BÁO CÁO BÀI TẬP LỚN PYTHON- THẦY KIM NGỌC BÁCH

HỌ VÀ TÊN: BÙI CÔNG HẬU

MSV: B22DCCN283

NHÓM: 11

CÂU 1: Dữ Liệu Thống Kê Giải Ngoại Hạng Anh 2023-2024

Mục Tiêu:

Mã Python này được xây dựng nhằm thu thập và xử lý dữ liệu thống kê cầu thủ từ trang web FBRef cho giải Ngoại Hạng Anh mùa giải 2023-2024. Dữ liệu thu thập được sẽ được lưu trữ dưới dạng file CSV, phục vụ cho các phân tích và báo cáo tiếp theo.

Thành Phần Chính:

1. Thư Viện Sử Dụng

+ pandas: Một thư viện mạnh mẽ cho việc xử lý và phân tích dữ liệu, cho phép làm việc với các cấu trúc dữ liệu như DataFrame.

+ BeautifulSoup: Thư viện dùng để phân tích cú pháp HTML và XML, giúp dễ dàng trích xuất dữ liệu từ các tài liệu web.

+ requests: Thư viện dùng để gửi các yêu cầu HTTP, hỗ trợ lấy dữ liệu từ internet một cách đơn giản.

2. Hàm crawler(url, id_div, cnt)

+ Chức năng: Hàm này thực hiện việc thu thập và lưu trữ dữ liệu từ các bảng thống kê cầu thủ từ FBRef.

+ Các bước thực hiện:

Gửi yêu cầu GET: Sử dụng requests.get(url) để lấy nội dung trang web.

Phân tích HTML: Dùng BeautifulSoup để phân tích nội dung phản hồi từ trang.

Tìm bảng dữ liệu: Sử dụng find để xác định div chứa bảng thống kê dựa trên id_div.

Trích xuất dữ liệu: Tìm các bình luận trong div và phân tích HTML để lấy tất cả các hàng trong bảng.

Lưu dữ liệu: Dữ liệu thu thập được được lưu vào một từ điển, và sau đó chuyển đổi thành DataFrame để lưu thành file CSV. Đặc biệt, nếu bảng là về thủ môn, mã sẽ loại bỏ các cột không cần thiết để đảm bảo tính chính xác và gọn gàng.

3. Hàm clean_data()

+ Chức năng: Hàm này có nhiệm vụ làm sạch và gộp dữ liệu từ nhiều file CSV, nhằm tạo ra một bảng dữ liệu thống nhất cho các cầu thủ.

+ Các bước thực hiện:

Đọc dữ liệu: Mở file table1.csv chứa dữ liệu chính.

Gộp dữ liệu: Lần lượt đọc các file từ table3.csv đến table10.csv, gộp chúng vào file chính, đồng thời loại bỏ các cột trùng lặp.

Gộp thêm dữ liệu từ bảng thủ môn: Đọc và kết hợp dữ liệu từ table2.csv, tạo danh sách các cột cần thiết để gộp.

Tiền xử lý dữ liệu: Chuyển đổi giá trị trong cột 'minutes' thành số nguyên bằng cách loại bỏ dấu phẩy và lọc ra các cầu thủ có số phút chơi lớn hơn 90.

Sắp xếp và lưu kết quả: Cuối cùng, dữ liệu được sắp xếp theo tên cầu thủ và tuổi, và lưu vào file result.csv.

4. Phần Chính của Chương Trình

+ Mã bắt đầu với việc xác định URL của trang web và ID của bảng thống kê cần thu thập. Sau đó, hàm crawler được gọi cho bảng chính và các bảng phụ, cung cấp một quy trình tự động cho việc thu thập dữ liệu.

+ Cuối cùng, hàm clean_data() được gọi để xử lý, gộp và lưu trữ dữ liệu đã thu thập được thành một bảng duy nhất.

Kết Quả:

Kết quả cuối cùng là file result.csv, chứa thông tin về các cầu thủ với các thông số như số bàn thắng, tuổi tác và thời gian thi đấu. File này sẽ được sử dụng cho các phân tích và báo cáo trong tương lai, cung cấp cái nhìn sâu sắc về hiệu suất cầu thủ trong giải Ngoại Hạng Anh.

