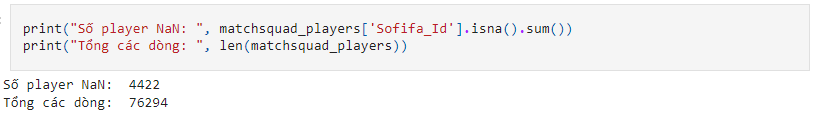




*Hình 5.76: Đoạn mã giúp tìm cầu thủ tương ứng*

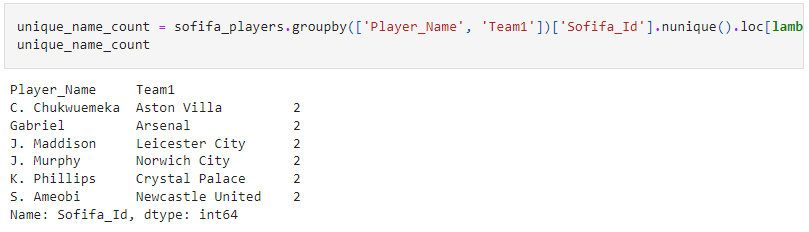
Với mỗi cầu thủ trong mỗi trận đấu FBref, nhóm tiến hành tìm kiếm và bổ sung như sau:

* Gắn Match\_Date (ngày diễn ra trận đấu) vào match\_date\_i: ví dụ 20/11/2023
* Tìm phiên bản cầu thủ gần nhất Sofifa gắn vào closest\_update\_date\_i: ví dụ 04/11/2023
* Lấy tên câu lạc bộ và số áo cầu thủ trong trận đó, lưu vào team\_fbref\_i và player\_kitnum\_fbref\_i.
* Lấy tất cả cầu thủ Sofifa ở phiên bản 04/11/2023 lưu vào biến sofifa\_players\_this\_update\_version.
* Tìm cầu thủ theo phương pháp tìm theo số áo và câu lạc bộ bằng cách gọi hàm Sofifa\_Player()
* Nếu tìm thấy cầu thủ thì bổ sung 4 thuộc tính như đã đề cập ở trước.



*Hình 5.77: Hiển thị kết quả*

Với phương pháp tìm theo câu lạc bộ và số áo, kiểm tra thì thấy có 4422 trong số 76294 các cầu thủ không tìm thấy, như đã đề cập ở trước thì ta còn phương pháp nữa là tìm theo tên.



*Hình 5.78: Đoạn mã tìm các cầu thủ trùng tên trong đội*

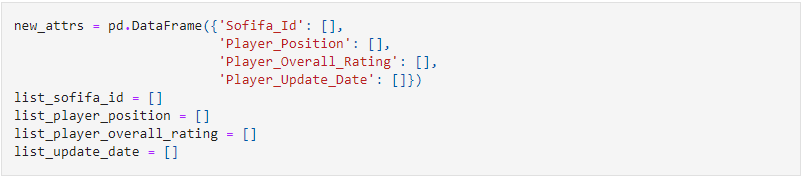
Liệt kê tất cả các cầu thủ bị trùng tên và trùng câu lạc bộ trong Sofifa thì có 6 cầu thủ, con số rất nhỏ so với toàn cầu thủ nên có thể chấp nhận một vài sự nhầm lẫn này.

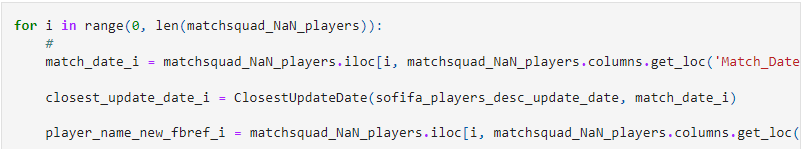


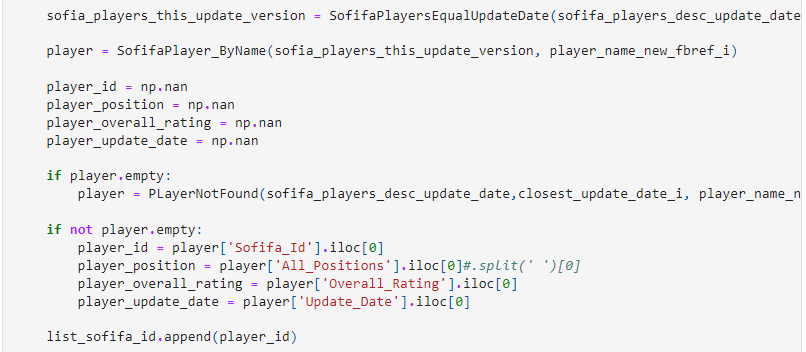


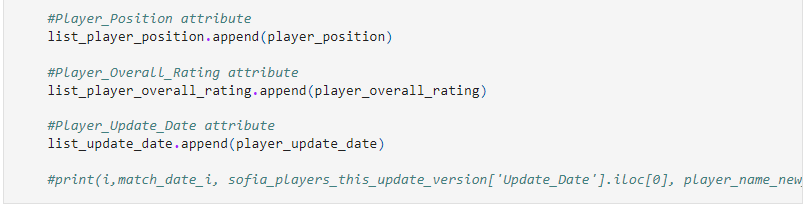
*Hình 5.79: Tạo DataFrame chứa cầu thủ bị thiếu thông tin.*

Tạo một DataFrame chứa 4422 cầu thủ bị thiếu thông tin.









*Hình 5.80: Bổ sung các cầu thủ bị thiếu chỉ số*

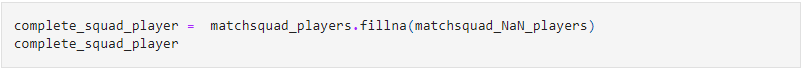
Với phương pháp tìm cầu thủ theo tên, nhóm tiến hành xử lý như ở cách bổ sung trước, chỉ khác là sử dụng phương pháp tìm theo tên – hàm SofifaPlayer\_ByName() và tìm cầu thủ nếu không có ở phiên bản hiện tại – hàm PlayerNotFound() :

* Gắn Match\_Date (ngày diễn ra trận đấu) vào match\_date\_i: ví dụ 20/11/2023
* Tìm phiên bản cầu thủ gần nhất Sofifa gắn vào closest\_update\_date\_i: ví dụ 04/11/2023
* Lấy tên câu lạc bộ và số áo cầu thủ trong trận đó, lưu vào team\_fbref\_i và player\_kitnum\_fbref\_i.
* Lấy tất cả cầu thủ Sofifa ở phiên bản 04/11/2023 lưu vào biến sofifa\_players\_this\_update\_version.
* Tìm cầu thủ theo phương pháp tìm theo tên bằng cách gọi hàm Sofifa\_Player(), nếu không có ở phiên bản hiện tại thì tìm ở phiên bản tiếp theo với hàm PlayerNotFound().
* Nếu tìm thấy cầu thủ thì bổ sung 4 thuộc tính như đã đề cập ở trước.



*Hình 5.81: Hiển thị kết quả sau khi xử lí*

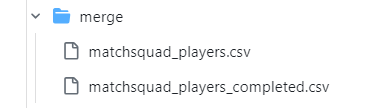
Số lượng các cầu thủ không được tìm thấy giảm đi đáng kể, xuống còn 850 trên tổng số 76294 cầu thủ.



*Hình 5.82: Bổ sung các giá trị bị thiếu của cầu thủ*



*Hình 5.83: Xuất kết quả ra CSV*



*Hình 5.84: Hình ảnh tệp kết quả CSV*

Tiến hành bổ sung các cầu thủ bị NaN và xuất kết quả ra tệp matchsquad\_players\_completed.csv trong thư mục merge.



*Hình 5.85: Kết hợp các thông tin cầu thủ trong Sofifa với các cầu thủ mỗi trận đấu trong FBref.*

### 5.3.3. Kết hợp các thông số sử dụng cho model:

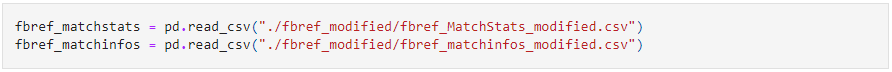
Phần này sẽ kết hợp các thuộc tính từ nhiều bảng để tạo thành một bảng duy nhất có đầy đủ các input và output phục vụ cho việc xây dựng mô hình dự đoán, được thể hiện trong model\_dataset.ipynb:





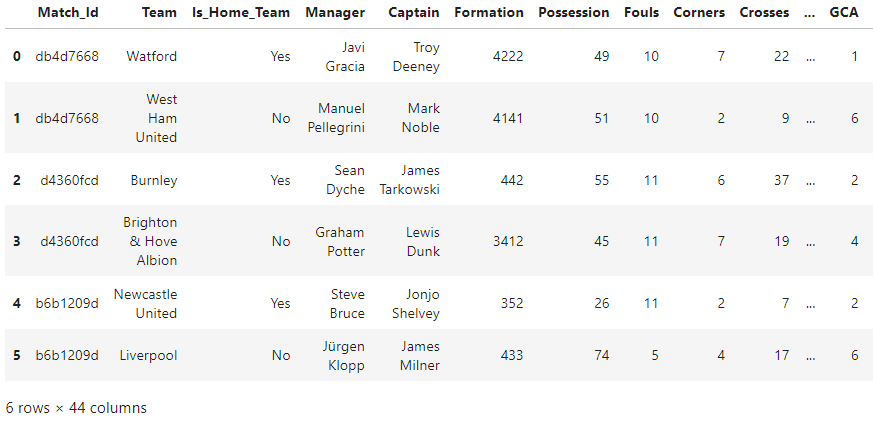
*Hình 5.86: Nhập thư viện*

Thư viện pandas tiếp tục được sử dụng để xử lý.

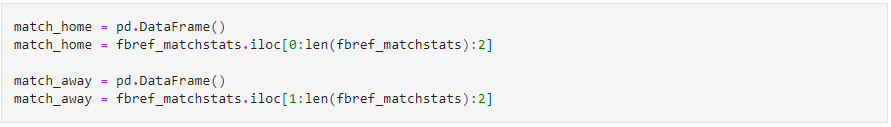


*Hình 5.87: Tạo dataframe*

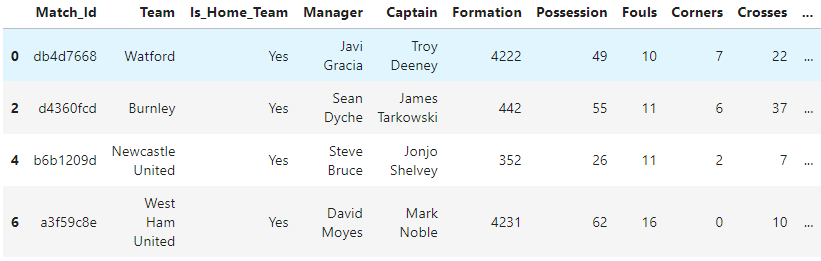
Sử dụng fbref\_matchinfos để lấy Match\_Date (ngày thi đấu), fbref\_matchstats chứa nhiều các thuộc tính dùng cho input.



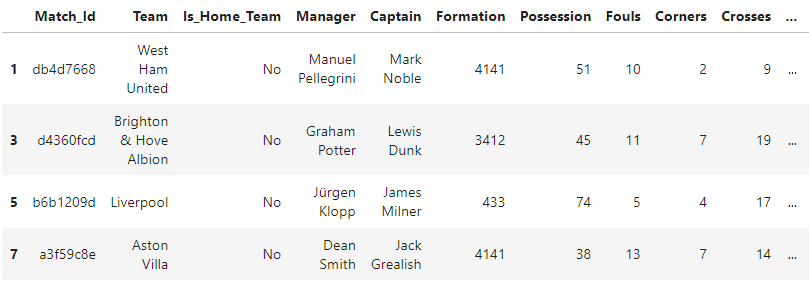
*Hình 5.88: FBref\_MatchStats bao gồm các thông số dạng numeric mô tả thông tin tổng kết của hai đội bóng trong mỗi trận đấu.*



*Hình 5.89: Tách dữ liệu cho sân nhà và sân khách*

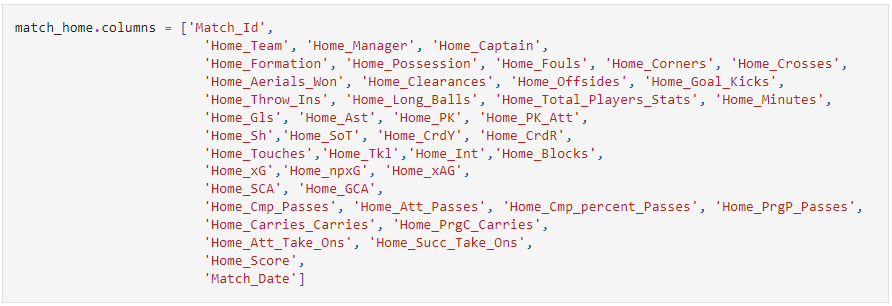


*Hình 5.90: Hình ảnh sau khi tách ra sân nhà*



*Hình 5.91: Hình ảnh sau khi tách sân khách*

Với mỗi 2 dòng trong tập dữ liệu các trận đấu ban đầu ứng với thông tin tổng kết của đội sân nhà và đội sân khách, nhóm tiến hành tách thành 2 thành phần: thông tin tổng kết đội sân nhà với các dòng chẵn (0, 2, 4,…) và thông tin tổng kết đội sân khách với các dòng lẻ (1, 3, 5,…)





*Hình 5.92: Đặt tên lại các thuộc tính*

Đổi tên các thuộc tính ứng với các thông số cho đội nhà và đội khách.



*Hình 5.93: Liên kết tập dữ liệu đội nhà và đội khách*

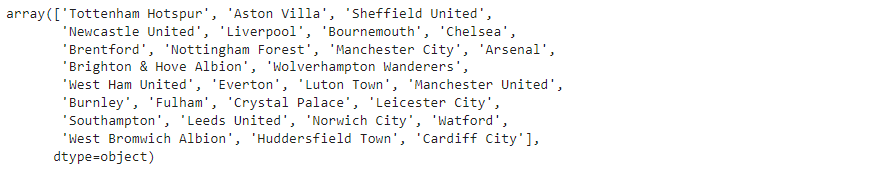
Kết hợp tất cả các thuộc tính đội nhà và đội khách thành một row duy nhất cho một trận đấu.



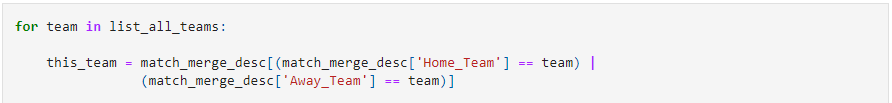
Có tổng cộng 1990 trận đấu tương ứng với 1990 dòng.

