Phân Tích Luồng Chạy Mã Nguồn Thu Thập Dữ Liệu Thống Kê Bóng Đá trong bài 1 (ETV)

Người làm: Nguyễn Hải Nam

Ngày: 3 tháng 5 năm 2025

Tài liệu này mô tả luồng chạy của mã nguồn dùng để thu thập và xử lý dữ liệu thống kê chi tiếtcủa cầu thủ bóng đá từ trang FBref.com, bao gồm việc tự động truy cập, trích xuất, làm sạch và tổng hợp các bảng thống kê thành một tập dữ liệu hoàn chỉnh phục vụ cho các phân tích tiếp theo.

Mục lục

1. Giới thiệu

Báo cáo này phân tích luồng thực thi của một đoạn mã Python được thiết kế để thu thập dữ liệu thống kê cầu thủ bóng đá từ trang web FBref cho mùa giải Premier League 2024-2025. Mã nguồn sử dụng các thư viện như Selenium, BeautifulSoup và Pandas để trích xuất, xử lý và gộp dữ liệu từ nhiều trang web thành một tệp CSV duy nhất. Phân tích này trình bày chi tiết cấu trúc mã, các thành phần chính và các bước xử lý dữ liệu.

2. Cấu trúc mã và các thư viện phụ thuộc

Mã nguồn bắt đầu bằng việc nhập các thư viện Python cần thiết, mỗi thư viện đảm nhiệm một vai trò cụ thể trong quy trình thu thập và xử lý dữ liệu:

- `time`: Quản lý thời gian chờ để đảm bảo trang web tải hoàn toàn.
- `pandas`: Xử lý và lưu trữ dữ liệu dưới dạng bảng (DataFrame).
- `BeautifulSoup, Comment`: Phân tích nội dung HTML để trích xuất dữ liệu.
- `selenium.webdriver`: Tự động hóa tương tác trình duyệt để thu thập nội dung web động.
- `selenium.webdriver.chrome.options, Service, webdriver_manager.chrome`: Cấu hình và quản lý ChromeDriver cho Selenium.
- `io.StringIO`: Xử lý chuỗi HTML như một tệp để Pandas đọc.
- `os`: Thao tác với hệ thống tệp, như tạo thư mục và đường dẫn.

3. Phân tích luồng thực thi

Quá trình thực thi mã nguồn có thể được chia thành một số giai đoạn chính, mỗi giai đoạn đóng góp vào mục tiêu tổng thể là thu thập và xử lý thống kê bóng đá.

3.1. Khởi tạo và cấu hình

Mã bắt đầu bằng việc xác định thư mục gốc để lưu trữ tệp bằng các hàm `os.path`, đảm bảo tương thích đa nền tảng. Hai hàm tiện ích được định nghĩa:

- `convert_age_to_decimal(age_str)`: Chuyển đổi tuổi từ định dạng chuỗi (ví dụ: “22-123”) thành số thập phân (ví dụ: 22.34), giả sử một năm có 365 ngày.
- `extract_country_code(nation_str)`: Trích xuất mã quốc gia từ chuỗi quốc tịch (ví dụ: “eng-Eng” thành “Eng”).

Selenium được cấu hình để chạy Chrome ở chế độ ẩn (không hiển thị cửa sổ trình duyệt) với các tùy chọn như `--disable-gpu` và `--no-sandbox` để tối ưu hóa hiệu suất và tương thích. ChromeDriver được tự động cài đặt và khởi tạo bằng `webdriver_manager`.

3.2. Xác định nguồn dữ liệu

Mã xác định danh sách các URL và ID bảng tương ứng để thu thập dữ liệu từ các danh mục cụ thể (ví dụ: thống kê tiêu chuẩn, thủ môn, sút bóng) trên trang FBref. Các URL nhắm đến các trang như:

- `https://fbref.com/en/comps/9/2024-2025/stats/...` cho thống kê tiêu chuẩn.
- `https://fbref.com/en/comps/9/2024-2025/keepers/...` cho thống kê thủ môn.

