

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

----------

**BÁO CÁO**

**BÀI TẬP LỚN PYTHON**

**Giảng viên: Kim Ngọc Bách**

**Sinh viên: Nguyễn Văn Đạt**

**Mã sinh viên:B22DCCN199**

**Lớp: D22CQCN07-B**

**Hà Nội, 2024**

**MỤC LỤC**

[Bài 1: Viết chương trình Python thu thập dữ liệu phân tích cầu thủ của tất cả các cầu thủ có số phút thi đấu nhiều hơn 90 phút tại giải bóng đá ngoại hạng Anh mùa 2023-2024. 3](#_Toc181393776)

[1.1 Lấy dữ liệu từ url 3](#_Toc181393777)

[1.2 Tìm liên kết đội bóng và tên đội bóng 3](#_Toc181393778)

[1.3 Lưu Trữ và Trích Xuất Chỉ Số Từ Các Đội Bóng 4](#_Toc181393779)

[1.4 Tổng hợp dữ liệu 6](#_Toc181393780)

[Bài 2: Phân Tích Chỉ Số Cầu Thủ và Đội Bóng Giải Ngoại Hạng Anh 2023-2024 6](#_Toc181393781)

[2.1 Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số 6](#_Toc181393782)

[2.2 Tìm trung vị của mỗi chỉ số. Tìm trung bình và độ lệch chuẩn của mỗi chỉ số cho các cầu thủ trong toàn giải và của mỗi. 8](#_Toc181393783)

[2.3 Vẽ historgram phân bố của mỗi chỉ số của các cầu thủ trong toàn giải và mỗi đội 9](#_Toc181393784)

[2.4 Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số. Theo bạn đội nào có phong độ tốt nhất giải ngoại Hạng Anh mùa 2023-2024 10](#_Toc181393785)

# Bài 1: Viết chương trình Python thu thập dữ liệu phân tích cầu thủ của tất cả các cầu thủ có số phút thi đấu nhiều hơn 90 phút tại giải bóng đá ngoại hạng Anh mùa 2023-2024.

## Lấy dữ liệu từ url

* Dữ liệu được lấy từ website có địa chỉ url = **'https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats**'.
* Khi có được url ta phân tích và lấy dữ liệu bằng cách sử sụng các thư viện có sẵn

url = 'https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats'

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

tables\_link = soup.find\_all('table')  # Tìm tất cả các bảng

table\_link = tables\_link[0]

rows = table\_link.find\_all('tr')

team\_links = []

team\_names = []

for row in rows:

* Requests và BeautifulSoup trong Python để lấy và phân tích cú pháp nội dung HTML từ một URL.
* Sử dụng BeautifulSoup để phân tích cú pháp và tìm các bảng (table) trong trang web

với **tables\_link[0]** là danh sách chứa các bảng chi tiết về các đội.

* Bảng đầu tiên trong danh sách **tables\_link** chứa dữ liệu về các đội bóng, vì vậy ta chọn **table\_link = tables\_link[0]** để lấy dữ liệu chi tiết.
* **table\_link.find\_all('tr'):** find\_all sẽ tìm tất cả các thẻ <tr> trong table\_link và trả về một danh sách các thẻ này. Mỗi phần tử trong danh sách đại diện cho một hàng của bảng.
* **rows =** **table\_link.find\_all('tr'):** sẽ là một danh sách các hàng (tức là các thẻ <tr>) trong bảng.
* Sau khi đã có được rows thì bắt đầu tạo ra các danh sách để lưu các liên kết đến trang thông tin chi tiết của từng đội(**team\_links**) và tên của các đội (**team\_names**).
* Sau đó bắt đầu duyệt qua từng hàng trong rows .

## Tìm liên kết đội bóng và tên đội bóng

* Với mỗi lần duyệt qua từng hàng trong rows ta sẽ lấy ra được tên của đội bóng và liên kết đến trang thông tin chi tiết của từng đội.
* **team\_cell:** Dòng này tìm một ô (th) chứa thông tin về đội với class là 'left' và data-stat là 'team'.
* Nếu ô này chứa thẻ <a> với thuộc tính **'href'**, liên kết sẽ được nối với **'https://fbref.com/'** để tạo URL đầy đủ, rồi thêm vào danh sách **team\_links**.