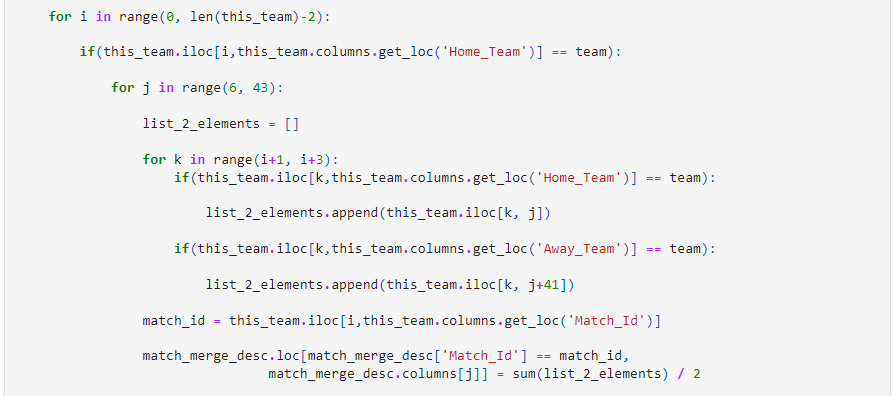


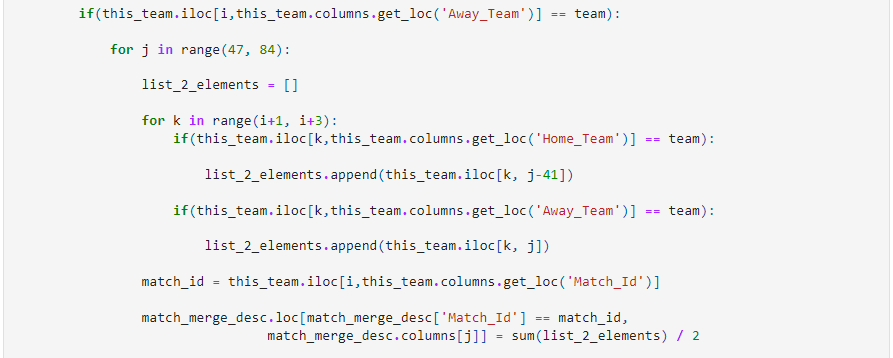


 *Hình 5.94: Sắp xếp các trận đấu theo thời gian giảm dần*

Sắp xếp các trận đấu theo thứ tự giảm dần thời gian Match\_Date và liệt kê tất cả các team tham gia vào tập các trận đấu.



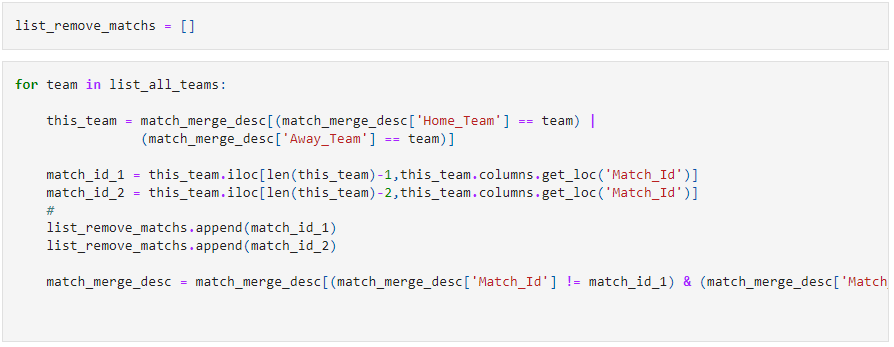




*Hình 5.95: Tính trung bình cộng của thông số của 2 trận đấu trước*

Với mỗi team xuất hiện trong tập dữ liệu, các thông số tổng kết sẽ được thay thế với trung bình cộng của 2 trận trước đó, nhóm tiến hành xử lý thông qua các bước:

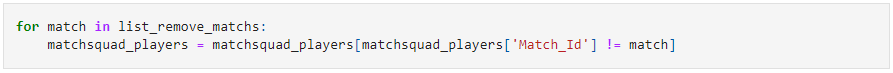
* Tổng hợp tất cả các trận đấu của một team vào this\_team.
* Lặp tất cả các trận đấu của team đó từ mới nhất đến cũ nhất, bỏ qua 2 trận cũ nhất vì 2 trận này không có thông số của 2 trận trước đó nữa.
* Với từng trận đấu trong vòng lặp, có thể team đó là sân nhà – Home\_Team hoặc sân khách Away\_Team. Nếu là Home\_Team thì thay đổi các thông số tổng kết của đội sân nhà từ thuộc tính từ index thứ 6 đến 42 (tổng cộng 37 thuộc tính), ngược lại, nếu là Away\_Team thì thay đổi các thuộc tính index từ 47 đến 83 (tổng cộng cũng là 37 thuộc tính).



*Hình 5.96: Loại bỏ 2 trận đấu cũ nhất của mỗi đội*

Như đã đề cập ở bước trước thì 2 trận cũ nhất không có thông số của 2 trận trước đó nữa nên sẽ bị loại bỏ, trận đấu cuối cùng có index -1, trận đấu kế cuối có index -2.





*Hình 5.97: Loại bỏ ra khỏi tập dữ liệu*

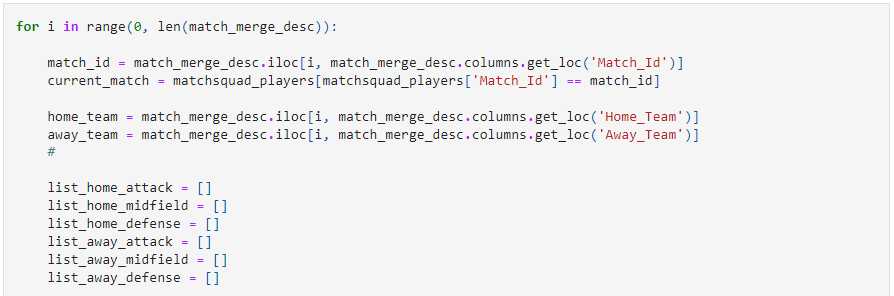
Sử dụng tập dữ liệu đã được kết hợp cầu thủ FBref và Sofifa ở phần 4.3.2 (hình ), và ta cũng không quên loại bỏ 2 trận đấu cuối cùng mỗi team đã lưu trong list\_remove\_matchs.

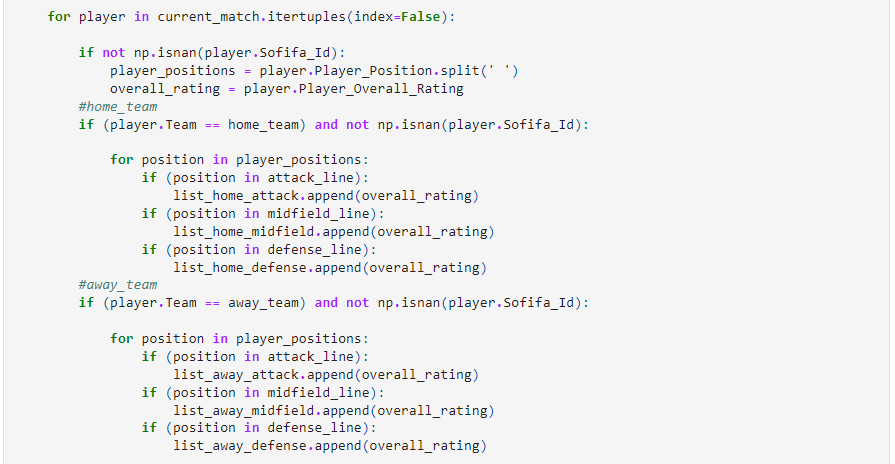


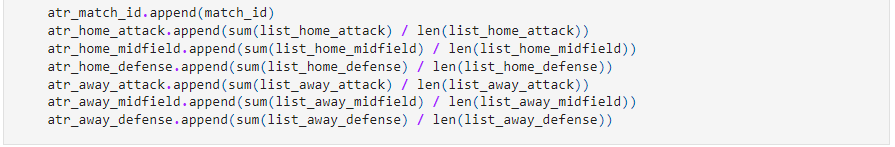
*Hình 5.98: Đoạn mã giúp chia vị trí các cầu thủ*

Nhóm tiến hành chia các vị trị của cầu thủ trong Sofifa thành 3 loại:

* Attack\_Line: bao gồm các vị trí tấn công.
* Midfield\_Line: bao gồm các vị trí ở hàng tiền vệ.
* Defense\_Line: bao gồm các vị trí phòng thủ.



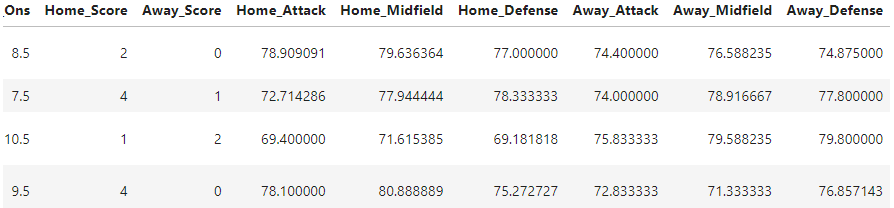




*Hình 5.99: Tính sức mạnh các cầu thủ theo vị trí*

Với 3 loại đánh giá thông số đội hình, nhóm tiến hành cập nhật 3 loại đó tương ứng với tất cả các cầu thủ góp mặt trong trận đấu của mỗi team như sau:

* Tách các vị trí có thể chơi của một cầu thủ (bởi một cầu thủ có thể chơi ở nhiều vị trí ví dụ “CB LB”) vào player\_positions.
* Ứng với mỗi vị trí, bổ sung Overall\_Rating (chỉ số đánh giá cầu thủ) vào một trong 3 loại attack, midfield, defense tùy thuộc vào vị trí đó nằm ở loại nào như đã chia ở phần trước.
* Sau khi đã tổng hợp 3 loại, tính trung bình cộng với mỗi loại ứng với đội nhà và đội khách.

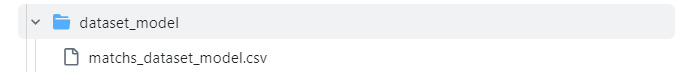


*Hình 5.100: Hình ảnh sau khi tính toán*

Tổng cộng có thêm 6 thuộc tính được bổ sung cho mỗi trận đấu bao gồm Home\_Attack, Home\_Midfield, Home\_Defense, Away\_Attack, Away\_Midfield, Away\_Defense.



*Hình 5.101: Xuất ra kết quả*



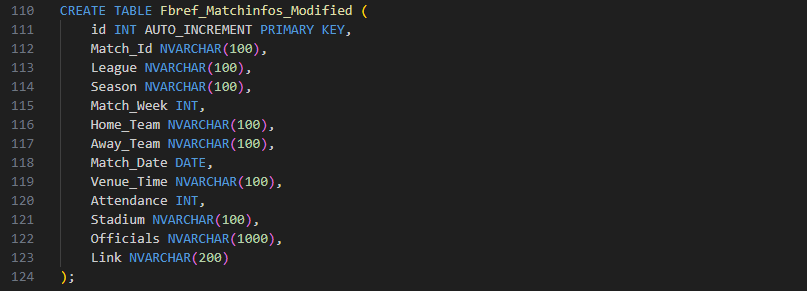
*Hình 5.102: Tệp CSV kết quả*

Cuối cùng, tập dữ liệu về các trận đấu được lưu vào tệp csv matchs\_dataset\_model nằm trong thư mục dataset\_model.

## 5.4. Tầng lưu trữ – Storage Layer với MySQL:

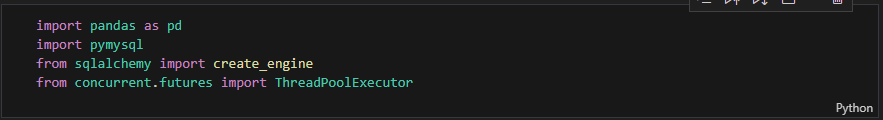
Dữ liệu về cầu thủ và các trận đấu sau khi đã hoàn tất xử lý và phần tích sẽ được lưu trữ ở trong Hệ quản trị cơ sở dữ liệu MySQL.



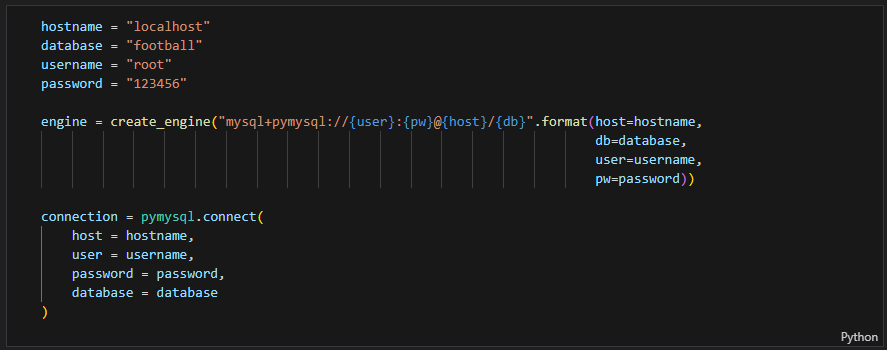


Hình: Minh họa một schema về thông tin trận đấu (MatchInfos) trong MySQL

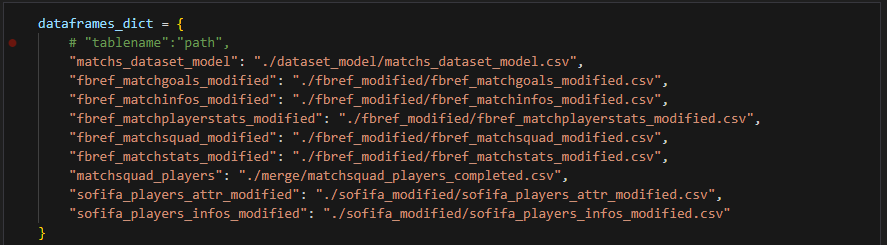
Đầu tiên nhóm tiến hành tạo các schema trong MySQL, dữ liệu sau này sẽ được nhập vào các schema tương ứng.