Các ID bảng (ví dụ: `stats_standard`, `stats_keeper`) xác định các bảng HTML chứa dữ liệu mong muốn. Danh sách `required_columns` chỉ định các chỉ số cần trích xuất, như tên cầu thủ, số bàn thắng, kiến tạo và kỳ vọng bàn thắng (xG). Một `column_rename_dict` ánh xạ tên cột thô từ trang web sang tên chuẩn hóa để đảm bảo nhất quán.

3.3. Quy trình thu thập dữ liệu web

Mã lặp qua các URL và ID bảng song song bằng vòng lặp `for`. Với mỗi cặp URL-bảng:

1. Selenium (`driver.get(url)`) tải trang web, với thời gian chờ 3 giây (`time.sleep(3)`) để đảm bảo nội dung JavaScript được hiển Bảng HTML được hiển thị.
2. BeautifulSoup phân tích HTML của trang (`driver.page_source`) để tìm bảng với ID được chỉ định.
3. Nếu tìm thấy bảng, `read_html` của Pandas chuyển bảng HTML thành DataFrame, sử dụng hàng đầu tiên làm tiêu đề.
4. Các cột được đổi tên bằng `column_rename_dict`, và các cột trùng lặp được loại bỏ để tránh dư thừa.
5. Một cột `Player_Team` được tạo bằng cách nối `Player` và `Team` để làm định danh duy nhất cho việc gộp dữ liệu.
6. Cột `Age`, nếu có, được chuyển đổi sang định dạng thập phân bằng `convert_age_to_decimal`.
7. DataFrame được lọc để chỉ bao gồm các `required_columns` và lưu vào từ điển `all_tables`.

3.4. Gộp dữ liệu

Mã gộp tất cả các DataFrame trong `all_tables` thành một `merged_df` duy nhất:

1. DataFrame đầu tiên khởi tạo `merged_df`.

2. Các DataFrame tiếp theo được gộp bằng `pd.merge` với phép nối ngoài (outer join) dựa trên cột `Player_Team`.
3. Các hàng trùng lặp trong mỗi DataFrame được loại bỏ trước khi gộp để tránh dư thừa.
4. Các cột xung đột (ví dụ: `Player` từ các bảng khác nhau) được xử lý bằng cách gộp giá trị và xóa cột thừa.
5. Cột `Player_Team` được xóa khỏi DataFrame cuối cùng.

3.5. Làm sạch và định dạng dữ liệu

DataFrame được gộp trải qua các bước làm sạch:

- Các cột được sắp xếp lại theo `required_columns`, loại bỏ các cột không có mặt.
- Các kiểu dữ liệu được áp dụng:
  - Các cột số nguyên (ví dụ: `Minutes`, `Gls`) được chuyển thành `Int64` (kiểu số nguyên có thể chứa giá trị null).
  - Các cột số thực (ví dụ: `xG`, `Save%`) được chuyển thành số thực và làm tròn đến hai chữ số thập phân.
  - Các cột chuỗi (ví dụ: `Player`, `Nation`) được giữ nguyên dưới dạng chuỗi.
- Các cầu thủ có thời gian thi đấu dưới 90 phút bị loại bỏ.
- Cột `Nation` được làm sạch bằng `extract_country_code` để chuẩn hóa mã quốc gia.

3.6. Tạo tệp đầu ra

Mã tạo một thư mục `csv` trong thư mục gốc nếu chưa tồn tại. DataFrame `merged_df` đã làm sạch được lưu dưới dạng `result.csv` với:

- Mã hóa UTF-8 với BOM (`utf-8-sig`) để tương thích với Excel.
- Không bao gồm cột chỉ số (`index=False`).
- Các giá trị thiếu được biểu diễn bằng "N/A" (`na_rep="N/A"`).

Mã in thông báo xác nhận với đường dẫn tệp và kích thước DataFrame, sau đó đóng Selenium WebDriver.

4. Kết luận

Mã Python thu thập và xử lý thống kê bóng đá từ FBref một cách hiệu quả, chuyển đổi các bảng HTML thô thành một tệp CSV có cấu trúc. Luồng thực thi được tổ chức tốt, tiến triển từ cấu

hình và trích xuất dữ liệu đến gộp, làm sạch và tạo đầu ra. Thiết kế mã ưu tiên tính mạnh mẽ và khả năng thích nghi, phù hợp cho các tác vụ thu thập dữ liệu tự động.