**team\_cell = row.find('th', {'class'**: 'left', 'data-stat': 'team'})

    if team\_cell:

        a\_tag = team\_cell.find('a')

        if a\_tag and 'href' in a\_tag.attrs:

            link = 'https://fbref.com/' + a\_tag['href']

            team\_links.append(link)

* **team\_tag:** Dòng này tìm một ô (th) chứa thông tin về tên đội với thuộc tính data-stat là ‘team’.
* Nếu team\_tag tồn tại và chứa thẻ <a> thì thêm vào danh sách **team\_names**.

team\_tag = row.find('th', {'data-stat': 'team'})

    if team\_tag and team\_tag.find('a'):

      team\_name = team\_tag.find('a').text.strip()

      team\_names.append(team\_name) # Thêm tên đội vào danh sách

## Lưu Trữ và Trích Xuất Chỉ Số Từ Các Đội Bóng

* Sau khi có thông tin về các đội và địa chỉ liên kết đến trang thì tạo ra các danh danh sách lưu trữ các chỉ số cần thu thập gồm 10 danh sách.

players\_data = []

goalkeeping\_data = []

shooting\_data = []

passing\_data = []

passtype\_data = []

goal\_shot\_data = []

defensive\_data = []

playing\_time\_data = []

miscellaneous\_data = []

possession\_data = []

* Sử dụng vòng lặp để lặp qua các đường dẫn dẫn đến thông tin chi tiết về từng thành viên trong đội đó.
* Với mỗi lần lặp thì cần Delay thời gian để không bị chặn truy cập và tìm tất cả các table có trong đường dẫn vì các chỉ số cần thu thập nằm trong 10 table.

for link in team\_links:

    response = requests.get(link)

    time.sleep(4) # Delay thoi gian khong thi bi chan

    soup = BeautifulSoup(response.text, 'html.parser')

    tables = soup.find\_all('table')  # Tìm tất cả các bảng

* Với **table = tables[0]** ở đây thì nó chứa các chỉ số cần thu thập bao gồm:
  + Nation, Team, Position, Age
  + Playing time: matches played, starts, minutes
  + Performance: non-PenaltyGoals, PenaltyGoals, Assists, YellowCards,RedCards
  + Expected: xG, npxG, xAG,
  + Progression: PrgC, PrgP, PrgR
  + Per 90 minutes: Gls, Ast, G+A, G-PK, G+A-PK, xG, xAG, xG+xAG, npxG, npxG + xAG
* Với mỗi bảng cần kiểm tra số phút thi đấu của cầu thủ.
* Điều kiện này đảm bảo chỉ thu thập thông tin của cầu thủ đã thi đấu tương đương ít nhất 90 phút. Cột minutes\_90s chứa số lần cầu thủ chơi trọn vẹn 90 phút chia cho 90, nên cầu thủ cần có giá trị từ 1 trở lên để được xét vào danh sách.

table = tables[0]

minutes = row.find('td', {'data-stat': 'minutes\_90s'})

if minutes and minutes.text.strip() and float(minutes.text.strip()) >= 1

* Sau đó thu thập các thông tin và chỉ số cần thiết. Chỉ số nào không có hoặc không áp dụng đề là ‘N/a’ và khi thu thập xong thì lưu vào danh sách tương ứng.

player\_info = {

    'Player': row.find('th', {'data-stat': 'player'}).text.strip() ,

    'Nation': row.find('td', {'data-stat': 'nationality'}).text.strip().split()[-1],

    'Team': team\_names[i],

    'Pos': row.find('td', {'data-stat': 'position'}).text.strip(),

    'Age': row.find('td', {'data-stat': 'age'}).text.strip(),

    # Playing time

    'matches\_played': row.find('td', {'data-stat': 'games'}).text.strip() or 'N/a',

    'starts': row.find('td', {'data-stat': 'games\_starts'}).text.strip() or 'N/a',

    'minutes': row.find('td', {'data-stat': 'minutes'}).text.strip() or 'N/a',

* Sau đó thay thế table bằng lần lượt các table có các chỉ số cần thu thập cần thiết nằm trong tables:

table = tables[2]  # Bảng Goalkeeping

minutes = row.find('td', {'data-stat': 'minutes\_90s'})

if minutes and minutes.text.strip() and float(minutes.text.strip()) >= 1

* Với **table = tables[2]** chứa các chỉ số về **Goalkeeping** bao gồm:
  + Performance: GA, GA90, SoTA, Saves, Save%, W, D, L, CS, CS%
  + Penalty Kicks: PKatt, PKA, PKsv, PKm, Save%
* Sau đó thu thập chỉ số tương tự như **table = tables[0]** và lần lượt làm với các table = tables[4], tables[5], tables[6], tables[7], tables[8], tables[9], tables[10] và tables[11].