*Hình 5.103: Thư viện sqlalchemy tạo engine kết nối MySQL và ThreadPoolExecutor tạo các luồng*



*Hình 5.104: Tạo kết nối MySQL với hostname, database, user và password*

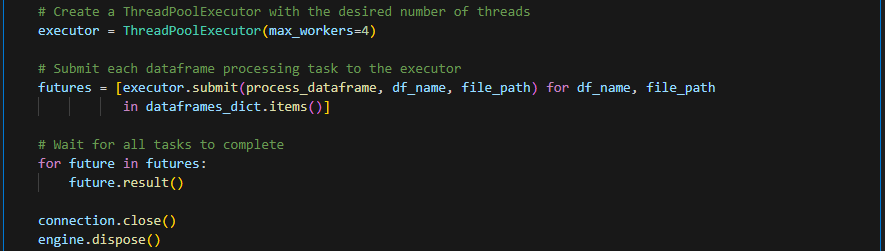


*Hình 5.105: Lưu các đường dẫn các file csv*



*Hình 5.106: Hàm đọc các file csv và nhập vào các schema*

Mỗi schema tương ứng với một file csv chứa data đã được xử lý ở các bước trước, đoạn mã tiến hành đọc từng file và nhập vào MySQL



*Hình 5.107: Tạo các luồng thực thi*

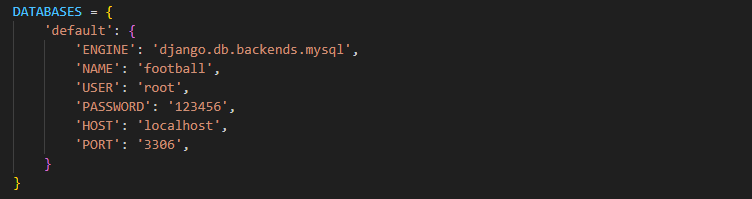
ThreadPoolExecutor thực hiện 4 worker như là các máy thực thi giúp tiến hành đẩy nhanh data vào MySQL, sau khi thực hiện xong thì ngắt kết nối và.

## 5.5. Tầng truy cập – Data Access Layer với Python Django và AngularJS:

Dữ liệu lịch sử về các trận đấu và cầu thủ đã được lưu ở Tầng lưu trữ nhóm tiến hành xây dựng một dashboard (trang hiển thị) như là một giao diện cho người dùng truy cập và theo dõi.

### 5.5.1. Xây dựng nền tảng truy cập dữ liệu với Python Django:

Với framework Django của Python, ta có thể sử dụng nó như là một ứng dụng backend xử lý các yêu cầu request, truy cập với một hệ quản trị cơ sở dữ liệu và trả về dữ liệu được yêu cầu.



*Hình 5.108: Tạo kết nối với hệ cơ sở dữ liệu MySQL*



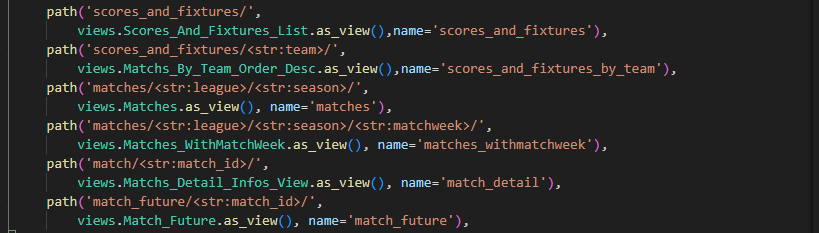
*Hình 5.109: Các module (thành phần) trong settings.py*

Đầu tiên nhóm tiến hành tạo kết nối với MySQL, sau đó là cấu hình các module như là các thành phần cấu tạo nên ứng dụng, một số module đặc trưng:

* Admin: trang admin quản lý ứng dụng
* Corsheaders: xử lý các yêu cầu từ các nguồn
* API: ứng dụng xử lý các request



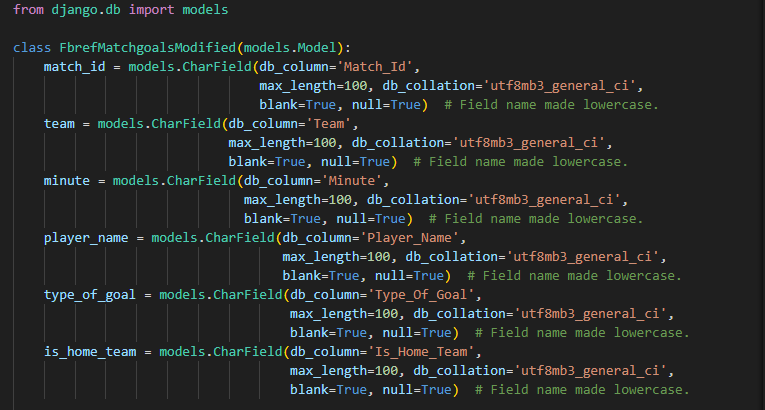
*Hình 5.110: 2 ứng dụng chính của Django*



*Hình 5.111: Xử lý các yêu cầu cho các trận đấu trong view.py*

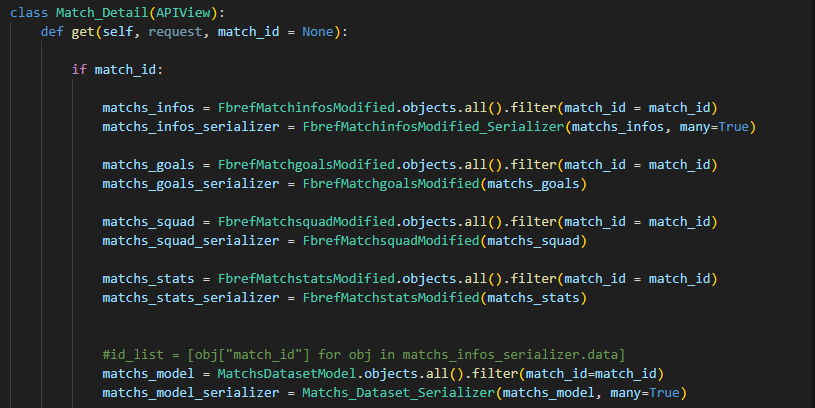
Các yêu cầu truy cập các trận đấu bao gồm 4 loại:

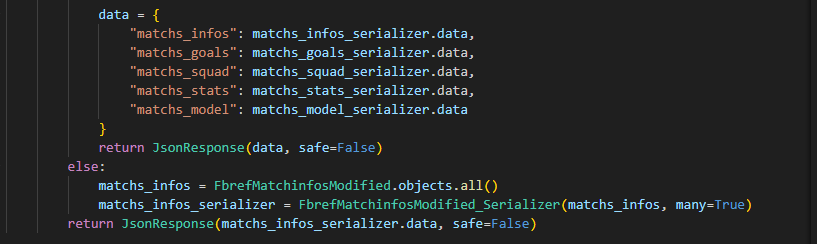
* Scores\_and\_fixtures: liệt kê mọi trận đấu
* Matches/League/Season: liệt kê các trận đấu theo giải đấu, mùa giải
* Matches/League/Season/MatchWeek: liệt kê các trận đấu theo giải đấu, mùa giải, tuần thi đấu
* Match/MatchId: thông tin chi tiết cho một trận đấu
* MatchFuture/MatchId: thông tin trận đấu sắp diễn ra



*Hình 5.112: Mô tả một model MatchGoals trong models.py*

Trong Django có một lớp trừu tượng ánh xạ các trường tương ứng trong schema MySQL, nhóm tiến hành tạo tất cả các model ứng với các schema như một model mô tả các bàn thắng trong trận đấu (MatchGoals) ở hình trên.





*Hình 5.113: Thực hiện lấy thông tin của trận đấu*

Sau khi nhận yêu cầu request, Django thực hiện truy cập MySQL, tiến hành xử lý theo yêu cầu và trả về kết quả là một dạng đối tượng JSON.

### 5.5.2. Xây dựng trang theo dõi và thống kê với AngularJS:

Nhóm sử dụng AngularJS như là một giao diện website cho người dùng có thể xem các thông tin về các trận đấu và cầu thủ một cách trực quan.



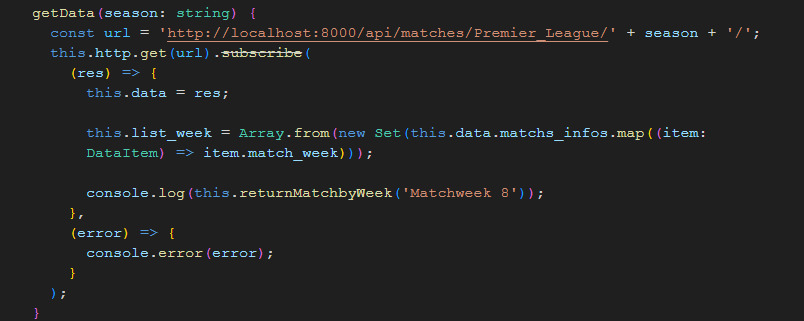




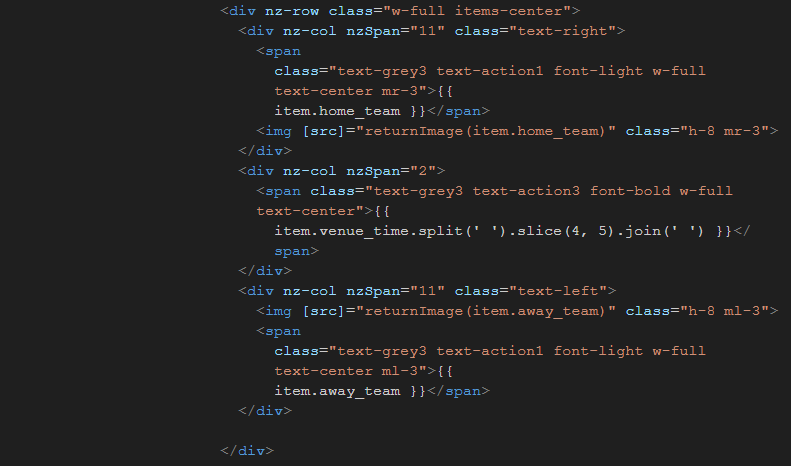
*Hình 5.114: Các thành phần cấu chức năng*

Đầu tiên nhóm xây dựng các thành phần bao gồm 4 nhóm chính, mỗi nhóm sẽ gửi yêu cầu đến Django ở phần trước, nhận về dữ liệu dạng JSON và hiển thị kết quả:

* Fixtures: giao diện tất cả các trận đấu theo mùa giải
* League: giao diện các giải đấu
* Matches: giao diện tất cả các trận đấu lịch sử
* Matches\_future: giao diện các trận đấu sắp diễn ra

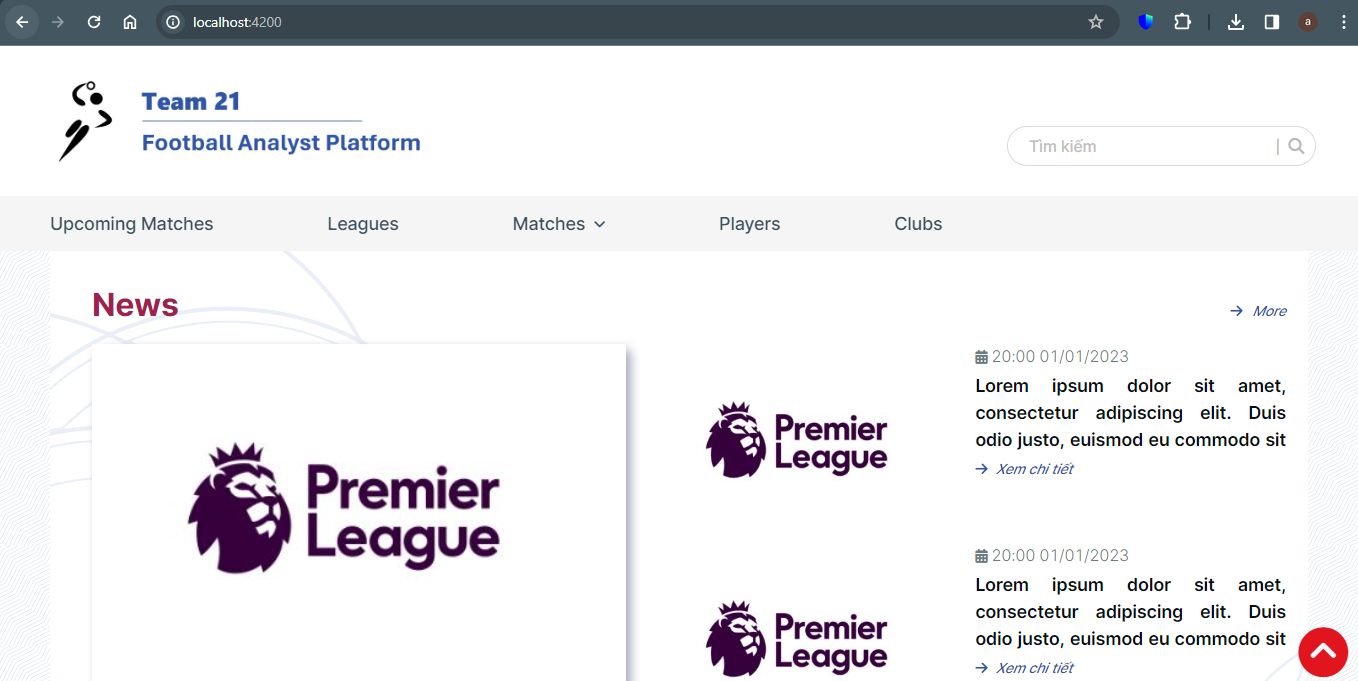


*Hình 5.115: Đoạn mã JS gửi request yêu cầu Django trả về tất cả các trận đấu JSON*

