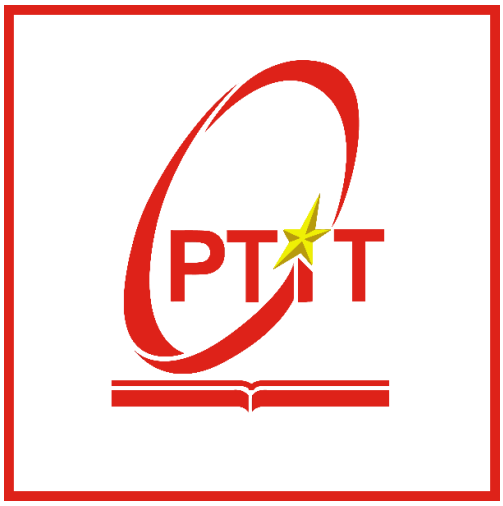
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



BÁO CÁO BÀI TẬP

MÔN HỌC: LẬP TRÌNH VỚI PYTHON

Giảng viên : Thầy Kim Ngọc Bách

Sinh viên : Lê Văn Hiệp

Mã sinh viên : B22DCCN296

Tháng 11/2024

CONTENTS

Bài 1:

I. Phân tích yêu cầu bài toán

1.Mục tiêu của bài toán

Viết chương trình thu thập dữ liệu thống kê về các cầu thủ thi đấu tại giải Ngoại hạng Anh mùa 2023-2024 từ trang fbref.com, sau đó xử lý và ghi kết quả ra file results.csv. Mỗi cầu thủ phải có đủ thông tin về các chỉ số quy định, và chỉ những cầu thủ có số phút thi đấu lớn hơn 90 phút mới được tính. Dữ liệu sau khi thu thập được sắp xếp và xử lý theo thứ tự cụ thể để đảm bảo đúng định dạng yêu cầu.

2. Yêu cầu chi tiết và cấu trúc dữ liệu đầu ra

2.1 Nguồn dữ liệu

- **Website**: https://fbref.com/en/
- **Đối tượng thu thập**: Tất cả các cầu thủ có số phút thi đấu nhiều hơn 90 phút tại giải Ngoại hạng Anh mùa giải 2023-2024.
- **URL tham khảo cụ thể**: Một ví dụ về trang dữ liệu của một đội bóng là trang thống kê của Liverpool: https://fbref.com/en/squads/822bd0ba/2023-2024/Liverpool-Stats. Từ đây, có thể suy ra cách tổ chức các URL của các đội khác.

2.2 Các chỉ số cần thu thập

Mỗi cầu thủ phải có các chỉ số sau. Mỗi cột của bảng kết quả sẽ tương ứng với một chỉ số thống kê, và nếu chỉ số nào không có hoặc không áp dụng, thì sẽ thay bằng "N/a".

- **Thông tin cơ bản**:
  - Nation: Quốc gia
  - Team: Đội bóng
  - Position: Vị trí
  - Age: Tuổi
- **Thời gian thi đấu**:
  - Matches Played: Số trận đã thi đấu
  - Starts: Số lần ra sân từ đầu trận
  - Minutes: Số phút thi đấu
- **Hiệu suất thi đấu**:
  - Non-Penalty Goals: Số bàn thắng không bao gồm phạt đền
  - Penalty Goals: Số bàn thắng từ phạt đền
  - Assists: Kiến tạo
  - Yellow Cards: Số thẻ vàng
  - Red Cards: Số thẻ đỏ
- **Các chỉ số kỳ vọng**:

    - xG: Expected Goals (bàn thắng kỳ vọng)
    - npxG: Non-Penalty Expected Goals (bàn thắng kỳ vọng không bao gồm phạt đền)
    - xAG: Expected Assists (kiến tạo kỳ vọng)
- **Chỉ số tiến bộ (Progression)**:
    - PrgC: Passes that progressed the ball towards the opponent's goal
    - PrgP: Progressive Passes
    - PrgR: Progressive Carries
- **Chỉ số mỗi 90 phút thi đấu**:
    - Gls, Ast, G+A, G-PK, G+A-PK, xG, xAG, xG + xAG, npxG, npxG + xAG
- **Chỉ số cho thủ môn**:
    - **Hiệu suất**:
        - GA: Goals Against
        - GA90: Goals Against per 90 minutes
        - SoTA: Shots on Target Against
        - Saves: Số lần cứu thua
        - Save%: Tỷ lệ cứu thua
        - W, D, L: Số trận thắng, hòa và thua
        - CS: Clean Sheets
        - CS%: Tỷ lệ Clean Sheets
    - **Phạt đền**:
        - PKatt: Penalty Kicks Attempted
        - PKA: Penalty Kicks Allowed
        - PKsv: Penalty Kicks Saved
        - PKm: Penalty Kicks Missed
        - Save%: Tỷ lệ cứu thua phạt đền
- **Chỉ số sút bóng (Shooting)**:
    - **Chuẩn**:
        - Gls: Goals
        - Sh: Shots