## 1.4 Tổng hợp dữ liệu

* Đầu tiên, chúng ta sẽ tạo các DataFrame từ các danh sách dữ liệu đã thu thập trước đó.
* Mỗi DataFrame sẽ chứa thông tin về một khía cạnh khác nhau của cầu thủ:

player = pd.DataFrame(players\_data)

goal = pd.DataFrame(goalkeeping\_data)

shoot = pd.DataFrame(shooting\_data)

passing = pd.DataFrame(passing\_data)

passtype = pd.DataFrame(passtype\_data)

goal\_shot = pd.DataFrame(goal\_shot\_data)

defensive = pd.DataFrame(defensive\_data)

playing\_time = pd.DataFrame(playing\_time\_data)

miscellaneous = pd.DataFrame(miscellaneous\_data)

possession = pd.DataFrame(possession\_data)

* Sau đó tạo một danh sách chứa tất cả các DataFrame cần gộp và sau đó thực hiện gộp dữ liệu dựa trên các cột chung như 'Player', 'Nation', 'Team', 'Pos', và 'Age'.

dataframes = [goal, shoot, passing, passtype, goal\_shot, defensive, possession, playing\_time, miscellaneous]

merged\_df = player

for df in dataframes:

    merged\_df = pd.merge(merged\_df, df, on=["Player", "Nation", "Team", "Pos", "Age"], how="left")

* Sau khi gộp xong, chúng ta sẽ tách phần tên đầu tiên từ cột 'Player' để dễ dàng sắp xếp sau này:

merged\_df['First Name'] = merged\_df['Player'].str.split().str[0]

sorted\_df = merged\_df.sort\_values(by=['First Name', 'Age'],

ascending=[True, False])

* Sau đó sắp xếp DataFrame theo 'First Name' và 'Age', với tên được sắp xếp theo thứ tự tăng dần và tuổi theo thứ tự giảm dần:
* Cuối cùng, chúng ta sẽ xóa cột 'First Name' vì nó không còn cần thiết trong kết quả cuối cùng và điền các giá trị rỗng là ‘N/a’.

sorted\_df = sorted\_df.drop(columns=['First Name'])

sorted\_df = sorted\_df.fillna("N/a")

sorted\_df.to\_csv('D:/results.csv', index=False, encoding='utf-8')

* Cuối cùng, DataFrame đã sắp xếp sẽ được ghi ra file results:

**Bài 2: Phân Tích Chỉ Số Cầu Thủ và Đội Bóng Giải Ngoại Hạng Anh 2023-2024**

**2.1 Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số**

* Sau khi có dữ liệu file từ câu 1 là results.csv thì đọc file CSV và chuyển thành DataFrame để dễ dàng phân tích
* Tạo danh sách colum chứa các cột chỉ số cần phân tích (loại bỏ các cột thông tin không phải chỉ số như 'Player', 'Nation', 'Team', 'Pos', 'Age').
* Duyệt qua từng cột trong colum và chuyển đổi các giá trị về kiểu số. Nếu có lỗi chuyển đổi, giá trị sẽ được gán là NaN để xử lý dễ dàng hơn sau đó.
* Sau đó tạo từ điển top\_players để lưu kết quả các cầu thủ có điểm cao nhất và thấp nhất cho từng chỉ số.

# Đọc file CSV và chuyển thành DataFrame

df = pd.read\_csv('D:/results.csv')

colum = [col for col in df.columns if col not in ['Player', 'Nation', 'Team', 'Pos', 'Age']]

for col in colum:

    df[col] = pd.to\_numeric(df[col], errors='coerce')

top\_players = {'Top 3 High': {}, 'Top 3 Low': {}}

* Duyệt qua từng chỉ số trong colum.
* Nếu một cột có toàn giá trị là NaN, bỏ qua.
* Với mỗi cột chỉ số:
  + Dùng nlargest(3, col) để lấy top 3 cầu thủ có giá trị cao nhất.
  + Dùng nsmallest(3, col) để lấy top 3 cầu thủ có giá trị thấp nhất.
* Lưu các kết quả vào từ điển top\_players.
* df[['Player', col]] giữ lại cả hai cột Player (tên cầu thủ) và col (chỉ số hiện tại).

for col in colum:

    # Bỏ qua cột nào có toàn giá trị NaN sau khi chuyển đổi

    if df[col].notna().sum() == 0:

        continue

    # Lấy top 3 cao nhất cho chỉ số

    top\_high = df[['Player', col]].nlargest(3, col)