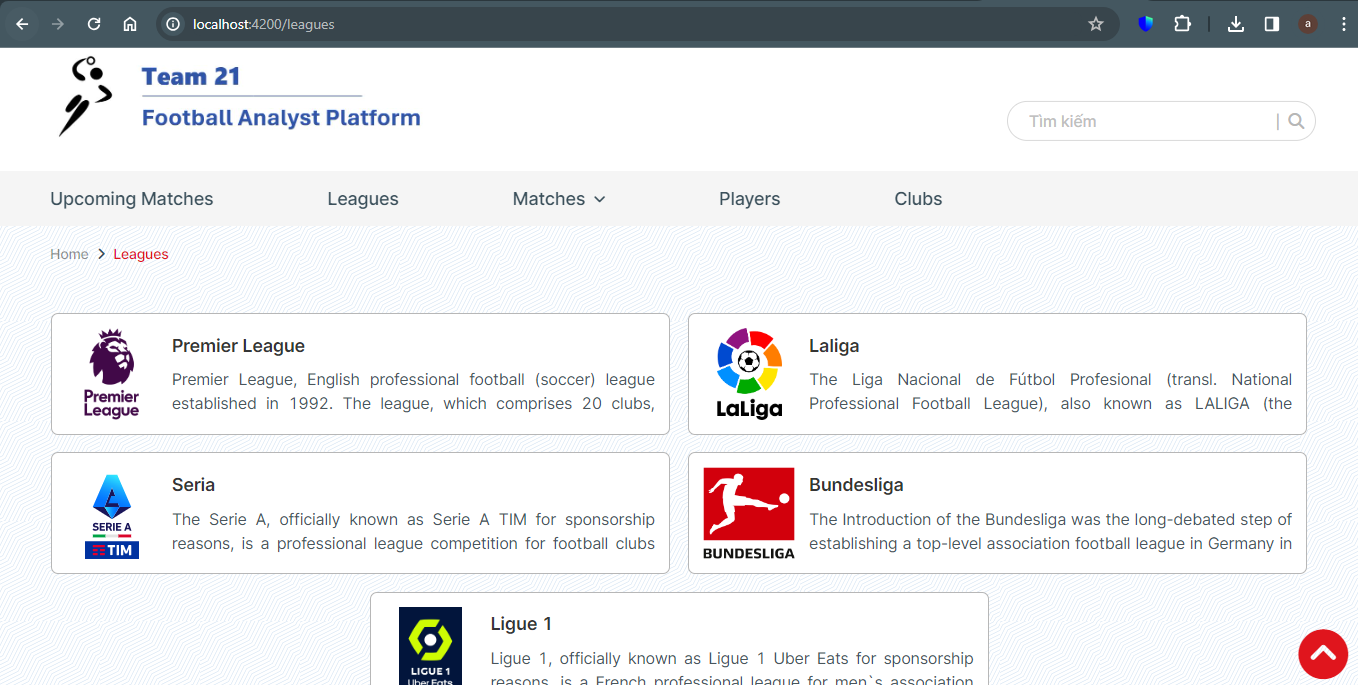


*Hình 5.116: Đoạn mã xử lý dữ liệu JSON và xuất ra HTML*

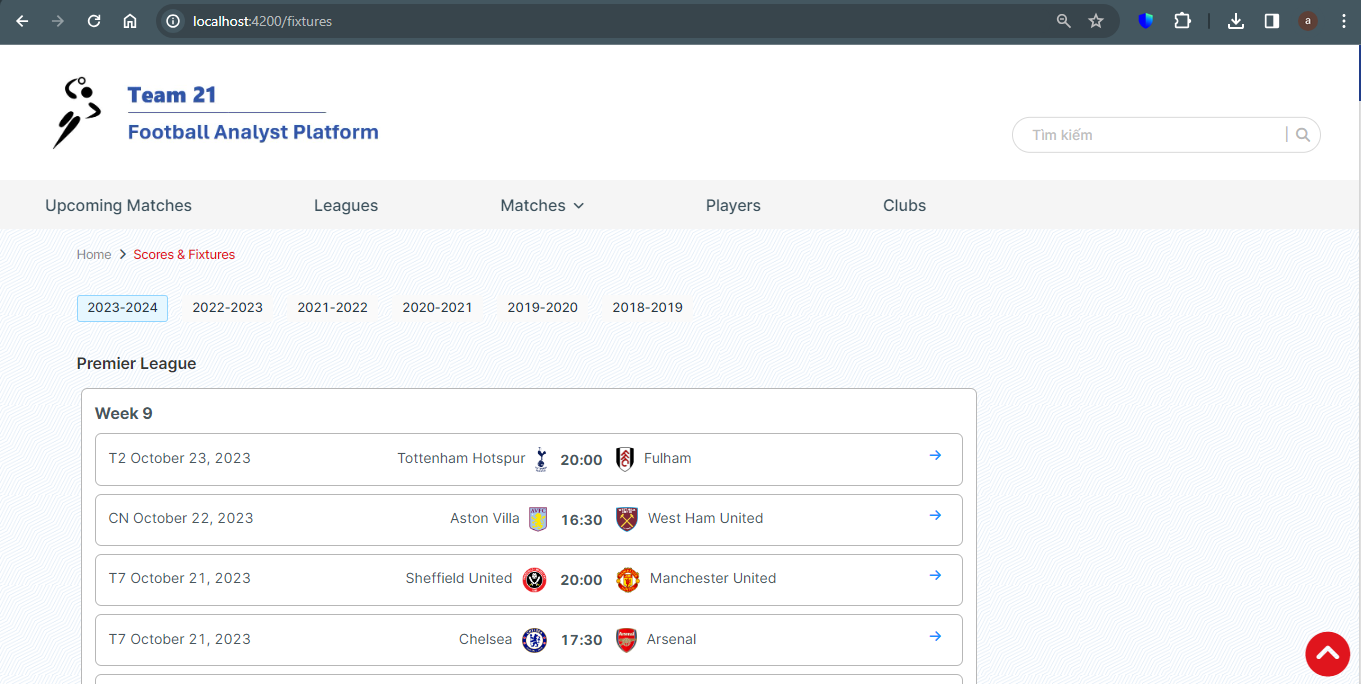
Ở mỗi thành phần đều có mã thực thi gửi các request đến Django yêu cầu trả về dữ liệu tương ứng, dữ liệu được thể hiện dưới dạng một đối tượng JSON, sau đó được xuất ra giao diện HTML.