        - SoT: Shots on Target
        - SoT%: Tỷ lệ sút trúng đích
        - Sh/90: Số lần sút mỗi 90 phút
        - SoT/90: Số lần sút trúng đích mỗi 90 phút
        - G/Sh, G/SoT: Bàn thắng mỗi lần sút hoặc mỗi lần sút trúng đích
        - Dist: Khoảng cách sút trung bình
        - FK: Free Kicks
        - PK, PKatt: Số bàn thắng và số lần thực hiện phạt đền
    - **Kỳ vọng**:
        - xG, npxG, npxG/Sh, G-xG, np:G-xG
- **Chuyền bóng (Passing)**:
    - **Tổng thể**: Cmp, Att, Cmp%, TotDist, PrgDist
    - **Ngắn (Short)**: Cmp, Att, Cmp%
    - **Trung bình (Medium)**: Cmp, Att, Cmp%
    - **Dài (Long)**: Cmp, Att, Cmp%
    - **Kỳ vọng**:
        - Ast, xAG, xA, A-xAG, KP, 1/3, PPA, CrsPA, PrgP
- **Loại đường chuyền (Pass Types)**:
    - Các loại: Live, Dead, FK, TB, Sw, Crs, TI, CK
    - Phạt góc (Corner Kicks): In, Out, Str
    - Kết quả: Cmp, Off, Blocks
- **Tạo cơ hội ghi bàn và sút bóng (Goal and Shot Creation)**:
    - **SCA**: SCA, SCA90
    - **Loại tạo cơ hội (SCA Types)**: PassLive, PassDead, TO, Sh, Fld, Def
    - **GCA**: GCA, GCA90
    - **Loại tạo cơ hội ghi bàn (GCA Types)**: PassLive, PassDead, TO, Sh, Fld, Def
- **Hành động phòng ngự (Defensive Actions)**:
    - **Tackles**: Tkl, TklW, Def 3rd, Mid 3rd, Att 3rd
    - **Challenges**: Tkl, Att, Tkl%, Lost

- **Blocks**: Blocks, Sh, Pass, Int, Tkl + Int, Clr, Err

- **Kiểm soát bóng (Possession)**:
  - **Touches**: Touches, Def Pen, Def 3rd, Mid 3rd, Att 3rd, Att Pen, Live
  - **Take-Ons**: Att, Succ, Succ%, Tkld, Tkld%
  - **Carries**: Carries, TotDist, ProDist, ProgC, 1/3, CPA, Mis, Dis
  - **Receiving**: Rec, PrgR
- **Thời gian thi đấu và thành công của đội (Playing Time & Team Success)**:
  - **Bắt đầu (Starts)**: Starts, Mn/Start, Compl
  - **Dự bị (Subs)**: Subs, Mn/Sub, unSub
  - **Đội hình**: PPM, onG, onGA, onxG, onxGA
- **Các chỉ số khác (Miscellaneous Stats)**:
  - **Hiệu suất**: Fls, Fld, Off, Crs, OG, Recov
  - **Không chiến**: Won, Lost, Won%

3. Định dạng và sắp xếp dữ liệu

- Kết quả được ghi vào file results.csv.
- Sắp xếp cầu thủ theo tên (First Name). Nếu trùng tên, sắp xếp theo độ tuổi từ lớn đến nhỏ.
- Bất kỳ giá trị nào thiếu hoặc không áp dụng sẽ được điền là "N/a".

I. Giải quết vấn đề

1. Thuật toán

- Do cần thu thập dữ liệu từ nhiều đường dẫn khác nhau nên sẽ có 2 công việc chính:
  - 1 là lấy dữ liệu từ các đường dẫn
  - 2 là gộp các bảng lại và xóa các trường dữ liệu thừa
- Các thư viện được sử dụng trong bài tập này
  - Beautifulsoup
  - Requests
  - Pandas

1.1. Lấy dữ liệu từ các đường dẫn

- Các bước thực hiện
  - Gửi request tới dường dẫn để nhận file html về
  - Chuyển sang dạng tags html
  - Tìm kiếu thẻ div chứa nội dung cần lấy

- Vì nội dung cần lấy khi kéo về ở trong comment nên cần thực hiện lấy ra phần comments rồi đưa lại về dạng tags(ví dụ em để trong file output.html)
- Tên các cột lấy theo thuộc tính ‘data-stat’. Do giá trị của thuộc tính trung với tên cột
- Lưu dữ liệu vào dictionary rồi chuyển về dataframe
- Lưu bảng thành các file .csv

o Do nhận thấy các bảng từ 2-> 10 đều lấy toàn bộ bảng và xóa các trường dữ liệu giống nhau nên em dặt vào vòng lặp tách riêng so với thực hiện thu thập dữ liệu từ bảng 1.