Câu 2:

2.1. Tìm Kiếm Cầu Thủ Top và Bottom.

Mục Tiêu:

Mã Python này được xây dựng nhằm tìm kiếm và hiển thị top 3 cầu thủ có điểm số cao nhất và thấp nhất cho một loạt chỉ số thống kê từ file result.csv. Điều này giúp người dùng dễ dàng xác định những cầu thủ xuất sắc và kém hơn trong các chỉ số cụ thể.

Thành Phần Chính

1. Thư Viện Sử Dụng:

pandas: Thư viện chính được sử dụng để đọc, xử lý và phân tích dữ liệu trong file CSV. Pandas cung cấp các công cụ mạnh mẽ để làm việc với dữ liệu dạng bảng.

2. Đọc Dữ Liệu:

+ Đọc file CSV: Dữ liệu được đọc từ file result.csv vào một DataFrame với lệnh pd.read_csv('result.csv').

+ Kiểm tra cột: Các cột trong DataFrame được in ra để người dùng có thể xác định các chỉ số có sẵn trong dữ liệu.

3. Hàm find_top_bottom_players(df, column):

+ Chức năng: Hàm này nhận DataFrame và tên một cột chỉ số, thực hiện tìm kiếm top 3 cầu thủ có điểm cao nhất và thấp nhất cho chỉ số đó.

+ Các bước thực hiện:

Chuyển đổi giá trị: Sử dụng pd.to_numeric để chuyển các giá trị không thể chuyển đổi thành số thành NaN, từ đó giúp loại bỏ các giá trị không hợp lệ.

Lọc dữ liệu: Loại bỏ các hàng có giá trị NaN để đảm bảo chỉ làm việc với các giá trị hợp lệ.

Tìm top và bottom: Sử dụng nlargest và nsmallest để tìm ra 3 cầu thủ có điểm cao nhất và thấp nhất cho chỉ số đã chỉ định.

4. Phân Tích Dữ Liệu:

+ Danh sách các chỉ số: Danh sách columns_to_analyze chứa tên các chỉ số mà bạn muốn phân tích, bao gồm các chỉ số như 'goals', 'assists', 'xG', 'Saves', v.v.

+ Kết quả: Mỗi chỉ số trong danh sách được kiểm tra xem có tồn tại trong DataFrame hay không. Nếu có, hàm find_top_bottom_players được gọi để tìm kiếm top và bottom cầu thủ, và kết quả được lưu vào từ điển results.

5. Hiển Thị Kết Quả

In kết quả: Cuối cùng, mã in ra top 3 cầu thủ và bottom 3 cầu thủ cho mỗi chỉ số, giúp người dùng dễ dàng nắm bắt được thông tin quan trọng về hiệu suất cầu thủ.

2.2. Thống Kê Dữ Liệu Từ Tệp CSV.

Mục Tiêu

Mã Python này được thiết kế nhằm đọc dữ liệu từ một tệp CSV (result.csv), tính toán các thống kê cơ bản như trung vị, trung bình, và độ lệch chuẩn cho các chỉ số của cầu thủ, sau đó lưu các kết quả này vào một tệp CSV khác (results2.csv). Mục tiêu là cung cấp cái nhìn tổng quát về hiệu suất cầu thủ theo từng đội cũng như tổng thể.

Thành Phần Chính

1. Thư Viện Sử Dụng

pandas: Thư viện chính được sử dụng để thao tác và phân tích dữ liệu dạng bảng.

2. Hàm get_statistics_from_csv(csv_file)

+ Chức năng: Hàm này thực hiện toàn bộ quá trình đọc dữ liệu, tính toán thống kê và lưu kết quả vào tệp CSV.

+ Các bước thực hiện:

Đọc Dữ Liệu:

- Dữ liệu được đọc từ tệp CSV sử dụng pd.read_csv(csv_file) và lưu vào DataFrame df.

- Danh sách các cột trong DataFrame được in ra để kiểm tra tính đầy đủ của dữ liệu.

- Kiểm Tra Cột 'team'

  Kiểm tra xem cột 'team' có tồn tại trong DataFrame hay không. Nếu không, một lỗi sẽ được raised.

- Tìm Kiếm Các Cột Kiểu Số

  Các cột có kiểu dữ liệu số được xác định và lưu vào danh sách numeric_columns_list.