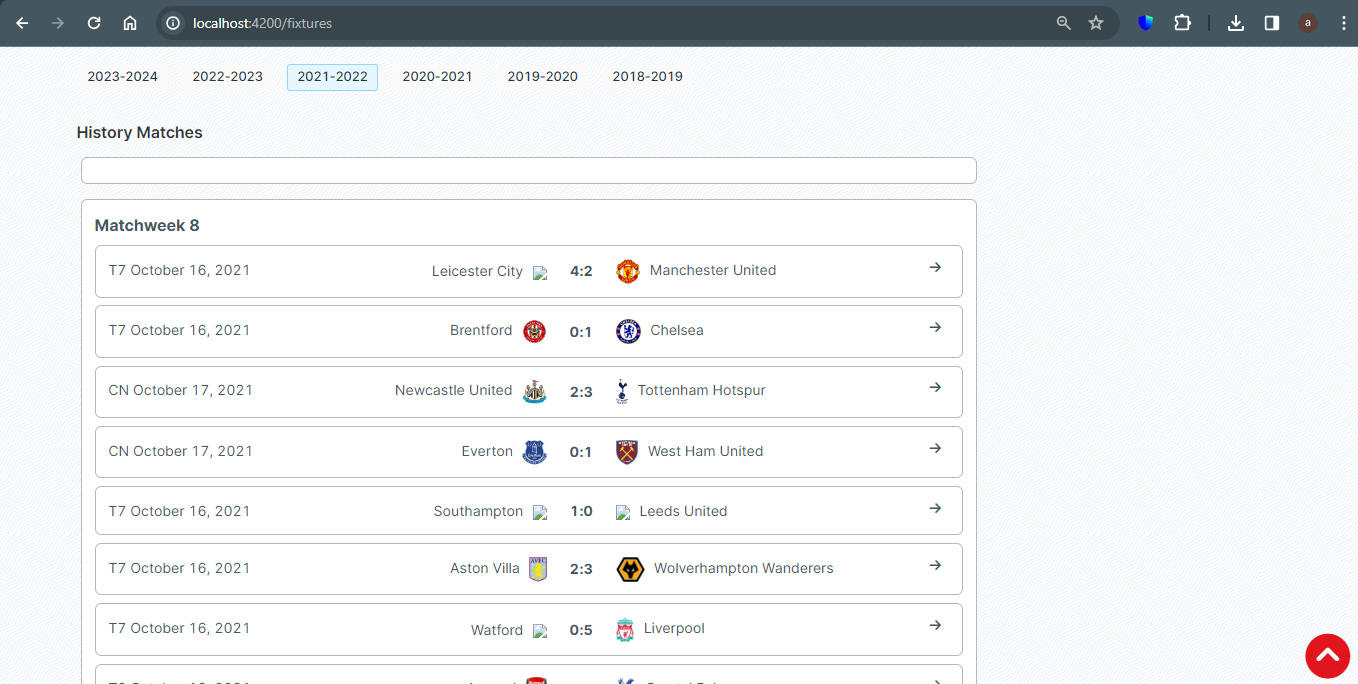
*Hình 5.117: Giao diện trang chủ*



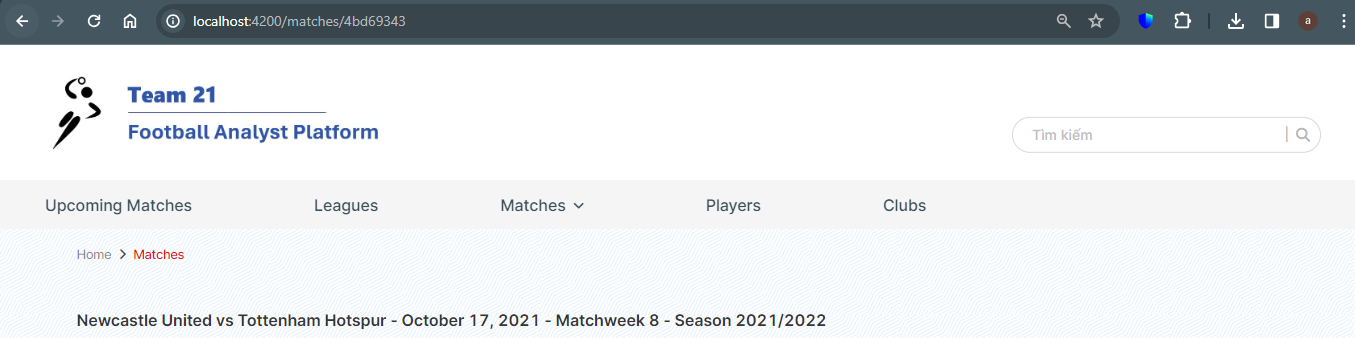
*Hình 5.118: Giao diện các giải đấu*



*Hình 5.119: Giao diện các trận đấu sắp diễn ra*

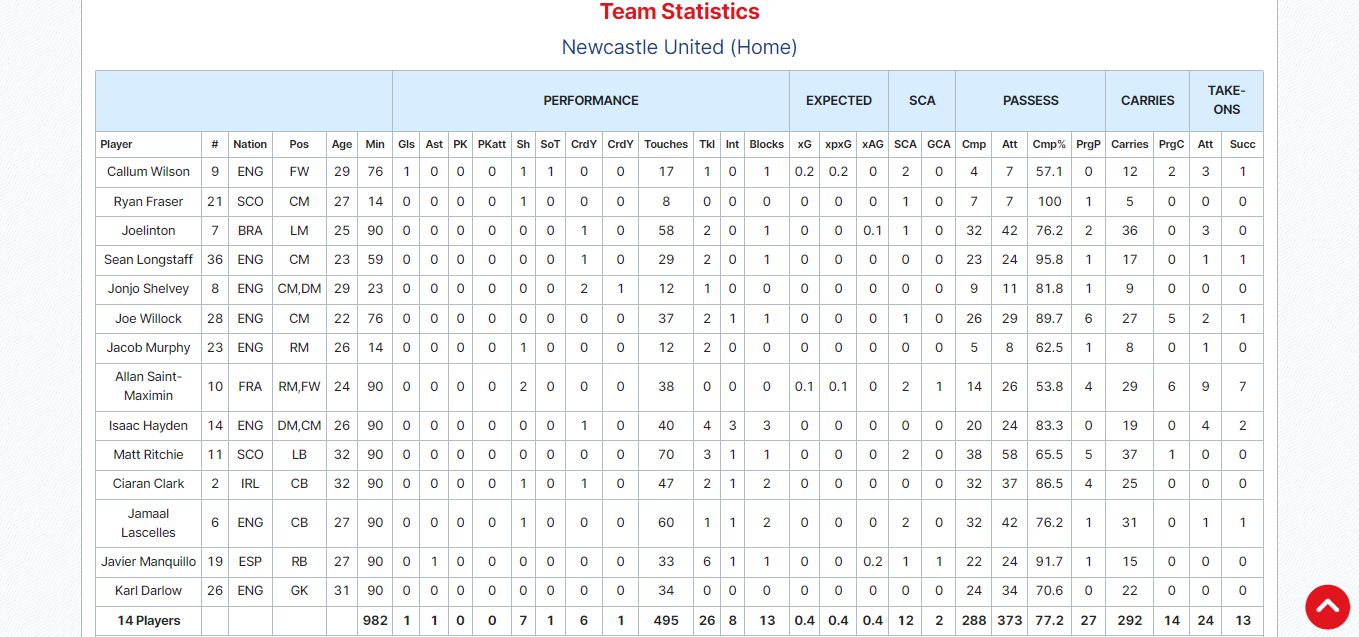


*Hình 5.120: Giao diện các trận đấu lịch sử*

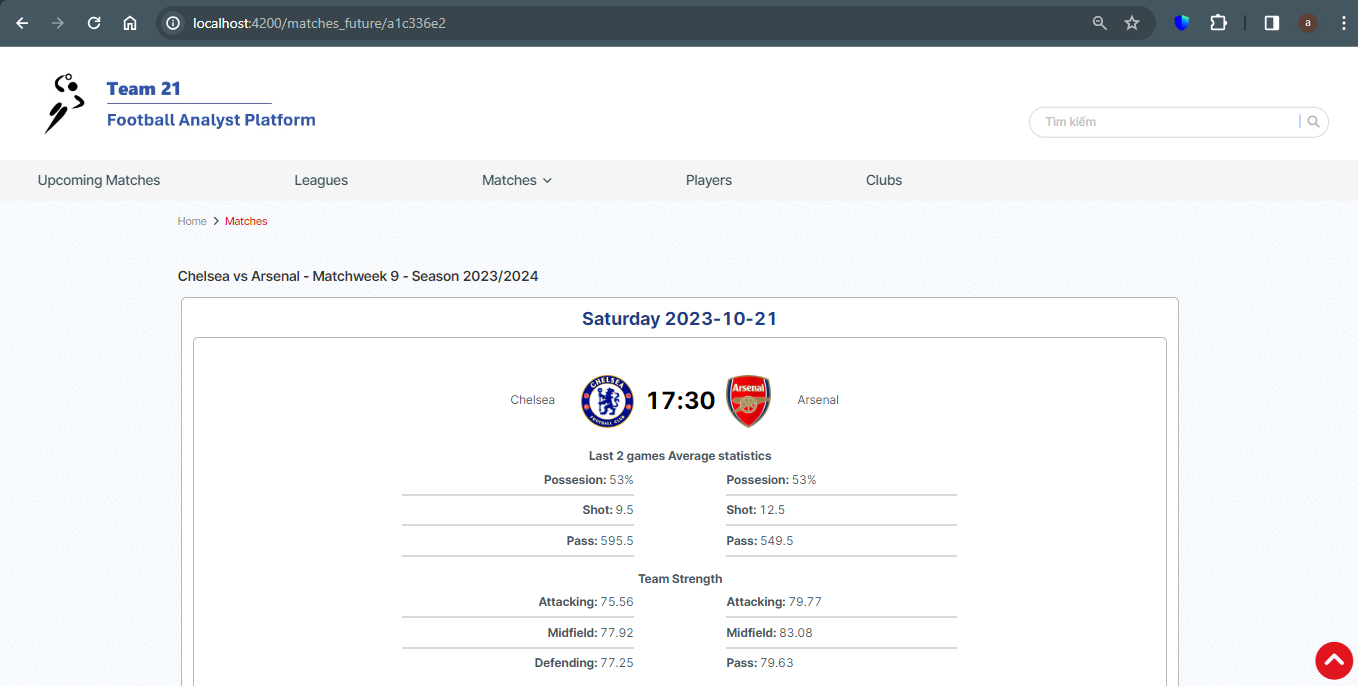


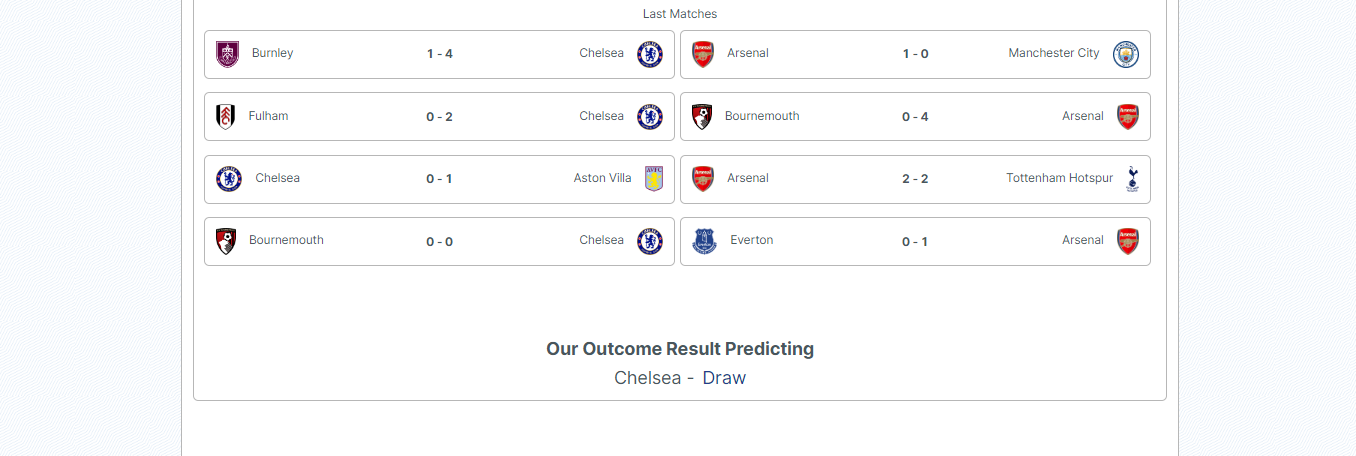


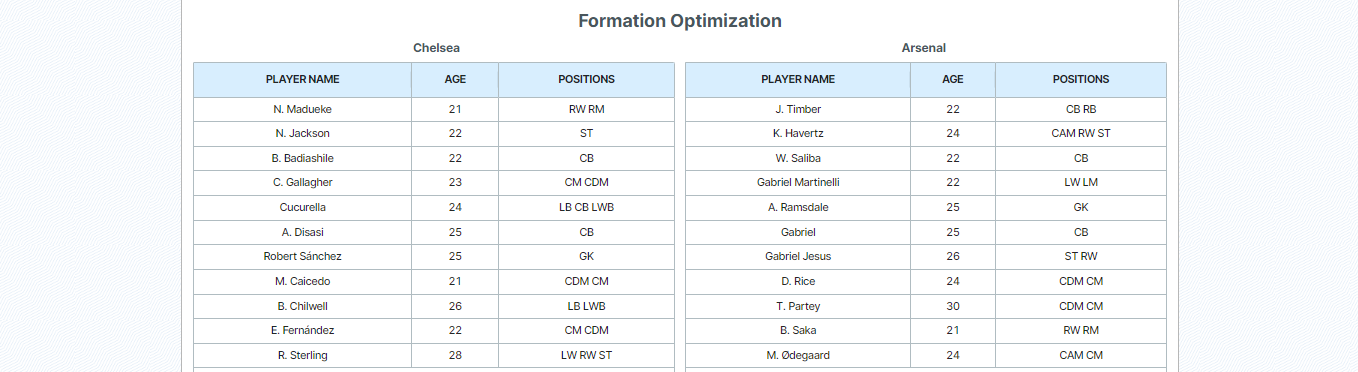




*Hình 5.121: Giao diện các thông tin chi tiết của trận đấu lịch sử*







*Hình 5.122: Giao diện thống kê một trận đấu sắp diễn ra và đưa ra dự đoán cùng đội hình tối ưu*

# PHẦN 6: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

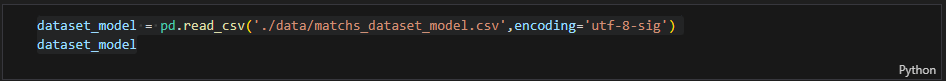
# CÁC TRẬN ĐẤU VÀ MÔ HÌNH TỐI ƯU HÓA

# ĐỘI HÌNH

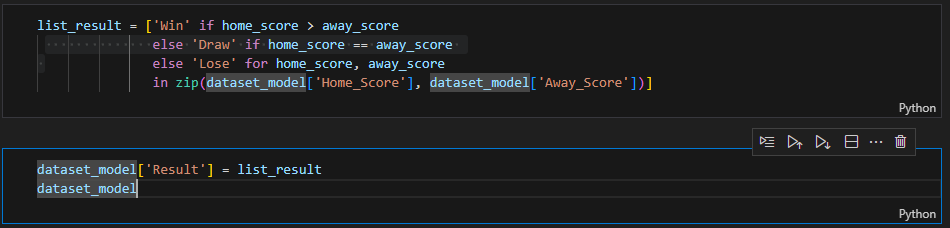
## 6.1. Mô hình dự đoán các trận đấu

### 6.1.1. Phân tích tập dữ liệu

Phần này sẽ bao gồm các tiến trình xử lý dữ liệu về các trận đấu với mục đích tạo thành tập TrainData (tập huấn luyện) cho mô hình dự đoán.

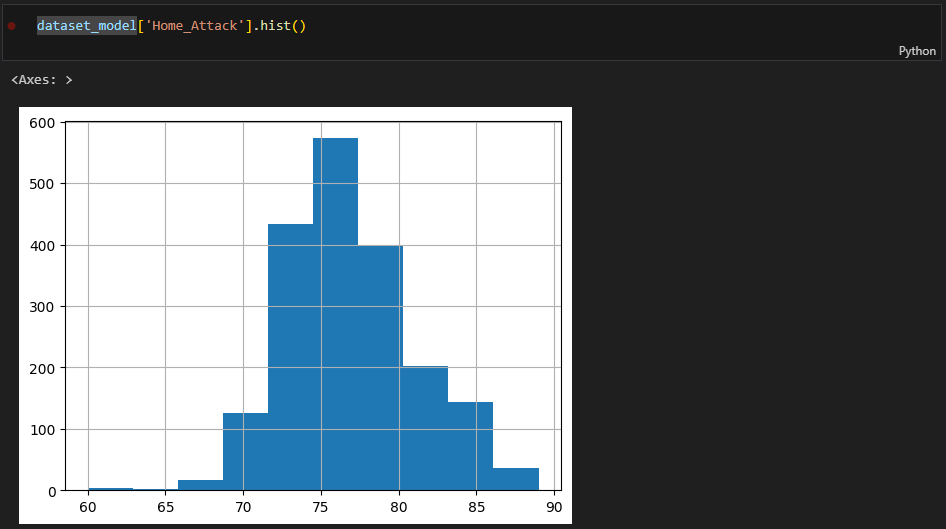


*Hình 6.1: Mã thực thi đọc file csv*



*Hình 6.2: Thêm thuộc tính Result*

Dữ liệu ở phần 5.5.5 trước đã được chuẩn hóa với mục đích chuẩn bị cho xây dựng mô hình dự đoán. Với mục tiêu dự đoán kết quả cho một trận đấu với kết quả là biến phân loại với 3 output Win, Lose, Draw, nhóm thêm một thuộc tính Result có giá trị là một trong 3 output trên.

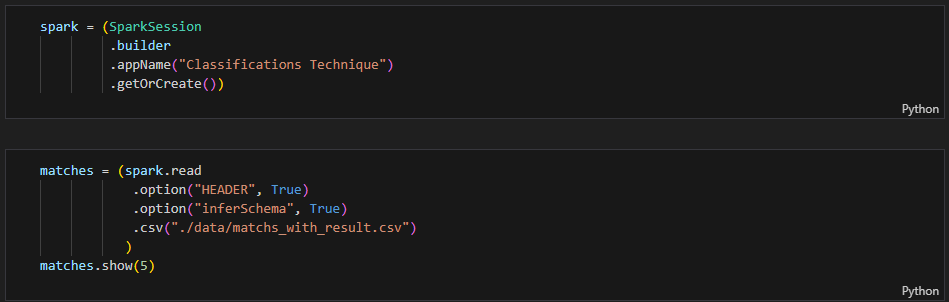


*Hình 6.3: Biểu đồ trực quan hóa chỉ số Attack của đội bóng*

### 6.1.2. Sử dụng các thuật toán phân loại trong Spark Machine Learning:

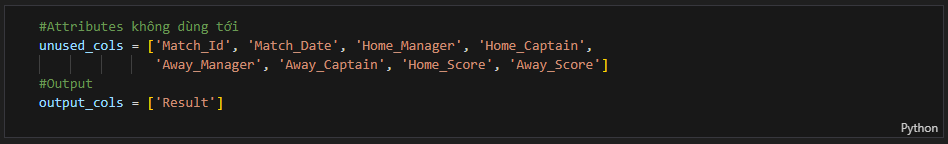


*Hình 6.4: Các thư viện của PySpark dùng cho xử lý dữ liệu và Machine Learning*

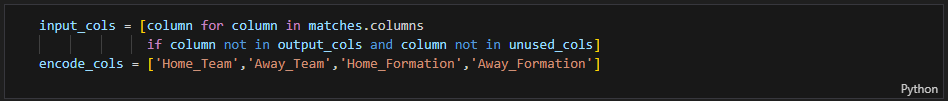


*Hình 6.5: Mã thực thi phiên làm việc trên Spark*

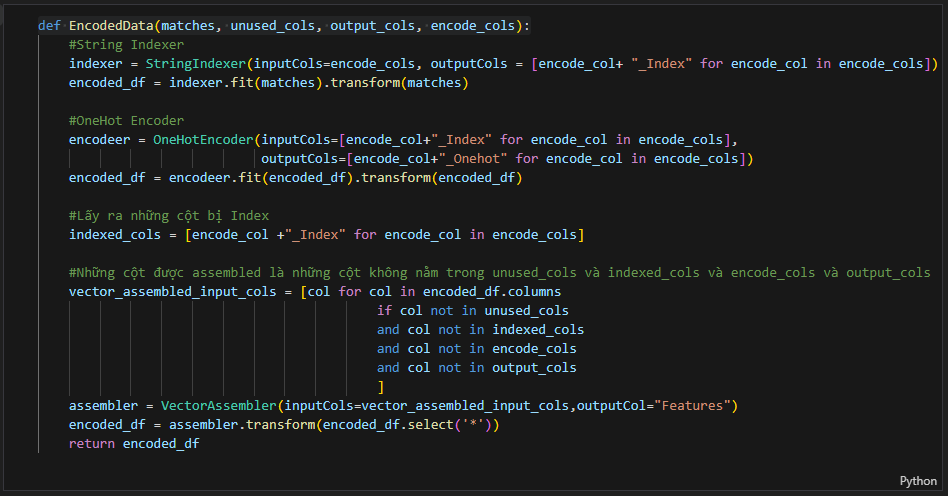
Với Apache Spark đã được đề cập ở phần 3, các tiến trình thực thi để xây dựng mô hình dự đoán được thực hiện dựa trên các thư viện, hàm của PySpark, nó kết nối với Apache Spark, tạo một phiên làm việc trên đó.



*Hình 6.6: Lưu trữ các thuộc tính không cần dùng cho xây dựng mô hình dự đoán*

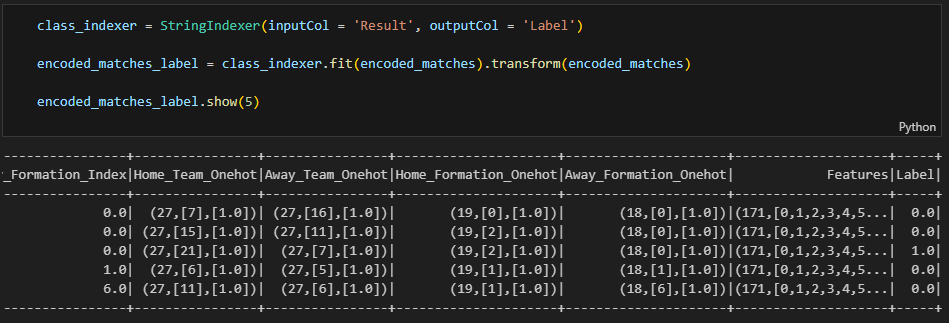


*Hình 6.7: Biến đổi các thuộc tính dạng string sang numeric*



*Hình 6.8: Mã thực thi chuyển đổi các thuộc tính theo phương pháp OneHotEncoding*

Với dữ liệu các trận đấu đã được mô tả ở phần 2, nhóm loại bỏ các thuộc tính không cần thiết như Match\_Id (mã trận đấu), Match\_Date (ngày thi đấu),… Sau đó, với những thuộc tính cần dùng cho model mà có kiểu dữ liệu string như Home\_Team (tên đội sân nhà), Home\_Formation (đội hình), nhóm đưa về kiểu numeric và sử dụng phương pháp OneHotEncoding biến đối các thuộc tính input.