1.2. Gộp các bảng lại và xóa các trường dữ liệu không cần thiết

o Các bảng 1 và từ 3 đến 10 dều có cột ‘Unname :0’(cột đánh chỉ số các cầu thủ) giống nhau.

⇨ Ghép các bảng 1 và từ 3 đến 10 dựa vào cột ‘Unname :0’.(gọi bảng này là bảng result)

o Do tên các thủ môn không trùng nhau nên bảng 2 sẽ ghép với bảng result đựa trên tên cầu thủ.

o Tiếp sẽ đưa chỉ số minutes về dạng số và xóa các cầu thủ có thời gian thi đầu <= 90 phút

o Sắp xếp lại bảng theo tên cầu thủ tăng dần và tuổi giảm dần

o Lưu bảng vào file result.csv

2. Kết quả:

[3]:

	Unnamed: 0	player	nationality	position	team	age	games	games_starts	minutes	assists	...	gk_wins	gk_ties	gk_losses	gk_clean_sheets	gk_clean_s
0	0	Max Aarons	eng	DF	Bournemouth	23	20	13	1237	1	...	NaN	NaN	NaN	NaN	
1	2	Tyler Adams	us	MF	Bournemouth	24	3	1	121	0	...	NaN	NaN	NaN	NaN	
2	3	Tosin Adarabioyo	eng	DF	Fulham	25	20	18	1617	0	...	NaN	NaN	NaN	NaN	
3	4	Elijah Adebayo	eng	FW	Luton Town	25	27	16	1419	0	...	NaN	NaN	NaN	NaN	
4	5	Simon Adingra	ci	FW	Brighton	21	31	25	2222	1	...	NaN	NaN	NaN	NaN	
5	6	Nayef Aguerd	ma	DF	West Ham	27	21	21	1857	0	...	NaN	NaN	NaN	NaN	
6	8	Naouirou Ahamada	fr	MF,FW	Crystal Palace	21	20	0	349	0	...	NaN	NaN	NaN	NaN	
7	9	Anel Ahmedhodžić	ba	DF	Sheffield Utd	24	31	29	2649	0	...	NaN	NaN	NaN	NaN	
8	10	Ola Aina	ng	DF	Nott'ham Forest	26	22	20	1692	1	...	NaN	NaN	NaN	NaN	
9	11	Rayan Aït-Nouri	dz	DF,MF	Wolves	22	33	29	2329	1	...	NaN	NaN	NaN	NaN	

Bài 2:

I. Phân tích yêu cầu bài toán

Dựa vào dữ liệu thu thập được ở câu 1 trong file 'results.csv', ta thực hiện các yêu cầu cần thực hiện.

1. Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số.

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 đội. Ghi kết quả ra file results2.csv, format như sau:

		Median of Atttribute 1	Mean of Atttribute 1	Std of Atttribute 1	...	...
0	all					
1	Team 1					
...						
n	Team n					

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

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.

II. Xử lý và giải quyết bài toán

Do bài toán có 4 yêu cầu vì vậy ta sẽ tách bài toán thành 4 file python với mỗi file sẽ xử lý 1 nhiệm vụ.

Đầu tiên ở mỗi đoạn code ta cần lấy dữ liệu từ file csv bằng thư viện Pandas và đưa giá trị của các chỉ số về dạng float nếu có thể.

```
import pandas as pd
from IPython.display import display
df = pd.read_csv('results.csv')
df.iloc[:, 5:] = df.iloc[:, 5:].replace(',', '', regex=True)
df.iloc[:, 4:] = df.iloc[:, 4:].apply(pd.to_numeric, errors='coerce')
pd.set_option('future.no_silent_downcasting', True)
df.fillna(0, inplace=True)
for x in df.columns[4:]:
    df[x] = df[x].astype(float)
```

1. Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất mỗi chỉ số

Ta duyệt theo từng cột và sắp xếp lại giá trị cột đó. Sau đó in ra top 3 vị trí đầu và cuối của cột đó.

```python
import pandas as pd
from IPython.display import display
df = pd.read_csv('results.csv')
df.iloc[:, 5:] = df.iloc[:, 5:].replace(',', '', regex=True)
df.iloc[:, 4:] = df.iloc[:, 4:].apply(pd.to_numeric, errors='coerce')
pd.set_option('future.no_silent_downcasting', True)
df.fillna(0, inplace=True)
for x in df.columns[4:]:
    df[x] = df[x].astype(float)
for x in df.columns[4:]:
    df_copy =df.copy()
    tmp =df_copy[['Player',x]].sort_values(x)
    tmp.dropna(axis=0, inplace=True)
    print(f'Top 3 cầu thủ có chỉ số {x} cao nhất')
    display(tmp[-3:][::-1])
    print(f'Top 3 cầu thủ có chỉ số {x} thấp nhất')
    display(tmp[:3])