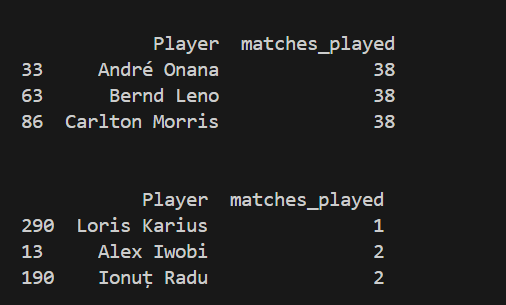
    # Lấy top 3 thấp nhất cho chỉ số

    top\_low = df[['Player', col]].nsmallest(3, col)

    top\_players['Top 3 High'][col] = top\_high

    top\_players['Top 3 Low'][col] = top\_low

* Kết quả như sau:



Top 3 cầu thủ có chỉ số cao nhất và thấp nhất ở matches\_played

## 2.2 Tìm trung vị của mỗi chỉ số. Tìm trung bình và độ lệch chuẩn của mỗi chỉ số cho các cầu thủ trong toàn giải và của mỗi.

* Tạo danh sách colum chứa tên các cột mà không bao gồm 'Player', 'Nation', 'Team', 'Pos', và 'Age'.
* Duyệt qua từng cột trong colum, chuyển đổi dữ liệu của cột thành kiểu số bằng cách:
  + Chuyển đổi thành kiểu chuỗi (astype(str)).
  + Loại bỏ dấu phẩy (str.replace(',', '')).
  + Sử dụng pd.to\_numeric() để chuyển đổi thành số, với errors='coerce' để biến các giá trị không thể chuyển đổi thành NaN.
* Tạo một DataFrame all\_stats chứa các thống kê (median, mean, std) cho tất cả các cột trong colum.
* Sử dụng dictionary comprehension để tính toán các giá trị thống kê cho từng cột.
* Thêm cột 'Team' với giá trị 'all' vào đầu DataFrame để chỉ định rằng các thống kê này là cho toàn giải đấu.

df = pd.read\_csv('D:/results.csv')

colum = [col for col in df.columns if col not in ['Player', 'Nation', 'Team', 'Pos', 'Age']]

for col in colum:

    df[col] = pd.to\_numeric(df[col].astype(str).str.replace(',', ''), errors='coerce')

all\_stats = pd.DataFrame(

    {f'Median of {col}': [df[col].median()] for col in colum}|

    {f'Mean of {col}': [df[col].mean()] for col in colum}|

    {f'Std of {col}': [df[col].std()] for col in colum}

    )

all\_stats.insert(0, 'Team', 'all')

* Sử dụng groupby() để nhóm dữ liệu theo cột 'Team', sau đó sử dụng agg() để tính toán median, mean và std cho từng cột trong colum. reset\_index() để đưa chỉ số về dạng chuẩn.
* Kết hợp all\_stats với team\_stats: Sử dụng pd.concat() để ghép hai DataFrame all\_stats và team\_stats thành một DataFrame duy nhất merged\_df. Tham số ignore\_index=True giúp đánh số lại các chỉ số trong DataFrame kết quả.
* Sử dụng to\_csv() để lưu DataFrame merged\_df vào file CSV

team\_stats = df.groupby('Team')[colum].agg(['median', 'mean', 'std']).reset\_index()

team\_stats.columns = ['Team'] + [f'{stat.capitalize()} of {col}' for col in colum for stat in ['median', 'std', 'mean']]

merged\_df = pd.concat([all\_stats, team\_stats], ignore\_index=True)

merged\_df.to\_csv('D:/results2.csv')

## 2.3 Vẽ historgram phân bố của mỗi chỉ số của các cầu thủ trong toàn giải và mỗi đội

* Sử dụng pd.read\_csv() để đọc dữ liệu từ file CSV và lưu vào một DataFrame có tên là df.
* Tạo danh sách colum chứa tên các cột mà không bao gồm 'Player', 'Nation', 'Team', 'Pos', và 'Age'.
* Vẽ histogram cho toàn giải đấu:
  + Vòng lặp: Duyệt qua từng cột trong colum để vẽ histogram cho mỗi chỉ số.
  + plt.figure(figsize=(10, 6)): Tạo một hình mới với kích thước 10x6 inch.
  + sns.histplot(df[col].dropna(), kde=True): Vẽ histogram cho cột hiện tại. dropna() được sử dụng để loại bỏ các giá trị thiếu (NaN) trước khi vẽ.
  + Tham số kde=True thêm đường phân phối xác suất (kernel density estimate) vào biểu đồ.
  + plt.title(), plt.xlabel(), plt.ylabel(): Đặt tiêu đề cho biểu đồ, nhãn cho trục x (chỉ số) và nhãn cho trục y (tần suất).
  + plt.show(): Hiển thị biểu đồ.