- Tính Toán Thống Kê

  Trung Vị: Trung vị của mỗi chỉ số được tính toán cho toàn bộ cầu thủ và lưu vào median_all.

Trung Bình và Độ Lệch Chuẩn: Tương tự, trung bình và độ lệch chuẩn cũng được tính cho toàn bộ cầu thủ.

Các thống kê này sau đó được gộp lại thành một DataFrame overall_df.

- Tính Toán Thống Kê Theo Đội

Các thống kê tương tự (trung vị, trung bình và độ lệch chuẩn) được tính toán cho từng đội và lưu vào DataFrame team_df.

- Gộp Kết Quả và Lưu Vào Tệp CSV

Hai DataFrame (overall_df và team_df) được gộp lại thành một DataFrame cuối cùng final_df.

Kết quả được ghi vào tệp results2.csv mà không có chỉ số.

- In Kết Quả

Cuối cùng, nội dung của tệp results2.csv được đọc lại và in ra để kiểm tra tính chính xác của kết quả.

2.3.Vẽ Biểu Đồ Histogram Từ Dữ Liệu CSV.

Mục Tiêu

Mã Python này được thiết kế nhằm đọc dữ liệu từ tệp CSV (result.csv), sau đó tạo ra các biểu đồ histogram cho các chỉ số của cầu thủ. Mục tiêu là giúp trực quan hóa phân bố của các chỉ số này, cả cho toàn bộ cầu thủ và cho từng đội.

Thành Phần Chính

1. Thư Viện Sử Dụng

pandas: Để thao tác và phân tích dữ liệu dạng bảng.

matplotlib.pyplot: Để vẽ biểu đồ.

seaborn: Để tạo các biểu đồ đẹp hơn, đặc biệt là histogram.

os: Để thao tác với hệ thống tệp, như tạo thư mục.

2. Đọc Dữ Liệu

Dữ liệu được đọc từ tệp CSV sử dụng pd.read_csv('result.csv') và lưu vào DataFrame df.

3. Các Chỉ Số Cần Phân Tích

Danh sách các cột cần phân tích được xác định và lưu vào biến columns_to_analyze, bao gồm các chỉ số như goals, assists, xG, và nhiều chỉ số khác.

4. Tạo Thư Mục Lưu Biểu Đồ

Nếu thư mục histograms chưa tồn tại, mã sẽ tạo ra thư mục này bằng os.makedirs('histograms') để lưu trữ các biểu đồ histogram.

5. Vẽ Histogram

a. Cho Tất Cả Cầu Thủ

+ Đối với mỗi chỉ số trong columns_to_analyze, mã sẽ:

Kiểm tra xem cột có tồn tại trong DataFrame không.

Vẽ histogram sử dụng sns.histplot() với các thông số như kích thước, tiêu đề, nhãn trục, và lưu biểu đồ vào tệp PNG trong thư mục histograms.

b. Cho Từng Đội

+ Sau khi vẽ cho tất cả cầu thủ, mã sẽ:

Lấy danh sách các đội duy nhất từ cột team.

Lặp qua từng đội, lọc dữ liệu cho đội đó và vẽ histogram cho mỗi chỉ số tương tự như trên.

6. Thông Báo Kết Quả

Cuối cùng, mã in ra thông báo rằng các biểu đồ đã được lưu trong thư mục histograms.

2.4. Phân tích Dữ liệu Cầu thủ.

Mục Tiêu

Mã Python này được phát triển nhằm đọc dữ liệu từ tệp CSV (result.csv) và xác định đội bóng có điểm số cao nhất cho mỗi chỉ số trong danh sách các chỉ số cầu thủ. Mục tiêu là để có cái nhìn tổng quát về những đội bóng hàng đầu theo các chỉ số quan trọng trong bóng đá.

Thành Phần Chính của Mã

1. Thư Viện Sử Dụng:

   pandas: Thư viện chính để thao tác và phân tích dữ liệu dạng bảng.

2. Đọc Dữ Liệu:

   Dữ liệu được đọc từ tệp CSV bằng lệnh pd.read_csv('result.csv') và lưu vào một DataFrame gọi là df.