```

Kết quả:

3 cầu thủ có Age cao nhất

	Player	Age
492	Łukasz Fabiański	38.0
47	Ashley Young	38.0
446	Thiago Silva	38.0

3 cầu thủ có Age thấp nhất

	Player	Age
277	Leon Chiwome	17.0
284	Lewis Miley	17.0
482	Wilson Odobert	18.0

3 cầu thủ có MP-Playing_Time-standard_stats cao nhất

	Player	MP-Playing_Time-standard_stats
210	James Tarkowski	38.0
346	Moussa Diaby	38.0
228	Joachim Andersen	38.0

3 cầu thủ có MP-Playing_Time-standard_stats thấp nhất

	Player	MP-Playing_Time-standard_stats
320	Matheus Nunes	2.0

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 đội. Ghi kết quả ra file results2.csv..

- Đầu tiên ta sẽ tạo 1 danh sách các tên cột để tính toán và gán lại vùng dữ liệu chỉ lấy những cột mà ta cần.

```
df2= df2[index2] # gán lại lấy những cột cần để xử lý thôi
df2=df2.set_index('Team') # đặt Team là chỉ số cho index
```

- Sau đó ta sẽ tính giá trị trung bình, trung vị, độ lệch chuẩn cho mỗi cột trong df2 theo chỉ số 'Team'.

```
mean = df2.groupby('Team').mean() # tính toán trung bình
median =df2.groupby('Team').median() # trung vị
std = df2.groupby('Team').std() # độ lệch chuẩn
```

- Tương tự ta tính toán cho toàn bộ cầu thủ.

```
mean_all= df2.mean()
median_all= df2.median()
std_all =df2.std()
```

- Cuối cùng là ta tạo ra các danh sách để lưu cột và giá trị cho kết quả cuối cùng. Và đưa giá trị vào trong đó thông qua các danh sách đã lưu từ trước.

```
result2 = dict()
col_result2 = []
result2['all']=[]
for x in mean.index:
    result2[x]=[]
for x in df.columns[5:]:
    col_result2.append(f'Median of {x}') # t
    col_result2.append(f'Mean of {x}')
    col_result2.append(f'Std of {x}')
    result2['all'].append(median_all[x]) # t
    result2['all'].append(mean_all[x])
    result2['all'].append(std_all[x])
    for y in mean.index:
        result2[y].append(median.loc[y][x])
        result2[y].append(mean.loc[y][x])
        result2[y].append(std.loc[y][x])
```

- Kết thúc yêu cầu là lưu kết quả ra file ‘results2.csv’.

```
result2=pd.DataFrame.from_dict(result2,orient='index',columns=col_result2)
result2.to_csv('results2.csv')
```

Kết quả thu được là:

	Median of MP-Playing_Time-standard_stats	Mean of MP-Playing_Time-standard_stats	Std of MP-Playing_Time-standard_stats	Median of Starts-Playing_Time-standard_stats	Mean of Starts-Playing_Time-standard_stats	Std of Starts-Playing_Time-standard_stats	Median of Min-Playing_Time-standard_stats	Mean of Min-Playing_Time-standard_stats	Std of Min-Playing_Time-standard_stats	Media Playing_Ti standard_
all	23.0	22.657201	10.136975	16.0	16.941176	11.167179	1419.0	1518.369168	949.241058	1
Arsenal	27.0	26.809524	10.191266	18.0	19.857143	13.093073	1649.0	1781.571429	1102.745872	1
Aston-Villa	27.0	24.173913	11.109587	20.0	18.130435	12.392462	1652.0	1629.130435	1057.004055	1
Bournemouth	25.5	22.076923	11.852166	13.0	16.038462	12.732575	1317.5	1438.423077	1074.136832	1
Brentford	26.0	22.960000	10.346014	15.0	16.720000	10.883933	1321.0	1496.840000	910.122459	1
Brighton-and-Hove-Albion	20.0	20.928571	8.751417	15.0	14.892857	8.603786	1344.5	1338.107143	768.792913	1
Burnley	16.0	20.392857	9.346575	14.0	14.928571	10.014540	1213.0	1334.857143	851.000706	1
Chelsea	23.0	21.880000	9.404432	18.0	16.720000	11.066165	1576.0	1495.120000	951.169820	1
Crystal-Palace	22.5	22.458333	9.477567	17.5	17.416667	10.993740	1587.5	1566.000000	910.738831	1
Everton	28.0	23.304348	11.561829	23.0	18.173913	13.720099	1884.0	1633.173913	1156.480384	2
Fulham	20.0	27.238095	7.993152	18.0	19.904762	10.084170	1593.0	1773.714286	834.245056	1