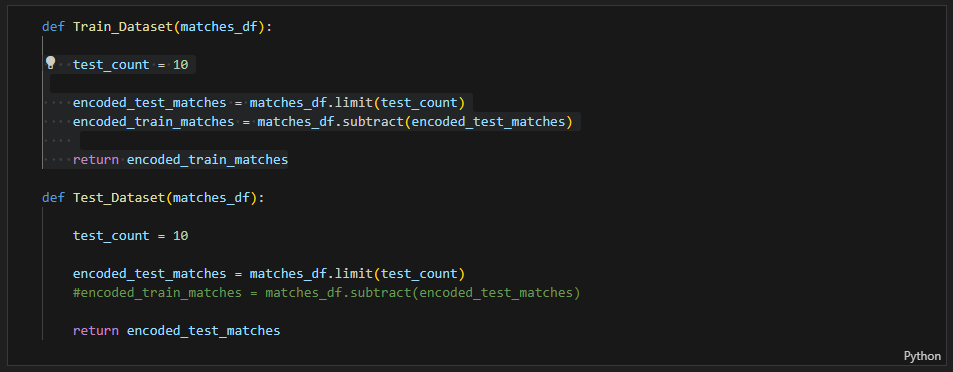
*Hình 6.9: Mã thực thi chuyển đổi biến dự đoán*

Tương tự đối với biến dự đoán output cũng vậy, có 3 loại tương ứng với 3 nhãn Label được gán: 0.0 (Win), 1.0 (Lose), 2.0 (Draw). Kết quả ta được hai trường mới:

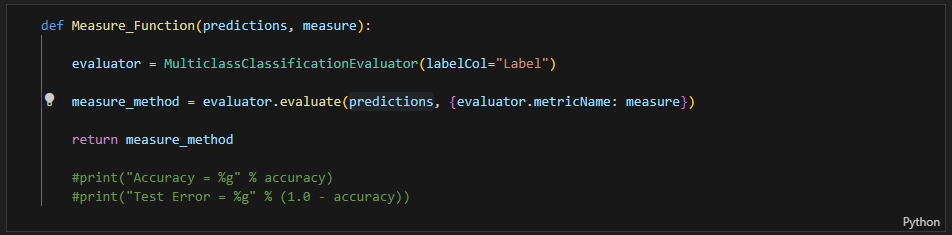
* Features: chứa giá trị của tất cả các input qua quá trình OneHotEncoding
* Label: nhãn kết quả



*Hình 6.10: Hàm loại bỏ 10 trận đấu mới nhất*

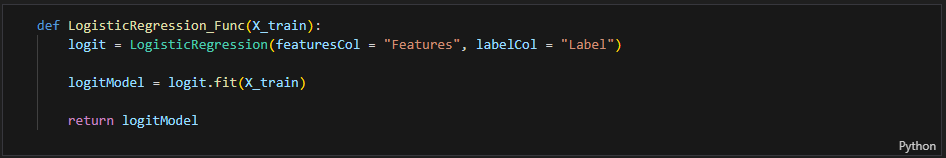


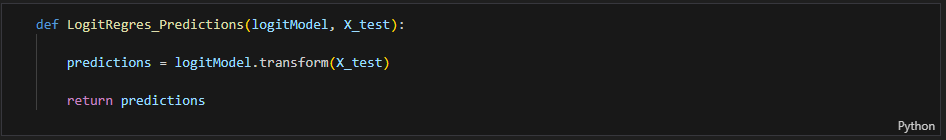
*Hình 6.11: Hàm trả về tập train và test*



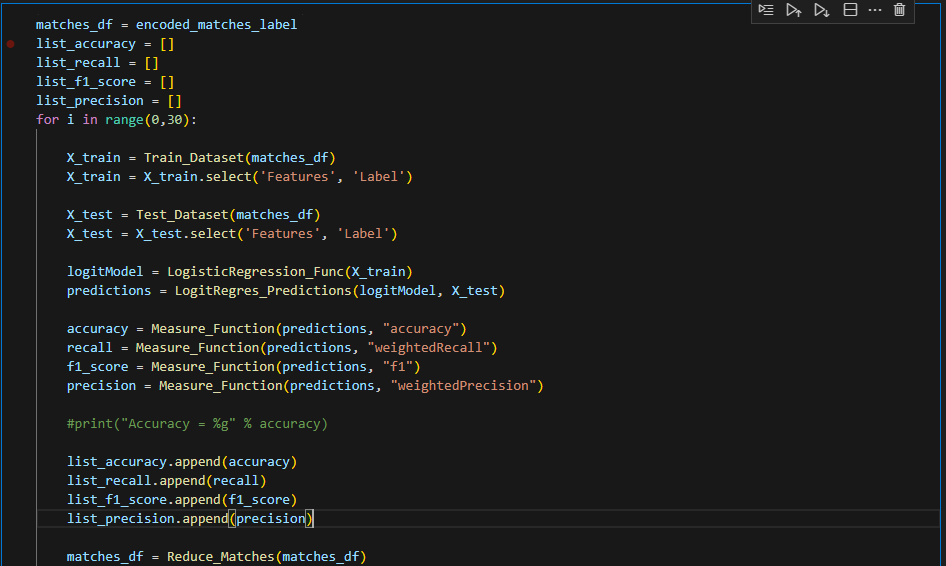
*Hình 6.12: Hàm tính toán các phương pháp đánh giá mô hình*

Vì dữ liệu về các trận đấu có tính chất về thời gian (ngày thi đấu) nên nhóm tiến hành xây dựng tập train cho các trận đấu trong quá khứ và dự đoán các trân đấu trong tương lai. Kết hợp với phương pháp Cross Validation và trên thực tế mỗi tuần diễn ra 10 trận đấu, nhóm lần lượt xây dựng mô hình, dự đoán và dùng các phương pháp đánh giá mỗi 10 trận đấu mới nhất.



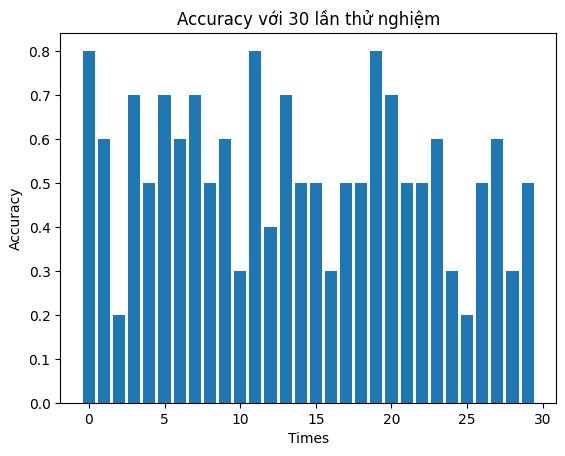


*Hình 6.13: Hàm xây dựng mô hình dự đoán theo thuật toán Logistic Regression*

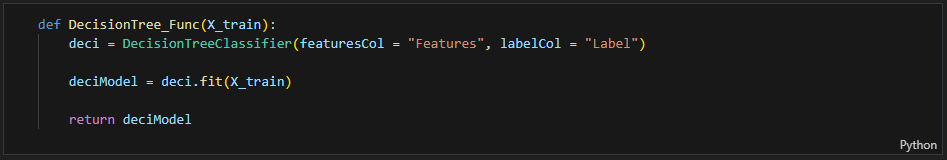


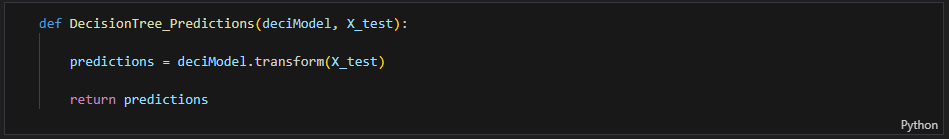
Hình: Mã thực thi xây dựng mô hình theo thuật toán Logistic Regression và lưu lại các độ đo đánh giá

Với các thuật toán đã được trình bày ở phần 2 – Cơ sở lý thuyết, nhóm áp dụng để xây dựng mô hình dự đoán. Đầu tiên là huấn luyện mô hình dựa trên tập trên với hàm LogisticRegression(), hàm transform() tiến hành dự đoán trên tập test, các cấp độ đo đánh giá mô hình Accuracy, Recall, F1\_Score và Precision được lưu lại sau nhiều lần huấn luyện và dự đoán như vậy để dùng cho đánh giá các thuật toán khác.

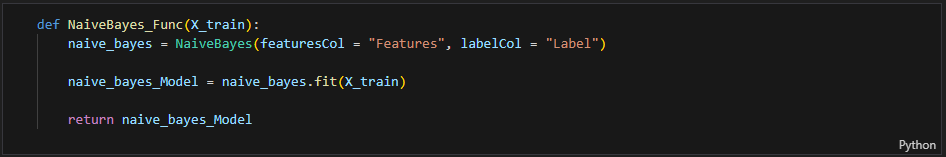


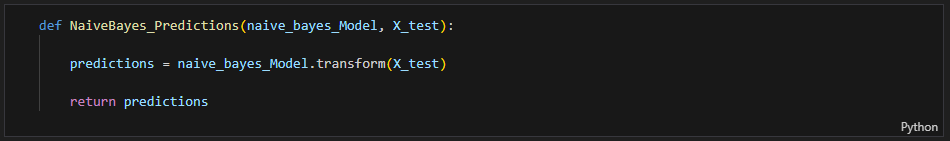
*Hình 6.14: Độ đo Accuracy sau mỗi lần huấn luyện, giảm tập dữ liệu*



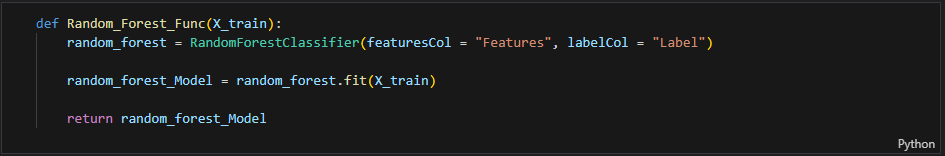


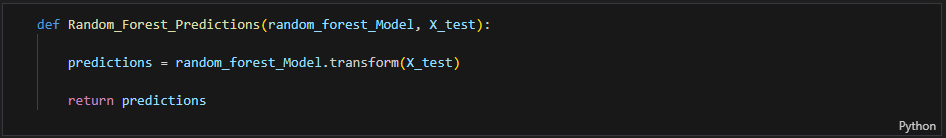
*Hình 6.15: Hàm xây dựng mô hình dự đoán theo thuật toán Decision Tree*



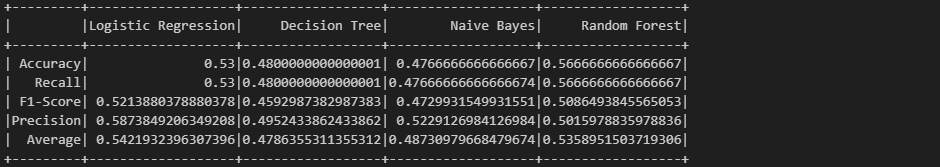


*Hình 6.16: Hàm xây dựng mô hình dự đoán theo thuật toán NaiveBayes*





*Hình 6.17: Hàm xây dựng mô hình dự đoán theo thuật toán Random Forest*

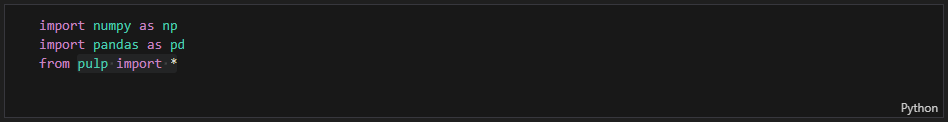


*Hình 6.18: Các độ đo đánh giá mô hình theo các thuật toán*

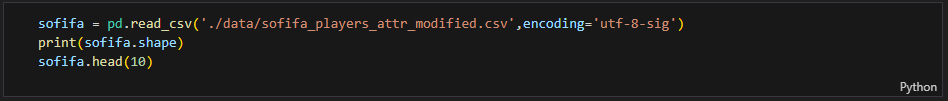
## 6.2. Mô hình tối ưu hóa đội hình

### 6.2.1. Phân tích tập dữ liệu

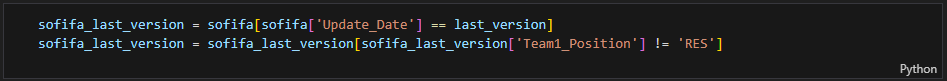
Mô hình tối ưu hóa đội hình của đội bóng sử dụng dữ liệu của các cầu thủ trong Sofifa.



*Hình 6.19: Thư viện pulp sử dụng cho bài toán tuyến tính nguyên hỗn hợp*

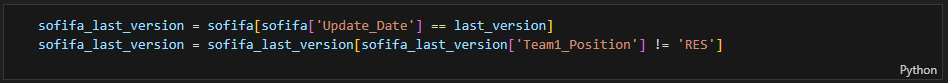


*Hình 6.20: Dữ liệu về các cầu thủ trong Sofifa*



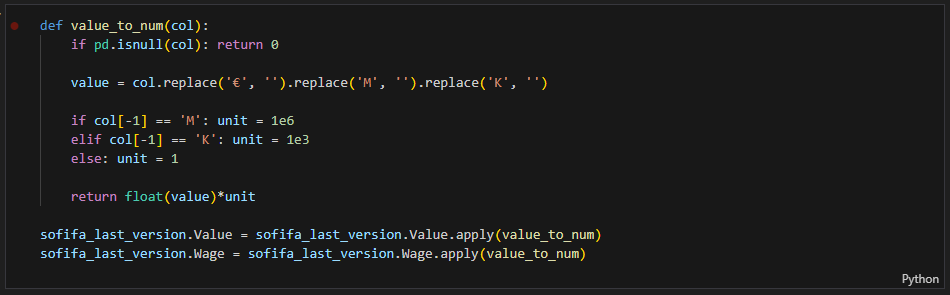
*Hình 6.21: Lọc ra những cầu thủ trong cùng một phiên bản mới nhất*

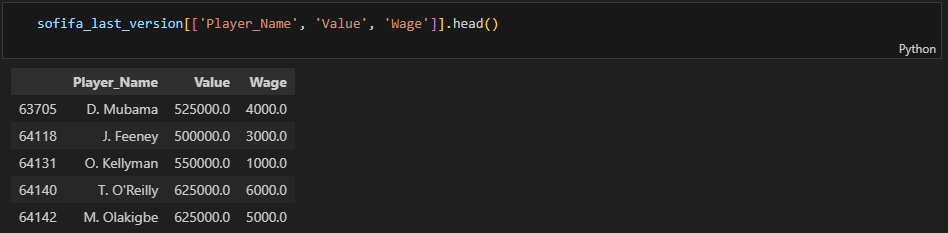
Như đã đề cập thì các thông số cầu thủ trong Sofifa nhóm sẽ tiến hành xử lý, do tính chất về thời gian nên mỗi đội hình tại các thời điểm thi đấu khác nhau sẽ bao gồm các cầu thủ ở phiên bản tại thời điểm đó. Với mục đích đưa ra đội hình tối ưu nhất ở thời điểm hiện tại nên nhóm lọc ra tất cả các cầu thủ mới nhất trong Sofifa.