3. Các Chỉ Số Cần Phân Tích:

   Danh sách các chỉ số cầu thủ cần phân tích được định nghĩa trong biến columns_to_analyze, bao gồm các chỉ số như:

   goals: Số bàn thắng

   assists: Số pha kiến tạo

   xG: XG (Expected Goals)

   Các chỉ số khác liên quan đến hiệu suất của cầu thủ và đội bóng.

4. Hàm find_top_team_per_stat:

   Hàm này có nhiệm vụ tìm đội bóng có điểm số cao nhất cho mỗi chỉ số.

   Nó thực hiện việc:

   Chuyển đổi các giá trị không thể chuyển đổi sang số thành NaN.

   Loại bỏ các hàng có giá trị NaN.

   Tìm đội bóng có chỉ số cao nhất và trả về thông tin tương ứng.

5. Phân Tích Dữ Liệu:

Mã lặp qua danh sách các chỉ số và sử dụng hàm find_top_team_per_stat để tìm đội bóng hàng đầu cho mỗi chỉ số.

Kết quả được lưu trong từ điển results.

6. Hiển Thị Kết Quả:

Cuối cùng, mã hiển thị thông tin về đội bóng có chỉ số cao nhất cho mỗi chỉ số, bao gồm tên đội và giá trị chỉ số.

Kết Quả

Kết quả cuối cùng cho biết đội bóng nào có chỉ số cao nhất trong từng danh mục. Điều này có thể giúp các huấn luyện viên, nhà phân tích và người hâm mộ hiểu rõ hơn về hiệu suất của các đội bóng trong giải đấu.

Câu 3:

3.1. Phân loại Cầu thủ Sử dụng Thuật toán K-means.

Mục tiêu

Mã Python này được thiết kế nhằm phân loại cầu thủ dựa trên các chỉ số như bàn thắng không phải phạt đền, bàn thắng phạt đền, kiến tạo và số phút thi đấu. Mục tiêu là nhóm các cầu thủ tương tự nhau thành các cụm để dễ dàng phân tích và so sánh hiệu suất.

Thành phần chính

1. Thư viện sử dụng:

   pandas: Để thao tác và phân tích dữ liệu dạng bảng.

   sklearn.cluster.KMeans: Để áp dụng thuật toán phân loại K-means.

   matplotlib.pyplot: Để trực quan hóa kết quả phân loại.

2. Đọc dữ liệu:

   Dữ liệu được đọc từ tệp CSV result.csv và lưu vào DataFrame data.

3. Chọn các chỉ số cần thiết:

   Các cột được chọn cho phân loại bao gồm:

   'non-Penalty Goals': Số bàn thắng không phải phạt đền.

   'Penalty Goals': Số bàn thắng phạt đền.

   'assists': Số kiến tạo.

   'minutes': Tổng số phút thi đấu.

4. Xử lý dữ liệu:

Mã xóa bỏ các hàng có giá trị NaN trong DataFrame để đảm bảo rằng dữ liệu đủ sạch cho quá trình phân loại.

5. Chọn số lượng cụm K:

Số lượng cụm K được chọn là 3. Giá trị này có thể điều chỉnh dựa trên yêu cầu phân tích.

6. Khởi tạo và huấn luyện mô hình K-means:

Mô hình K-means được khởi tạo với số cụm đã chọn và huấn luyện trên các chỉ số đã chọn.

Nhãn cụm được dự đoán cho từng cầu thủ và được thêm vào DataFrame gốc.

7. Hiển thị kết quả:

In ra danh sách cầu thủ cùng với nhãn cụm của họ để dễ dàng theo dõi và phân tích.

8. Trực quan hóa:

Biểu đồ phân tán được tạo ra để minh họa phân loại cầu thủ theo số bàn thắng không phải phạt đền và số kiến tạo. Mỗi cụm được hiển thị bằng một màu sắc khác nhau.

Kết quả:

Mã đã hoàn thành và hiển thị các cụm cầu thủ dựa trên các chỉ số đã chọn. Biểu đồ phân tán cho thấy sự phân bố của cầu thủ trong không gian hai chiều của các chỉ số, giúp dễ dàng nhận diện các nhóm cầu thủ tương tự.