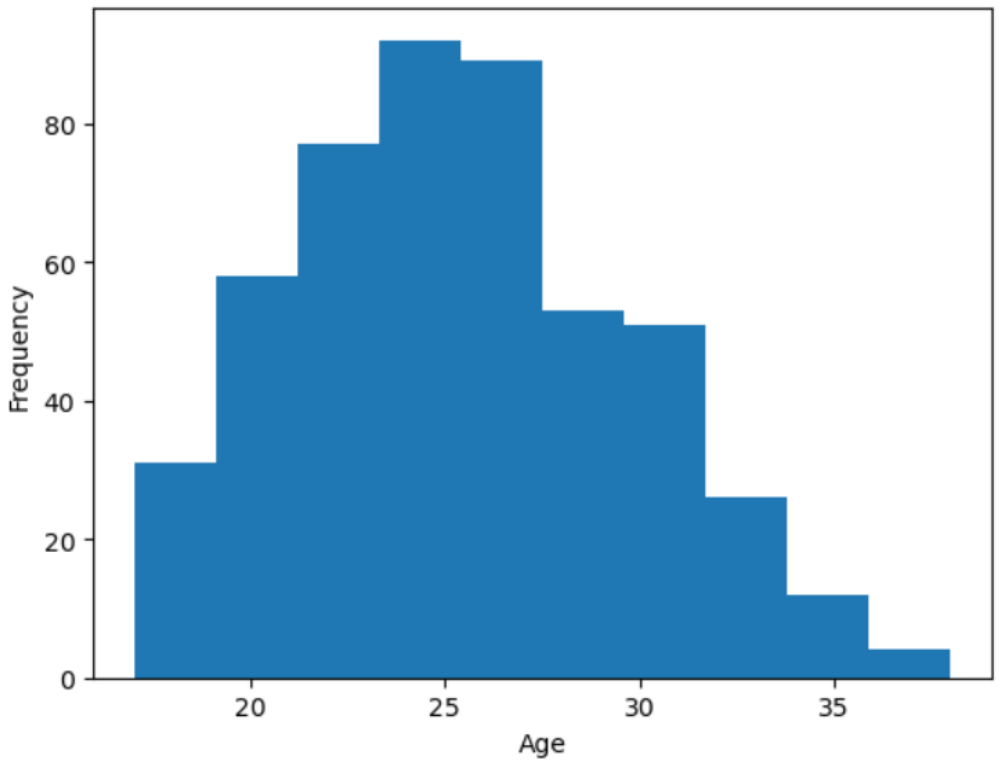
3. Vẽ historgram phân bố của mỗi chỉ số của các cầu thủ trong toàn giải và mỗi đội.

- Ta sử dụng thư viện Matplotlib.

```
plt.ion()
for i, x in enumerate(df.columns[4:]):
    print(f"Phân bố của {x}")
    tmp = df[x].dropna()
    plt.hist(tmp)
    plt.xlabel(x)
    plt.ylabel("Frequency")
    plt.draw()
    plt.pause(0.5)  # Tạm dừng có thể tăng lên nếu muốn
    plt.clf()  # Xóa biểu đồ hiện tại trước khi vẽ biểu đồ mới
plt.ioff()
```

Kết quả sẽ là hình ảnh các biểu đồ hiện lên và tắt đi chứ không cần phải tắt biểu đồ thủ công bằng tay để hiện biểu đồ tiếp theo.

```
Phân bố của mỗi chỉ số của các cầu thủ trong toàn giải

phân bố của Age
```



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.

- Ta chỉ cần lọc ra theo chỉ số team và tính toán xem đội nào có chỉ số đó cao nhất và in ra.

```python
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('results.csv')
df.iloc[:, 5:] = df.iloc[:, 5:].replace(',', '', regex=True)
df.iloc[:, 4:] = df.iloc[:, 4:].apply(pd.to_numeric, errors='coerce')
pd.set_option('future.no_silent_downcasting', True)
df.fillna(0, inplace=True)
for x in df.columns[4:]:
    df[x] = df[x].astype(float)
columns_to_sum = df.columns[4:].tolist()
team_stats = df.groupby('Team')[columns_to_sum].sum()
highest_team_stats = team_stats.idxmax()
for stat in team_stats.columns:
    print(f"'{stat}': {highest_team_stats[stat]}")
```

- Kết quả thu được là:

```
PS D:\pythonptit> python -u "d:\pythonptit\BTL2_4.py"
'Age': Nottingham-Forest
'MP-Playing_Time-standard_stats': Brighton-and-Hove-Albic
'Starts-Playing_Time-standard_stats': Brentford
'Min-Playing_Time-standard_stats': Crystal-Palace
'90s-Playing_Time-standard_stats': Crystal-Palace
'Gls-Performace-standard_stats': Manchester-City
'Ast-Performace-standard_stats': Manchester-City
'G+A-Performace-standard_stats': Manchester-City
'G-PK-Performace-standard_stats': Manchester-City
'PK-Performace-standard_stats': Chelsea
'PKatt-Performace-standard_stats': Chelsea
'CrdY-Performace-standard_stats': Chelsea
'CrdR-Performace-standard_stats': Burnley
'xG-Expected-standard_stats': Liverpool
'npxG-Expected-standard_stats': Liverpool
'xAG-Expected-standard_stats': Liverpool
'npxG+xAG-Expected-standard_stats': Liverpool
'PrgC-Progression-standard_stats': Manchester-City
'PrgP-Progression-standard_stats': Liverpool
'PrgR-Progression-standard_stats': Tottenham-Hotspur
'Gls-Per_90_Minutes-standard_stats': Newcastle-United
'Ast-Per_90_Minutes-standard_stats': Tottenham-Hotspur
'G+A-Per_90_Minutes-standard_stats': Newcastle-United
'G-PK-Per_90_Minutes-standard_stats': Newcastle-United
'G+A-PK-Per_90_Minutes-standard_stats': Newcastle-United
```

Từ đấy thấy rằng đội có phong độ tốt nhất là: **Manchester City**

Bài 3:

I. Phân tích yêu cầu bài toán

Dựa vào dữ liệu thu thập được ở câu 1 trong file ‘results.csv’, ta thực hiện các yêu cầu cần thực hiện.

1. Sử dụng thuật toán K-means để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau. Theo bạn thì nên phân loại cầu thủ thành bao nhiêu nhóm? Vì sao? Bạn có nhận xét gì về kết quả. Sử dụng thuật toán PCA, giảm số chiều dữ liệu xuống 2 chiều, vẽ hình phân cụm các điểm dữ liệu trên mặt 2D.

2. Viết chương trình python vẽ biểu đồ rada (radar chart) so sánh cầu thủ với đầu vào như sau:

+ python radarChartPlot.py --p1 <player Name 1> --p2 <player Name 2> --Attribute <att1,att2,...,att_n>
+ --p1: là tên cầu thủ thứ nhất.
+ --p2: là tên cầu thủ thứ hai.
+ --Attribute: là danh sách các chỉ số cần so sánh.

II. Xử lý và giải quyết bài toán

Do bài toán gồm 2 nhiệm vụ riêng biệt nên ta sẽ tách ra làm 2 file python, 1 file là kmeans và pca còn file còn lại vẽ rada.

1. Sử dụng thuật toán K-means để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau. . Sử dụng thuật toán PCA, giảm số chiều dữ liệu xuống 2 chiều, vẽ hình phân cụm các điểm dữ liệu trên mặt 2D.

-Ban đầu lọc lại dữ liệu

```
# Đọc dữ liệu từ file CSV
df = pd.read_csv('results.csv')
df.iloc[:, 5:] = df.iloc[:, 5:].replace(',', '', regex=True)
df.iloc[:, 4:] = df.iloc[:, 4:].apply(pd.to_numeric, errors='coerce')
pd.set_option('future.no_silent_downcasting', True)
df.fillna(0, inplace=True)
for x in df.columns[4:]:
    df[x] = df[x].astype(float)
```

- Lấy dữ liệu cần phân nhóm và chuẩn hóa dữ liệu

```
indicator_columns = df.columns[4:]

data = df[indicator_columns]
# Chuẩn hóa dữ liệu
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

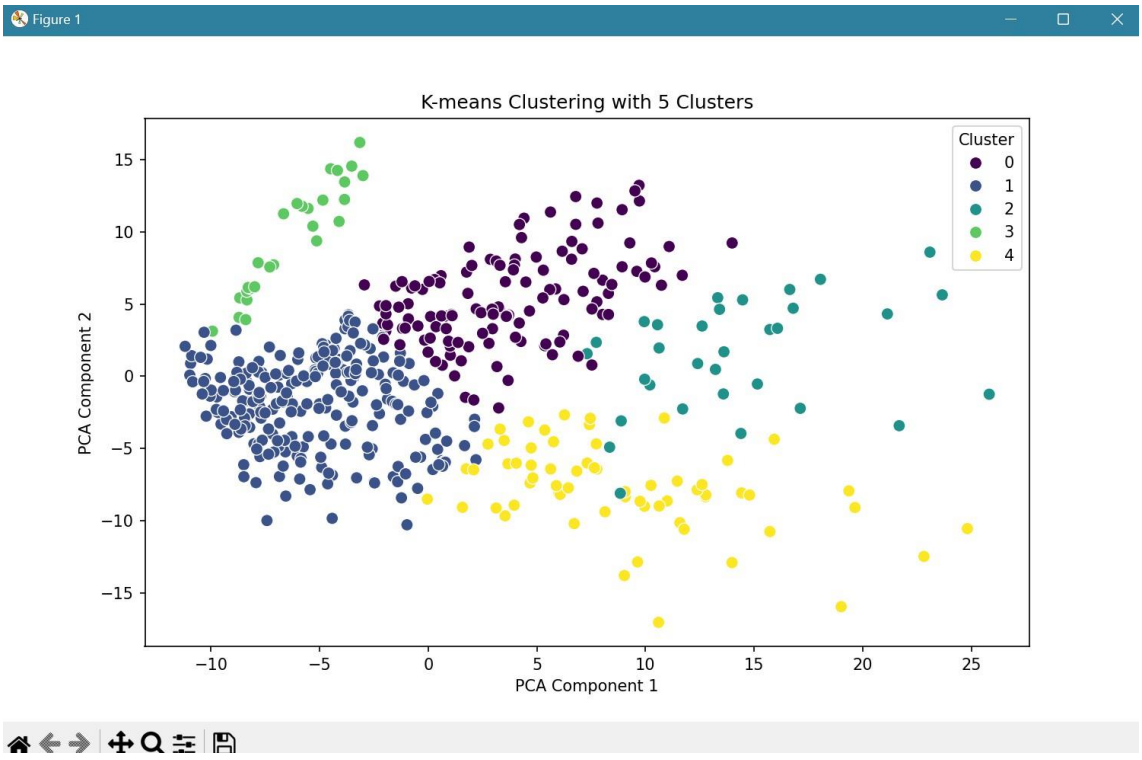
- Phân nhóm và gán nhãn cho các cầu thủ