*Hình 6.22: Xóa những cầu thủ nghỉ thi đấu hoặc chấn thương*

Với những cầu thủ nghỉ thi đấu hoặc nghỉ do chấn thương thì thuộc tính Team\_Position có giá trị là “RES”, ta loại bỏ những cầu thủ này.



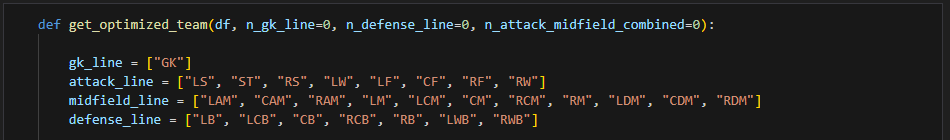


*Hình 6.23: Đổi giá trị Value cho cầu thủ*

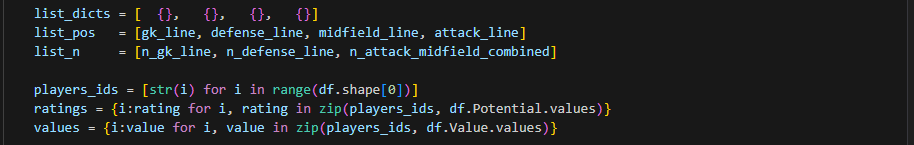
Giá trị Value (giá trị cầu thủ) là một trong các giá trị để đánh giá một cầu thủ, nhóm sẽ đề cập ở phần tiếp.

### 6.2.2. Xây dựng mô hình tối ưu hóa đội hình dựa vào các thống số cầu thủ:

Mỗi đội bóng tại một thời điểm sẽ bao gồm các cầu thủ có trong đội hình, bao gồm cả “RES” (cầu thủ nghỉ thi đấu hoặc do chấn thương) và “SUB” (vị trí dự bị).



*Hình 6.24: Chia các vị trí cầu thủ trên sân thành 4 nhóm*



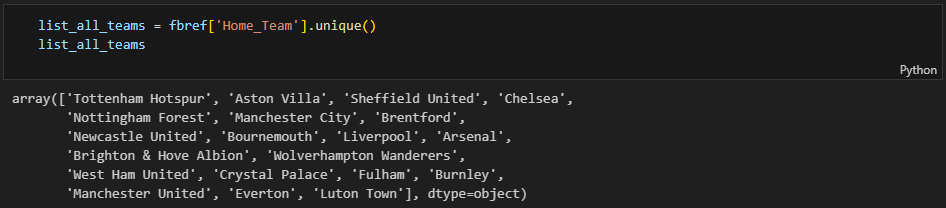
*Hình 6.25: Lấy thông số Potential và Value của cầu thủ*

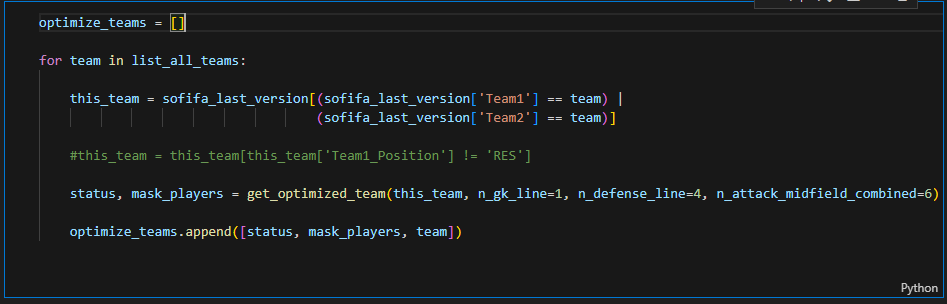


*Hình 6.26: Xây dựng mô hình tuyến tính nguyên hỗn hợp*

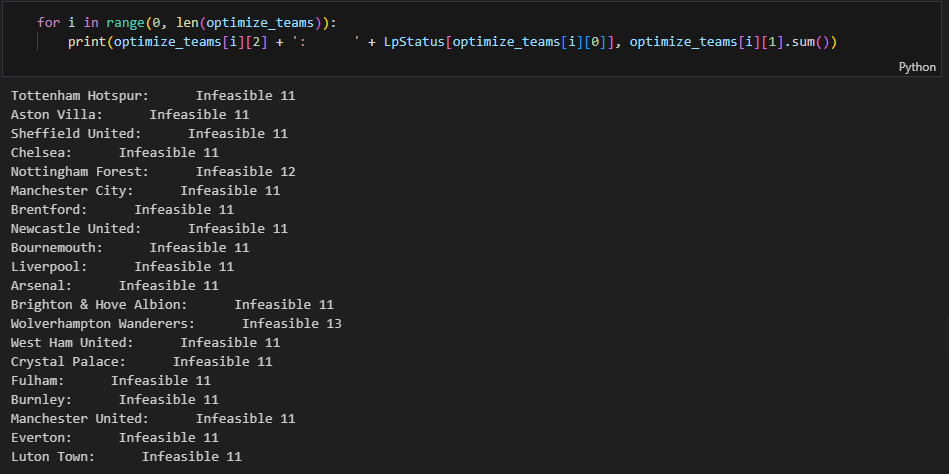


*Hình 6.27: Lấy dữ liệu các trận đấu trong mùa giải mới nhất*





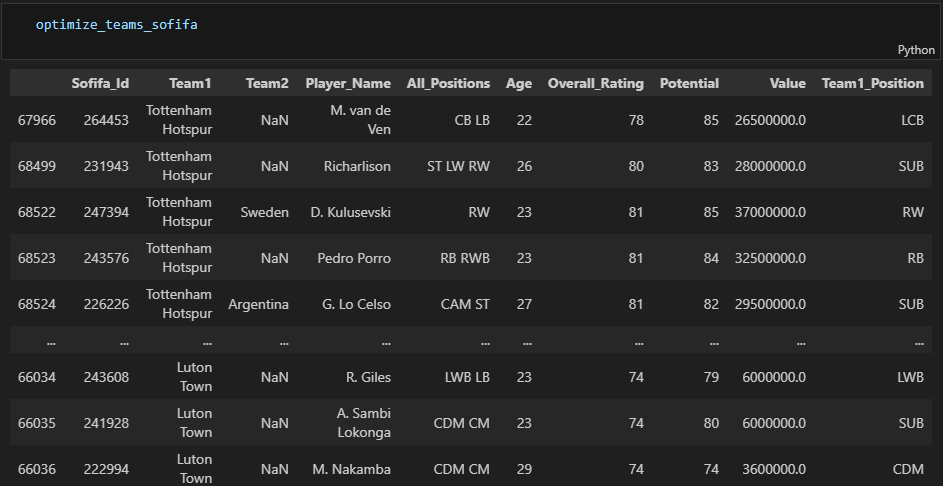
*Hình 6.28: Thực thi tối ưu hóa các cầu thủ cho mỗi team*

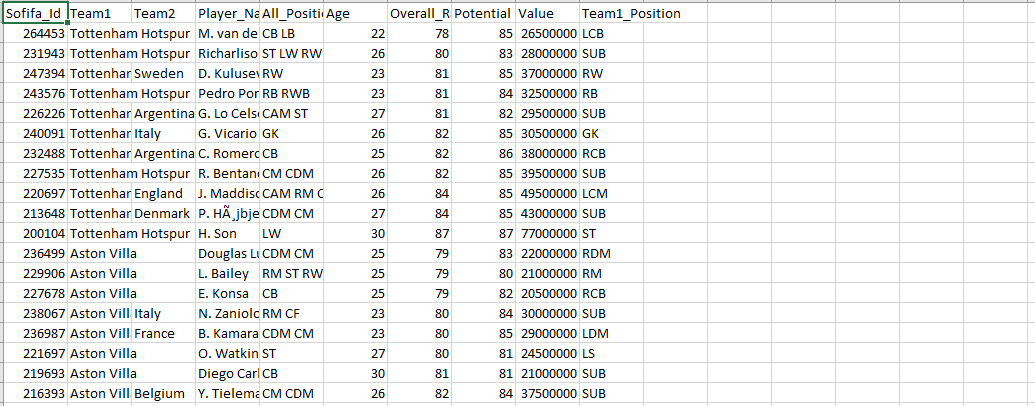


*Hình 6.29: Kết quả các cấu thủ tối ưu cho mỗi đội bóng*

Các đội bóng trong mùa giải mới nhất (2023/2024) lần lượt được cho vào mô hình tối ưu, mô hình chọn ra tối thiểu 11 cầu thủ mạnh nhất. Các giá trị của cầu thủ dùng để đánh giá chất lượng cầu thủ:

* Value: giá trị cầu thủ
* Potential: chỉ số đánh giá tổng thể chất lượng cầu thủ
* All\_Positions: tất cả các vị trí có thể chơi





*Hình 6.30: Kết quả đội hình tối ưu cho mỗi đội bóng*

Kết quả là với mỗi đội bóng, mô hình sẽ cho ra ít nhất 11 cầu thủ tối ưu nhất dựa vào các thông số đánh giá đã đề cập ở trên.

# PHẦN 7: ĐÁNH GIÁ MÔ HÌNH BẰNG THỰC NGHIỆM

## 7.1. Các độ đo đánh giá mô hình

### 7.1.1. Độ chính xác - Accuracy

Độ chính xác là tỉ lệ giữa giá trị dự đoán đúng của mô hình so với thực tế và tổng giá trị trong kiểm thử. [24]

### 7.1.2. Độ đúng đắn - Precision

Độ đúng đắn được định nghĩa là tỉ lệ số điểm dương tính thật trong số những điểm được phân loại là dương tính (Dương tính thật + Dương tính giả). Công thức tính độ đúng đắn: [24]

Độ đúng đắn cao có nghĩa rằng độ chính xác của các điểm tìm được là cao.

### 7.1.3. Recall

Recall được định nghĩa là tỉ lệ số điểm dương tính thật trong số những điểm thực sự là dương tính (Dương tính thật + Âm tính giả). Recall cao đồng nghĩa với việc tỉ lệ dương tính thật cao, tức tỉ lệ bỏ sót các điểm thực sự dương tính là thấp. [24]

### 7.1.4. F1-Score

Điểm F1 là trung bình hài hoà của độ đúng đắn và recall (giả sử rằng hai đại lượng này khác 0). Công thức để tính toán F1-score: [24]

Điểm F1 có giá trị nằm trong khoảng (0;1]. Nếu F1-score càng lớn, có nghĩa rằng bộ phân lớp hoạt động càng tốt. Trong trường hợp cả Độ đúng đắn và Recall đều có giá trị là 1 thì điểm F1 sẽ đạt giá trị lớn nhất là 1. Ngược lại, nếu cả Độ đúng đắn và Recall đều thấp, thì điểm F1 sẽ càng gần tiến về 0. [24]

### 7.1.5. Weighted recall:

Weighted recall là một độ đo được tính từ recall của từng lớp riêng biệt trong bài toán phân lớp, sau đó tính trung bình dựa trên trọng số của mỗi lớp. Trọng số này được tính bằng tỷ lệ tần suất của mỗi lớp trong tập dữ liệu. Những lớp có tần suất xuất hiện cao hơn sẽ có trọng số cao hơn, trong khi các lớp có tần suất xuất hiện thấp hơn sẽ có trọng số thấp. Việc tính toán này đảm bảo rằng "weighted recall" có tác động đến hiệu suất phân loại cho cả các lớp quan trọng và các lớp ít quan trọng trong tập dữ liệu. [24]

## 7.2. Các phương pháp đánh giá mô hình

### 7.2.1. Kiểm chứng chéo – Cross Validation:

Kiểm chứng chéo là một phương pháp kiểm tra độ chính xác của một phương pháp máy học dựa trên một tập dư liệu học cho trước. Thay vì chỉ dùng một phần dữ liệu làm tập dữ liệu học thì kiểm chứng chéo dùng toàn bộ dữ liệu để dạy cho máy. [24]

Ba loại phương pháp kiểm chứng chéo phổ biến được sử dụng rộng rãi:

* Hold-out: phương pháp đơn giản nhất. Dữ liệu được chia một cách ngẫu nhiên thành một tập dữ liệu học và một tập dữ liệu kiểm tra. Dùng tập đầu tiên để dạy máy rồi dùng ngay tập còn lại để kiểm tra.
* K-fold: đây là phương pháp nâng cấp của hold-out. Toàn bộ dữ liệu được chia thành K tập con. Quá trình học của máy có K lần. Trong mỗi lần, một tập con được dùng để kiểm tra và K-1 tập còn lại dùng để dạy.
* Leave-one-out: Phương pháp này tương tự như K-Fold nhưng tối đa hóa số tập con (K = số dữ liệu).

## 7.3. Các trường hợp thực nghiệm

### 7.3.1. Đánh giá, nhận xét và so sánh sau khi sử dụng nhiều mô hình

Nhóm sử dụng 4 mô hình phân loại khác nhau bao gồm: Mô hình hồi quy Logistic, mô hình Cây quyết định, mô hình Naïve Bayes, và cuối cùng là mô hình Rừng ngẫu nhiên nhằm có thể chọn ra mô hình tốt nhất trong 4 mô hình.

### 7.3.2. Đánh giá và nhận xét mô hình khi sử dụng phương pháp kiểm chứng chéo

Ở trong đề tài này, nhóm sử dụng phương pháp kiểm chứng chéo K-fold để đánh giá mô hình. Nhưng do để dự đoán trận đấu với các mốc thời gian khác nhau, nhóm quyết định rằng sẽ chọn tất cả dữ liệu huấn luyện cho mô hình đều phải có mốc thời gian quá khứ so với tất cả dữ liệu kiểm thử. Và để đánh giá mô hình một cách tối ưu, nhóm sẽ huấn luyện và kiểm thử 30 lần khác nhau, mỗi lần sẽ giảm đi 10 trận đấu. Nói theo cách khác, với tập dữ liệu có N trận đấu, lần huấn luyện thứ *i* sẽ có *10\*i* trận đấu dùng để kiểm thử, và *N-(10\*i)* dùng để kiểm thử.