3.2. Phân loại Cầu thủ Sử dụng Thuật toán K-means và Phương pháp Elbow.

Mục tiêu

Mục tiêu của mã Python này là phân loại cầu thủ dựa trên các chỉ số như bàn thắng không phải phạt đền, bàn thắng phạt đền, kiến tạo và số phút thi đấu. Bằng cách sử dụng thuật toán K-means, mã giúp nhóm các cầu thủ tương tự nhau thành các cụm để dễ dàng phân tích và so sánh hiệu suất.

Thành phần chính

1. Thư viện sử dụng:

   pandas: Để thao tác và phân tích dữ liệu dạng bảng.

   sklearn.cluster.KMeans: Để áp dụng thuật toán phân loại K-means.

   matplotlib.pyplot: Để trực quan hóa kết quả phân loại.

2. Đọc dữ liệu:

   Dữ liệu được đọc từ tệp CSV result.csv và lưu vào DataFrame data.

3. Chọn các chỉ số cần thiết:

   Các cột được chọn cho phân loại bao gồm:

   'non-Penalty Goals': Số bàn thắng không phải phạt đền.

   'Penalty Goals': Số bàn thắng phạt đền.

   'assists': Số kiến tạo.

   'minutes': Tổng số phút thi đấu.

4. Xử lý dữ liệu:

   Mã xóa bỏ các hàng có giá trị NaN trong DataFrame mà không gây ra cảnh báo, nhằm đảm bảo dữ liệu sạch cho quá trình phân loại.

5. Tính WCSS (Within-Cluster Sum of Squares):

   WCSS được tính toán cho các giá trị K khác nhau từ 1 đến 10. Đây là thông số giúp đánh giá mức độ chặt chẽ của các cụm.

6. Vẽ biểu đồ Elbow:

Biểu đồ Elbow được tạo ra để xác định số lượng cụm tối ưu. Điểm gãy trong biểu đồ sẽ cho biết số cụm K phù hợp.

7. Chọn số lượng cụm K tối ưu:

   Dựa trên biểu đồ Elbow, số lượng cụm K được chọn là 3 (giá trị này có thể điều chỉnh tùy theo kết quả biểu đồ).

8. Khởi tạo và huấn luyện mô hình K-means:

   Mô hình K-means được khởi tạo với số cụm đã chọn và huấn luyện trên các chỉ số đã chọn.

   Nhãn cụm được dự đoán cho từng cầu thủ và được thêm vào DataFrame gốc.

9. Hiển thị kết quả:

   In ra danh sách cầu thủ cùng với nhãn cụm của họ để dễ dàng theo dõi và phân tích.

10. Trực quan hóa:

    Biểu đồ phân tán được tạo ra để minh họa phân loại cầu thủ theo số bàn thắng không phải phạt đền và số kiến tạo. Mỗi cụm được hiển thị bằng một màu sắc khác nhau.

Kết quả:

Mã đã hoàn thành và hiển thị các cụm cầu thủ dựa trên các chỉ số đã chọn. Biểu đồ Elbow giúp xác định số lượng cụm tối ưu, trong khi biểu đồ phân tán cho thấy sự phân bố của cầu thủ trong không gian hai chiều của các chỉ số, giúp dễ dàng nhận diện các nhóm cầu thủ tương tự.

3.3. Phân tích Thành phần Chính và Phân cụm Cầu thủ.

Mục tiêu

Mục tiêu của mã Python này là áp dụng phân tích thành phần chính (PCA) để giảm chiều dữ liệu và sử dụng thuật toán K-means để phân cụm cầu thủ dựa trên các chỉ số như bàn thắng không phải phạt đền, bàn thắng phạt đền, kiến tạo và số phút thi đấu. Phân tích này nhằm mục đích giúp hiểu rõ hơn về mối quan hệ giữa các cầu thủ và nhóm họ thành các cụm tương đồng.

Thành phần chính

1. Thư viện sử dụng:

   pandas: Để thao tác và phân tích dữ liệu dạng bảng.

   numpy: Để thực hiện các phép toán số học.

   matplotlib.pyplot: Để trực quan hóa dữ liệu.

   sklearn.decomposition.PCA: Để thực hiện phân tích thành phần chính.

   sklearn.preprocessing.StandardScaler: Để chuẩn hóa dữ liệu.

   sklearn.cluster.KMeans: Để áp dụng thuật toán K-means.

2. Tải dữ liệu:

   Dữ liệu được tải từ tệp CSV result.csv và lưu vào DataFrame data.

3. Lựa chọn các chỉ số cần phân tích:

   Các cột được chọn cho phân tích bao gồm:

   'non-Penalty Goals': Số bàn thắng không phải phạt đền.

   'Penalty Goals': Số bàn thắng phạt đền.

   'assists': Số kiến tạo.

   'minutes': Tổng số phút thi đấu.

4. Chuẩn hóa dữ liệu:

Dữ liệu được chuẩn hóa bằng cách sử dụng StandardScaler để đảm bảo rằng các chỉ số có cùng quy mô, giúp tăng cường hiệu quả của thuật toán phân cụm.

5. Áp dụng PCA:

Phân tích thành phần chính (PCA) được áp dụng để giảm chiều dữ liệu từ nhiều chỉ số xuống còn hai thành phần chính (PC1 và PC2). Điều này giúp trực quan hóa dữ liệu trong không gian hai chiều.

6. Phân cụm dữ liệu:

Sử dụng thuật toán K-means với số nhóm đã xác định là 3, nhãn cụm được dự đoán cho từng cầu thủ và được thêm vào DataFrame gốc.

7. Vẽ biểu đồ phân cụm:

Biểu đồ phân tán được tạo ra để hiển thị các cầu thủ trong không gian hai chiều của các thành phần chính. Mỗi cụm được hiển thị bằng một màu sắc khác nhau, cho phép nhận diện dễ dàng các nhóm cầu thủ tương tự.

Kết quả:

Mã đã hoàn thành và hiển thị biểu đồ phân cụm cầu thủ dựa trên PCA. Biểu đồ cho thấy sự phân bố của các cầu thủ trong không gian hai chiều, giúp dễ dàng nhận diện các nhóm cầu thủ với hiệu suất tương tự.

3.4. Biểu đồ Radar So sánh Cầu thủ.

Mục tiêu

Mã Python này được thiết kế để so sánh các chỉ số giữa hai cầu thủ bóng đá thông qua việc sử dụng biểu đồ radar. Biểu đồ radar là một công cụ trực quan hữu ích, cho phép hiển thị nhiều chỉ số đồng thời và giúp người dùng dễ dàng nhận thấy sự khác biệt giữa hai cầu thủ.

Thành phần chính

1. Thư viện sử dụng:

   pandas: Để thao tác và xử lý dữ liệu.

   numpy: Để thực hiện các phép toán mảng.

   matplotlib.pyplot: Để tạo biểu đồ và trực quan hóa dữ liệu.

2. Hàm radar_chart:

   Tham số đầu vào:

   player1 và player2: Tên của hai cầu thủ muốn so sánh.

   attributes: Danh sách các chỉ số cần so sánh (ví dụ: bàn thắng, kiến tạo, đường chuyền).

   Tải dữ liệu: Dữ liệu được tải từ tệp result.csv và chỉ số của từng cầu thủ được trích xuất dựa trên tên cầu thủ.

   Kiểm tra dữ liệu: Mã kiểm tra xem thông tin của cầu thủ có tồn tại trong DataFrame hay không.

   Tạo góc cho các chỉ số: Mã tạo một mảng góc cho các chỉ số để vẽ biểu đồ radar.

   Vẽ biểu đồ radar: Sử dụng matplotlib để vẽ biểu đồ, bao gồm việc lấp đầy màu sắc cho mỗi cầu thủ và thiết lập các nhãn cho các chỉ số.

3. Khối if __name__ == '__main__':

   Nhận đầu vào từ người dùng cho tên cầu thủ và các chỉ số cần so sánh.

Chia nhỏ chuỗi các chỉ số thành danh sách và gọi hàm radar_chart để vẽ biểu đồ.

Kết quả

Khi chạy chương trình, người dùng được yêu cầu nhập tên hai cầu thủ cùng với danh sách các chỉ số cần so sánh. Biểu đồ radar được tạo ra sẽ cho thấy sự phân bố của các chỉ số này, cho phép người dùng nhanh chóng nhận diện ưu điểm và nhược điểm của từng cầu thủ trong các lĩnh vực cụ thể.