```
# Sử dụng K-means để phân nhóm
n_clusters = 5  # Có thể thay đổi số lượng nhóm
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
kmeans.fit(data_scaled)

# Gán nhãn cho mỗi cầu thủ
df = df.loc[data.index]
df = pd.concat([df, pd.DataFrame({'Cluster': kmeans.labels_}, index=df.index)], axis=1)
```

- Giảm chiều và vẽ biểu đồ phân cụm

```
# Giảm chiều dữ liệu với PCA
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)

# Vẽ biểu đồ phân cụm
plt.figure(figsize=(10, 6))
sns.scatterplot(x=data_pca[:, 0], y=data_pca[:, 1], hue=df['Cluster'], palette='viridis', s=60)
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.title(f'K-means Clustering with {n_clusters} Clusters')
plt.legend(title='Cluster')
plt.show()
```

Kết quả :



Nên phân từ 3 tới 5 nhóm. Vì giả sử ta sử dụng phương pháp Elbow để tìm số lượng cụm phù hợp cho thuật toán K-means, thì thấy

điểm "khuỷu tay" ở khoảng từ 3 đến 5 cụm, cho thấy đó có thể là số lượng cụm phù hợp để phân nhóm cầu thủ.

2. Viết chương trình python vẽ biểu đồ rada (radar chart) so sánh cầu thủ

Tham khảo dựa trên đường dẫn:
https://matplotlib.org/stable/gallery/specialty_plots/radar_chart.html

- Đầu tiên ta đọc file dữ liệu từ câu 1, sau đó ta thiết lập phân tích tham số và tách các thuộc tính, lấy dữ liệu của các cầu thủ với thuộc tính đã tách. Cuối cùng là vẽ biểu đồ rada.

```
df = pd.read_csv('results.csv')
# Thiết lập phân tích tham số
parser = argparse.ArgumentParser(description='Radar Chart Comparison between Players')
parser.add_argument('--p1', type=str, required=True, help='Player 1 Name')
parser.add_argument('--p2', type=str, required=True, help='Player 2 Name')
parser.add_argument('--Attribute', type=str, required=True, help='Comma separated list of attributes')
args = parser.parse_args()
# Tách các thuộc tính
attributes = args.Attribute.split(',')

# Lấy dữ liệu cho các cầu thủ
player1_data = df.loc[df['Player'] == args.p1, attributes].values.flatten()
player2_data = df.loc[df['Player'] == args.p2, attributes].values.flatten()

# Kiểm tra xem dữ liệu có hợp lệ không
if player1_data.size == 0 or player2_data.size == 0:
    raise ValueError("Player not found or attributes invalid.")