### 7.3.3. Đánh giá độ ảnh hưởng của các siêu tham số sau khi tối ưu mô hình:

Đối với mô hình hồi quy Logistic, nhóm thực hiện tối ưu với các siêu tham số như sau, với mong muốn mô hình có thể tối ưu hơn lúc chưa được tối ưu:

|  |  |
| --- | --- |
| Biến chuẩn hoá | Biến chuẩn hoá mạng đàn hồi |
| 0 | 0 |
| 0.08 | 0.6 |
| 0.02 | 0.6 |
| 0.08 | 0.2 |
| 0.02 | 0.2 |

## 7.4. Kết quả thực nghiệm:

### 7.4.1. Đánh giá, nhận xét và so sánh sau khi sử dụng nhiều mô hình kết hợp sử dụng phương pháp kiểm chứng chéo

Biểu đồ của độ chính xác từ các mô hình:

A graph with blue lines

Description automatically generated

*Hình 7.1: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình hồi quy Logistic.*

A graph of blue bars

Description automatically generated

*Hình 7.2: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình Cây quyết định.*

A graph with blue lines

Description automatically generated

*Hình 7.3: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình Naïve Bayes.*

A graph of blue bars

Description automatically generated

*Hình 7.4: Biểu đồ độ chính xác sau 30 lần chạy và kiểm chứng chéo của mô hình Rừng ngẫu nhiên.*

Sau khi chạy 30 lần kiểm chứng chéo, các độ đo của mô hình sẽ được ghi lại và lấy trung bình, kết quả độ đo của các mô hình khác nhau như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Logistic Regression | Decision Tree | Naïve Bayes | Random Forest |
| Độ chính xác | 0.53 | 0.48 | 0.48 | 0.56 |
| Recall | 0.53 | 0.48 | 0.48 | 0.56 |
| Điểm F1 | 0.52 | 0.46 | 0.47 | 0.50 |
| Độ đúng đắn | 0.59 | 0.50 | 0.52 | 0.48 |

Nhận xét: Qua 4 mô hình trên với 30 lần kiểm chứng chéo tương ứng với mỗi mô hình, ta nhận thấy rằng như sau:

* Độ chính xác và recall của các mô hình là bằng nhau, điều này nằm ngoài dự đoán của nhóm thực hiện nên nhóm đã tìm hiểu nguyên nhân. Độ đo được sử dụng để tính Recall thực chất là Weighted Recall. Để hiểu rõ về vấn đề này, chúng ta cần phải xem lại công thức của Độ chính xác được tính là:

, và của weighted recall được tính là:

trong đó

, từ đó tính trọng số cho Weighted Recall theo công thức:

*= Accuracy*

Do vậy, Weighted recall chính là Accuracy. [25]

* Mô hình Rừng ngẫu nhiên có mọi chỉ số cao hơn mô hình cây quyết định như dự tính ban đầu của nhóm vì mô hình Rừng ngẫu nhiên được cấu thành nên bởi nhiều Cây quyết định.
* Ngoài ra, với mô hình hồi quy Logistic và Rừng ngẫu nhiên có hầu hết độ đo cao hơn với hai mô hình còn lại. Dù cho mô hình Rừng ngẫu nhiên cho ra độ chính xác cao hơn, nhưng điểm F1 lại thấp hơn hồi quy Logistic. Do vậy, mô hình hồi quy Logistic có thể được xem là mô hình tốt nhất trong 4 mô hình này.

### 7.4.3. Đánh giá, nhận xét độ ảnh hưởng của các siêu tham số sau khi tối ưu mô hình:

Sau khi thực hiện truyền các siêu tham số kết hợp với phương pháp kiểm chứng chéo đã nhắc tới ở trên, nhóm thu được kết quả như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Biến chuẩn hoá | Biến chuẩn hoá mạng đàn hồi | Độ chuẩn xác | Recall | Điểm F1 | Độ đúng đắn |
| 0 | 0 | 0.53 | 0.53 | 0.52 | 0.59 |
| 0.08 | 0.6 | 0.54 | 0.54 | 0.48 | 0.48 |
| 0.02 | 0.6 | 0.54 | 0.54 | 0.48 | 0.48 |
| 0.08 | 0.2 | 0.55 | 0.55 | 0.49 | 0.49 |
| 0.02 | 0.2 | 0.55 | 0.55 | 0.49 | 0.48 |

Nhận xét: Qua việc tối ưu mô hình bằng hai siêu tham số là biến chuẩn hoá và biến chuẩn hoá mạng đàn hồi thì nhóm có nhận xét như sau:

* Các độ đo có sự thay đổi nhưng có sự thay đổi này diễn ra không đáng kể.
* Riêng về độ đo chuẩn xác và recall thì tăng lên, còn điểm F1 và độ đúng đắn.
* Do nhóm xác định rằng điểm F1 là độ đo tiên quyết để chọn lựa ra mô hình tối ưu nhất, mà trong quá trình tối ưu lại chưa có mô hình nào vượt qua giá trị 0.52 (giá trị điểm F1 trước khi tối ưu mô hình) nên nhóm nhận xét rằng việc thực hiện tối ưu mô hình chưa được thành công.

# PHẦN 8: KẾT LUẬN

## 8.1. Kết quả

Nhóm xin đưa ra nhận xét chung về đề tài tiểu luận “Xây dựng nền tảng thống kê và dự đoán các trận đấu bóng đá”. Nhóm đã xây dựng được các tính năng cơ bản của một hệ thống nền tảng giúp người dùng.

### 8.1.1. Về phần thu thập dữ liệu

Về phần thu thập dữ liệu, nhóm nhận định rằng phần mềm thu thập dữ liệu chạy ổn định, chưa gặp lỗi. Đối với Fbref và Sofifa, lượng dữ liệu thu thập từ thông số của cầu thủ và các trận đấu đã diễn ra là khá lớn, nếu như có thể thu thập tổng cộng 5 giải lớn nhất trên thế giới thì khối lượng thông tin có thể cải thiện.

### 8.1.2. Về phần mô hình dự đoán

Mô hình dự đoán kết quả trận đấu với dữ liệu huấn luyện và dữ liệu kiểm thử đa dạng, được chọn lọc, chuẩn bị kĩ càng. Độ chính xác của mô hình dự đoán nằm ở mức chấp nhận được, cần phải cải thiện thêm nếu muốn hoạt động hiệu quả và ứng dụng thực tế.

Việc tối ưu hoá mô hình được nhóm đánh giá là không thành công, mặc dù có cải thiện về độ đo Độ chính xác và Recall, nhưng nhóm đã xác định lấy độ đo điểm F1 làm thước đo chính nhưng sau những lần điều chỉnh, điểm F1 chưa được cải thiện, thậm chí còn giảm đi so với trước khi tối ưu hoá.

## 8.2. Hạn chế, khó khăn

### 8.1.1. Về phần thu thập dữ liệu

Thời gian chạy của phần thu thập dữ liệu là khá lâu, lên tới 3 tới 4 ngày trên một máy, nhóm chưa thể thu thập dữ liệu bằng cách xoay chuyển IP trong mỗi lần yêu cầu trang web mà chỉ dừng lại thay đổi header của gói tin. Do đó, với trang web Sofifa và Fbref cần phải để thời gian hoãn mỗi request để tránh bị liệt kê vào danh sách đen của các trang web đó.

Đối với các trang web có JavaScript, nhóm chưa hoàn toàn ứng dụng được hết thế mạnh của Selenium, đó là sử dụng Selenium Server cho phép sử dụng máy chủ để truy cập trang web để lấy dữ liệu mà không cần phải chạy trình duyệt ở máy khách.

Nhóm thực hiện ứng dụng Kafka còn sơ sài, chưa phát huy được hết thế mạnh của một hàng đợi tin là đăng kí vào topic, hàng chờ, …

### 8.1.2. Về phần mô hình dự đoán

Mô hình dự đoán với tỉ lệ chính xác không được cao do bóng đá là một lĩnh vực khó, có nhiều tình huống bất ngờ khiến cả cục diện trận đấu thay đổi. Ngoài ra, do thời gian hạn hẹp nên số lượng áp dụng mô hình dự đoán còn khá ít và cũng do đó mà việc tối ưu mô hình chưa được đặt nhiều công sức.

## 8.3. Hướng phát triển

### 8.3.1. Về tương lai gần

Với các thành phần trong hệ thống thông tin, nhóm có thể triển khai các ứng dụng trên điện toán đám mây nhằm tăng cường khả năng xử lý, phân tích.

Các tập dữ liệu về các trận đấu và cầu thủ trong tương lai gần có thể giúp nhóm đánh giá và cải tiến hơn cho mô hình dự đoán.

### 8.3.2. Về tương lai xa

Tập dữ liệu về bóng đá là rất lớn, rất nhiều giải đấu đi với đó là vô số cầu thủ, do đó có thể xây dựng một kho dữ liệu về bóng đá bao gồm các bước trong quá trình ETL.

Các thành phần được triển khai trên các nền tảng điện toán đám mây nên ta có thể xây dựng một pipeline nơi mà các thành phần hoạt động độc lập với nhau nhưng phối hợp với nhau, các hoạt động từ thu thập dữ liệu về các trận đấu mới cho đến xử lý, phân tích và cho ra kết quả có thể được thực hiện một cách tự động.

Khi có một dữ liệu lớn về bóng đá thì ta có thể xây dựng thêm nhiều mô hình khác ngoài như các mô hình trên tập dữ liệu cầu thủ, phân tích chuyên sâu.

# **TÀI LIỆU THAM KHẢO, PHỤ LỤC**

[1] Học viện Agilem, *Khái niệm và những bí mật chưa ai kể về Supervised Learning* - <https://hocvienagile.com/khai-niem-va-nhung-bi-mat-ve-supervised-learning/>

[2] Amazon, *Hồi quy logistic là gì?* - <https://aws.amazon.com/vi/what-is/logistic-regression/>

[3] Trí tuệ nhân tạo, *Cây quyết định*, 06/06/2019 - <https://trituenhantao.io/kien-thuc/decision-tree/>

[4] Tran Duc Tan, *Mô hình phân lớp Naïve Bayes*, 22/06//2019 - <https://viblo.asia/p/mo-hinh-phan-lop-naive-bayes-vyDZO0A7lwj>

[5] ThongKe.club, *Ứng Dụng Của Thuật Toán Rừng Ngẫu Nhiên – Random Forests*, 06/09/2022 - <https://thongke.club/ung-dung-cua-thuat-toan-rung-ngau-nhien-random-forests/>

[6] MathWorks, *Mixed-Integer Linear Programming (MILP) Algorithms*, 2023 - https://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html

[7] Mayank Banoula, The Best Guide to Regularization in Machine Learning, 07/11/2023 - https://www.simplilearn.com/tutorials/machine-learning-tutorial/regularization-in-machine-learning

[8] Amazon Web Services, Python là gì?, 2023 - [aws.amazon.com/vi/what-is/python/](https://aws.amazon.com/vi/what-is/python/)

[9] Lê Ngọc Hồng Quân, *Visual Studio Code là gì?*, 31/05/2022 - [fptshop.com.vn/tin-tuc/danh-gia/visual-studio-code-la-gi-cac-tinh-nang-noi-bat-cua-visual-studio-code-146213](https://fptshop.com.vn/tin-tuc/danh-gia/visual-studio-code-la-gi-cac-tinh-nang-noi-bat-cua-visual-studio-code-146213)

[10] HCM23\_FRF\_EMB\_05\_MinhVT, *Jupyter Notebook - Công Cụ Python Cơ Bản Cho Beginner*, 03/08/2020 - [codelearn.io/sharing/jupyter-notebook-tutorial](https://codelearn.io/sharing/jupyter-notebook-tutorial)

[11] Trần Đức Trung, *Tìm hiểu chung về Web Scraping và các vấn đề cần quan tâm*, 13/11/2020 - [viblo.asia/p/tim-hieu-chung-ve-web-scraping-va-cac-van-de-can-quan-tam-djeZ1yXJZWz](https://viblo.asia/p/tim-hieu-chung-ve-web-scraping-va-cac-van-de-can-quan-tam-djeZ1yXJZWz)

[12] [github.com/scrapy/scrapy](https://github.com/scrapy/scrapy)

[13] VANANHTO, *Làm quen: Tóm tắt cơ bản về Selenium*, 19/07/2017 - <https://vananhtooo.wordpress.com/2017/07/19/lam-quen-tom-tat-co-ban-ve-selenium/>

[14] Wikipedia , *Apache Kafka*, 18/04/2023- [vi.wikipedia.org/wiki/Apache\_Kafka](https://vi.wikipedia.org/wiki/Apache_Kafka)

[15] Vinh Phạm, *Kafka là gì?* *Tìm hiểu khái niệm cơ bản về Apache Kafka*, 10/08/2022 - [bizflycloud.vn/tin-tuc/kafka-la-gi-gioi-thieu-tong-quan-ve-kafka-20220810160652162.htm](https://bizflycloud.vn/tin-tuc/kafka-la-gi-gioi-thieu-tong-quan-ve-kafka-20220810160652162.htm)

[16] Wikipedia, *Apache Spark*, 8/11/2023 - [en.wikipedia.org/wiki/Apache\_Spark](https://en.wikipedia.org/wiki/Apache_Spark)

[17] Wikipedia, *Apache Hadoop*, 30/11/2023 -[en.wikipedia.org/wiki/Apache\_Hadoop](https://en.wikipedia.org/wiki/Apache_Hadoop)

[18] Dhruba Borthakur, *HDFS Architecture Guide*, 18/05/2022 - [hadoop.apache.org/docs/r1.2.1/hdfs\_design.html](https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html)

[19] FPT Cloud, *MySQL là gì? Cơ chế hoạt động và cách thức cài đặt MySQL*, 16/03/2022 - [https://fptcloud.com/mysql-la-gi/](%20https:/fptcloud.com/mysql-la-gi/)

[20] Kiên Nguyễn, *Django là gì? Tất cả những điều cần biết về framework này*, 2023 - <https://topdev.vn/blog/django-la-gi-tat-ca-nhung-dieu-can-biet-ve-framework-nay/>

[21] Son Che Dinh, *[AngularJS toàn tập] Phần 1: AngularJS là gì?*,26/07/2016 - <https://viblo.asia/p/angularjs-toan-tap-phan-1-angularjs-la-gi-MgNvWYZEeYx>

[22] Kiên Nguyễn, *AngularJS Là Gì? Khác Biệt Nào Giữa Angular Và Frontend Framework Khác* - <https://topdev.vn/blog/angularjs-la-gi/>

[23] Fbref, *All About FBref.com*, fbref.com/en/about/

[24] Vũ Hữu Tiệp, *Bài 33: Các phương pháp đánh giá một hệ thống phân lớp*, 3/1/2018 - <https://machinelearningcoban.com/2017/08/31/evaluation/>

[25] Evidently AI Team, *Accuracy vs. precision vs. recall in machine learning: what's the difference?* - https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall