Phân Tích Luồng Chạy Mã Nguồn Phân Tích Dữ Liệu Thống Kê Bóng Đá trong bài 2

Người làm: Nguyễn Hải Nam

Ngày: 3 tháng 5 năm 2025

Tài liệu này mô tả luồng chạy của mã nguồn Python dùng để phân tích dữ liệu thống kê cầu thủ bóng đá từ tệp result.csv, được tạo từ dữ liệu của trang FBref.com cho mùa giải Premier League 2024-2025. Mã thực hiện các phân tích thống kê, tạo biểu đồ và xác định đội bóng xuất sắc nhất, lưu kết quả vào các tệp CSV, TXT và PNG.

Mục lục

1. Giới thiệu

Báo cáo này phân tích luồng thực thi của mã Python được thiết kế để xử lý và phân tích dữ liệu thống kê bóng đá từ tệp `result.csv`. Mã sử dụng các thư viện `pandas`, `os`, và `matplotlib` để thực hiện các tác vụ như xếp hạng cầu thủ, tính toán thống kê đội bóng, tạo biểu đồ histogram, và xác định đội bóng có hiệu suất tốt 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:

- `pandas`: Xử lý và phân tích dữ liệu dưới dạng bảng (DataFrame).
- `os`: Quản lý đường dẫn tệp và tạo thư mục.
- `matplotlib.pyplot`: Tạo biểu đồ histogram để trực quan hóa dữ liệu.

3. Phân tích luồng thực thi

Mã được tổ chức thành một hàm chính `main()` gọi các hàm riêng biệt cho từng chức năng. Luồng thực thi có thể được chia thành các giai đoạn chính sau.

3.1. Khởi tạo và cấu hình

Mã bắt đầu bằng việc xác định thư mục gốc (`base_dir`) bằng `os.path`, đảm bảo tương thích đa nền tảng. Các thư mục `csv`, `txt`, và `histograms` (với các thư mục con `all_players` và `teams`) được tạo nếu chưa tồn tại. Tệp `result.csv` được đọc vào DataFrame `df` với các giá trị "N/A" được xử lý thành `NaN`. Một bản sao `df_calc` được tạo để tính toán mà không ảnh hưởng đến dữ liệu gốc.

Các cột được phân loại:

- `exclude_columns`: Các cột không tính toán (`Player`, `Nation`, `Team`, `Position`).
- `numeric_columns`: Các cột số còn lại, được ép kiểu thành số và điền `NaN` bằng 0.

3.2. Xếp hạng cầu thủ

Hàm `generate_top_3_rankings()` tạo xếp hạng cho mỗi cột số:

1. Với mỗi cột, lấy 3 cầu thủ có giá trị cao nhất và thấp nhất, kèm theo `Player` và `Team`.
2. Đổi tên cột số thành `Value` và thêm cột `Rank` ("1st", "2nd", "3rd").

3. Lưu kết quả vào từ điển `rankings` với hai khóa: `Highest` và `Lowest`.
4. Ghi kết quả vào `top_3.txt` với định dạng dễ đọc, phân tách bằng dấu gạch ngang.

3.3. Tính toán thống kê đội bóng

Hàm `compute_team_statistics()` tính toán median, mean và độ lệch chuẩn:

1. Cho toàn giải: Tính các chỉ số cho tất cả cầu thủ, lưu vào từ điển `all_stats`.
2. Cho từng đội: Lọc dữ liệu theo `Team`, tính các chỉ số và lưu vào `team_stats`.
3. Tạo DataFrame từ danh sách các từ điển, làm tròn giá trị đến 2 chữ số thập phân.
4. Lưu vào `results2.csv` với mã hóa UTF-8 BOM cho tương thích Excel.

3.4. Tạo biểu đồ histogram

Hàm `generate_histograms()` tạo biểu đồ cho các thống kê được chọn (`Gls per 90`, `xG per 90`, `SCA90`, `GA90`, `TklW`, `Blocks`):

1. Lọc cầu thủ chơi ít nhất 300 phút (nếu có cột `Min`).
2. Cho toàn giải: Vẽ histogram với 20 bin, lưu vào `histograms/all_players`.
3. Cho từng đội: Vẽ histogram với 10 bin, sử dụng màu xanh lá cho thống kê phòng ngự, lưu vào `histograms/teams`.
4. Các biểu đồ có tiêu đề, nhãn trục và lưới, được lưu dưới dạng PNG.

3.5. Xác định đội dẫn đầu từng thống kê

Hàm `compute_highest_team_stats()` tìm đội có giá trị trung bình cao nhất cho mỗi thống kê:

1. Nhóm dữ liệu theo `Team`, tính trung bình các cột số.
2. Với mỗi cột, tìm đội có giá trị trung bình lớn nhất, lưu thông tin vào danh sách.
3. Tạo DataFrame và lưu vào `highest_team_stats.csv`.

3.6. Xác định đội xuất sắc nhất

Hàm `identify_best_team()` xác định đội dẫn đầu nhiều thống kê tích cực nhất:

1. Đọc `highest_team_stats.csv`.
2. Loại bỏ các thống kê tiêu cực (ví dụ: `GA90`, `CrdY`).
3. Đếm số lần mỗi đội xuất hiện trong các thống kê tích cực.

4. Xác định đội có số lần dẫn đầu cao nhất và ghi kết quả vào `The best-performing team.txt`.

3.7. Thực thi chính

Hàm `main()` gọi các hàm trên theo thứ tự, đảm bảo `compute_highest_team_stats()` chạy trước `identify_best_team()` do phụ thuộc dữ liệu. Mã in thông báo xác nhận sau mỗi tác vụ.

4. Kết luận

Mã Python phân tích dữ liệu bóng đá từ `result.csv` một cách hiệu quả, tạo ra các xếp hạng, thống kê, biểu đồ và đánh giá đội bóng xuất sắc. Luồng thực thi được tổ chức thành các hàm độc lập, đảm bảo tính mô-đun và dễ bảo trì. Kết quả được lưu dưới dạng CSV, TXT và PNG, phù hợp cho phân tích tiếp theo và trực quan hóa dữ liệu.