# Vẽ biểu đồ radar
radar_chart(player1_data, player2_data, attributes)
```

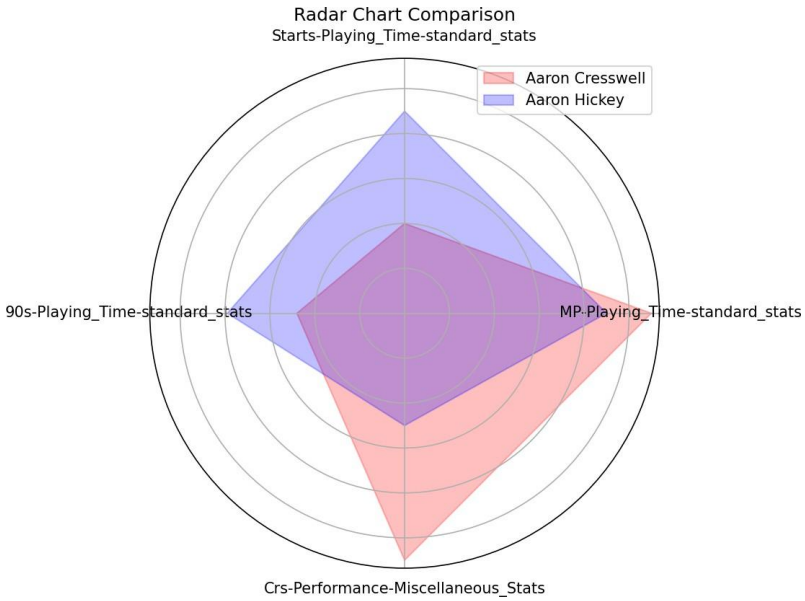
- Khi muốn vẽ biểu đồ rada so sánh giữa cầu thủ 1 và cầu thủ 2 ta chỉ cần gõ lệnh có dạng như sau ở terminal:

```
python radaChartPlot.py --p1 'Player1' --p2 'Player2' --Attribute 'attributes'
```

Ví dụ: python radarChartPlot.py --p1 "Aaron Cresswell" --p2 "Aaron Hickey" --Attribute "MP-Playing_Time-standard_stats,Starts-Playing_Time-standard_stats,90s-Playing_Time-standard_stats,Crs-Performance-Miscellaneous_Stats"

```
PS D:\pythonptit>
PS D:\pythonptit>
PS D:\pythonptit> python radarChartPlot.py --p1 "Aaron Cresswell" --p2 "Aaron Hickey" --Attribute "MP-Playing_Time-standard_stats,Starts-Playing_Time-standard_stats,90s-Playing_Time-standard_stats,Crs-Performance-Miscellaneous_Stats"
```

Kết quả :



Bài 4: Thu thập giá chuyển nhượng của các cầu thủ trong mùa 2023-2024 từ trang web https://www.footballtransfers.com. - Đề xuất phương pháp định giá cầu thủ.

1.Yêu cầu và phân tích bài toán định giá cầu thủ.

- Thu thập dữ liệu: Sinh viên cần thu thập giá chuyển nhượng của cầu thủ trong mùa giải 2023-2024 từ trang web footballtransfers.com.1

- Đề xuất phương pháp: Dựa trên dữ liệu đã thu thập, sinh viên cần đề xuất một phương pháp để định giá cầu thủ.1

Phân Tích Bài Toán

2.Thu thập dữ liệu:

- Nguồn dữ liệu: Trang web footballtransfers.com cung cấp thông tin về chuyển nhượng cầu thủ, bao gồm giá trị chuyển nhượng. Sinh viên cần sử dụng kỹ thuật web scraping để thu thập dữ liệu này.

- Dữ liệu cần thu thập: Ngoài giá chuyển nhượng, sinh viên có thể thu thập thêm các thông tin khác liên quan đến cầu thủ như: tên, tuổi, vị trí, câu lạc bộ, quốc tịch, số trận đấu, số bàn thắng, kiến tạo,... Các thông tin này có thể được sử dụng để xây dựng mô hình định giá.1

- Lưu trữ dữ liệu: Dữ liệu sau khi thu thập cần được lưu trữ một cách có hệ thống, ví dụ sử dụng file CSV hoặc cơ sở dữ liệu.

2. Đề xuất phương pháp định giá cầu thủ:

- Xây dựng mô hình: Sinh viên có thể sử dụng các kỹ thuật học máy (machine learning) để xây dựng mô hình định giá cầu thủ. Mô hình này sẽ dựa trên các đặc trưng của cầu thủ (tuổi, vị trí, số liệu thống kê,...) và giá chuyển nhượng của họ.

- Các mô hình tiềm năng:

+ Hồi quy tuyến tính: Mô hình đơn giản, dễ hiểu, phù hợp cho bài toán dự đoán giá trị liên tục.

+ Hồi quy phi tuyến: Phức tạp hơn hồi quy tuyến tính, có thể mô tả được các mối quan hệ phức tạp giữa các biến.

+ Cây quyết định: Dễ dàng diễn giải, trực quan, có thể xử lý được cả dữ liệu dạng số và dạng hạng mục.

+ Mạng nơ ron: Mô hình phức tạp, có khả năng học các mẫu phức tạp trong dữ liệu, nhưng khó diễn giải.

+ Đánh giá mô hình: Sau khi xây dựng mô hình, cần đánh giá hiệu quả của mô hình bằng cách sử dụng các độ đo như: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), R-squared,...

- Ứng dụng: Mô hình sau khi được xây dựng và đánh giá có thể được sử dụng để dự đoán giá trị của các cầu thủ khác hoặc phân tích yếu tố nào ảnh hưởng đến giá trị của cầu thủ.