df = pd.read\_csv('D:/results.csv')

colum = [col for col in df.columns if col not in ['Player', 'Nation', 'Team', 'Pos', 'Age']]

for col in colum:

    plt.figure(figsize=(10, 6))

    sns.histplot(df[col].dropna(), kde=True)

    plt.title(f'Distribution of {col} for All Players')

    plt.xlabel(col)

    plt.ylabel('Frequency')

    plt.show()

* Vẽ histogram cho từng đội:
  + for team, group in df.groupby('Team'): Nhóm dữ liệu theo cột 'Team'. team là tên đội, và group là DataFrame chứa dữ liệu của đội đó.
  + Vòng lặp: Duyệt qua từng cột trong colum để vẽ histogram cho mỗi chỉ số của từng đội.
  + Quy trình vẽ biểu đồ tương tự như phần trước, chỉ khác ở tiêu đề và dữ liệu được vẽ (dữ liệu của đội hiện tại group).

for team, group in df.groupby('Team'):

    for col in colum:

        plt.figure(figsize=(10, 6))

        sns.histplot(group[col].dropna(), kde=True)

        plt.title(f'Distribution of {col} for Team {team}')

        plt.xlabel(col)

        plt.ylabel('Frequency')

        plt.show()

## 2.4 Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số. Theo bạn đội nào có phong độ tốt nhất giải ngoại Hạng Anh mùa 2023-2024

* Tạo danh sách colum chứa tên các cột mà không bao gồm 'Player', 'Nation', 'Team', 'Pos', và 'Age'.
* Duyệt qua từng cột trong colum, và chuyển đổi dữ liệu của cột thành kiểu số
* Sử dụng groupby('Team') để nhóm dữ liệu theo cột 'Team' và tính trung bình (mean()) cho các chỉ số trong colum. Kết quả là một DataFrame mới team\_means, trong đó mỗi hàng tương ứng với một đội và các cột chứa giá trị trung bình của các chỉ số.
* Sử dụng idxmax() để tìm tên đội có điểm số cao nhất cho mỗi chỉ số. Kết quả là một Series top\_teams, trong đó chỉ số là tên các chỉ số và giá trị là tên đội bóng có điểm số cao nhất cho chỉ số đó.

df = pd.read\_csv('D:/results.csv')

colum = [col for col in df.columns if col not in ['Player', 'Nation', 'Team', 'Pos', 'Age']]

for col in colum:

    df[col] = pd.to\_numeric(df[col].astype(str).str.replace(',', ''), errors='coerce')

team\_means = df.groupby('Team')[colum].mean()

top\_teams = team\_means.idxmax()

* Để xem đội nào có phong độ tốt nhất giải ngoại Hạng Anh mùa 2023-2024 thì em sẽ kiểm tra xem đội nào có số các chỉ số cao nhất là nhiều nhất
* team\_means.max() tìm giá trị tối đa trong từng cột của DataFrame team\_means, tức là tìm điểm số cao nhất cho mỗi chỉ số.
* team\_means == team\_means.max() tạo ra một DataFrame boolean với giá trị True cho những ô mà giá trị tương ứng bằng điểm số cao nhất và False cho những ô còn lại.
* .sum(axis=1) tính tổng số lượng True (hoặc điểm số cao nhất) cho từng hàng (từng đội). Kết quả sẽ là một Series, trong đó chỉ ra số lượng chỉ số mà mỗi đội có được giá trị cao nhất.
* highest\_counts.idxmax() tìm chỉ số của đội có số lượng chỉ số cao nhất (tức là hàng có giá trị lớn nhất trong Series highest\_counts). Chỉ số này tương ứng với tên đội có nhiều chỉ số cao nhất. highest\_counts.max() tìm giá trị tối đa trong Series highest\_counts, tức là số lượng chỉ số mà đội có điểm số cao nhất.

highest\_counts = (team\_means == team\_means.max()).sum(axis=1)

max\_highest\_count\_team = highest\_counts.idxmax()

max\_count = highest\_counts.max()

* Kết quả : Đội có nhiều chỉ số cao nhất là: Manchester City với 63 chỉ số.
* Vậy đội Manchester City là đội có phong độ tốt nhất giải ngoại Hạng Anh mùa 2023-2024.