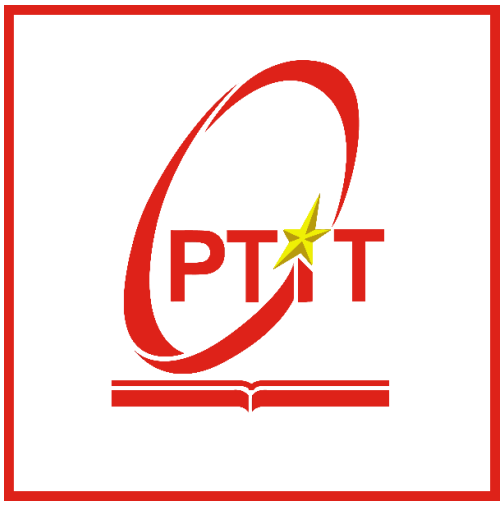
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



BÁO CÁO BÀI TẬP

MÔN HỌC: LẬP TRÌNH VỚI PYTHON

Giảng viên : Thầy Kim Ngọc Bách

Sinh viên : Võ Hoàng An

Lớp : D22CQCN08-B

Mã sinh viên : B22DCCN008

Tháng 10/2022

CONTENTS

Câu 1:

I. Phân tích yêu cầu bài toán

1. Mục tiêu chính của bài toán

Viết chương trình Python thu thập dữ liệu phân tích các cầu thủ thi đấu trên 90 phút tại giải bóng đá ngoại hạng Anh mùa giải 2023-2024, dữ liệu được lấy từ trang web https://fbref.com/en/ và ghi ra kết quả ra file 'results.csv' với kết cấu bảng theo yêu cầu đề ra.

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

Mỗi cột tương ứng với một chỉ số.

Thứ tự các cầu thủ sắp xếp theo thứ tự tên (First Name), nếu trùng tên thì xếp theo độ tuổi từ lớn đến nhỏ.

Chỉ số nào không có hoặc không áp dụng thì để là N/a

Các chỉ số cụ thể như sau:

+ Nation
+ Team
+ Position
+ Age
+ Playing time: matches played, starts, minutes
+ Performance: non-Penalty Goals, Penalty Goals, Assists, Yellow Cards, Red Cards
+ 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
+ Goalkeeping:
    - Performance: GA, GA90, SoTA, Saves, Save%, W, D, L, CS, CS%
    - Penalty Kicks: PKatt, PKA, PKsv, PKm, Save%
+ Shooting:
    - Standard: Gls, Sh, SoT, SoT%, Sh/90, SoT/90, G/Sh, G/SoT, Dist, FK, PK, PKat
    - Expected: xG, npxG, npxG/Sh, G-xG, np:G-xG
+ Passing:
    - Total: Cmp, Att, Cmp%, TotDist, PrgDist
    - Short: Cmp, Att, Cmp%
    - Medium: Cmp, Att, Cmp%
    - Long: Cmp, Att, Cmp%
    - Expected: Ast, xAG, xA, A-xAG, KP, 1/3, PPA, CrsPA, PrgP

+ Pass Types:
    - Pass Types: Live, Dead, FK, TB, Sw, Crs, TI, CK
    - Corner Kicks: In, Out, Str
    - Outcomes: Cmp, Off, Blocks
+ Goalkeeping:
    - SCA: SCA, SCA90
    - SCA Types: PassLive, PassDead, TO, Sh, Fld, Def
    - GCA: GCA, GCA90
    - GCA Types: PassLive, PassDead, TO, Sh, Fld, Def
+ 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
+ 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
+ Playing Time:
    - Starts: Starts, Mn/Start, Compl
    - Subs: Subs, Mn/Sub, unSub
    - Team Success: PPM, onG, onGA
    - Team Success xG: onxG, onxGA
+ Goalkeeping:
    - Performance: Fls, Fld, Off, Crs, OG, Recov
    - Aerial Duels: Won, Lost, Won%
+ Tham khảo: https://fbref.com/en/squads/822bd0ba/2023-2024/Liverpool-Stats

II. Các bước thực hiện

1. Ý tưởng và thư viện cần sử dụng

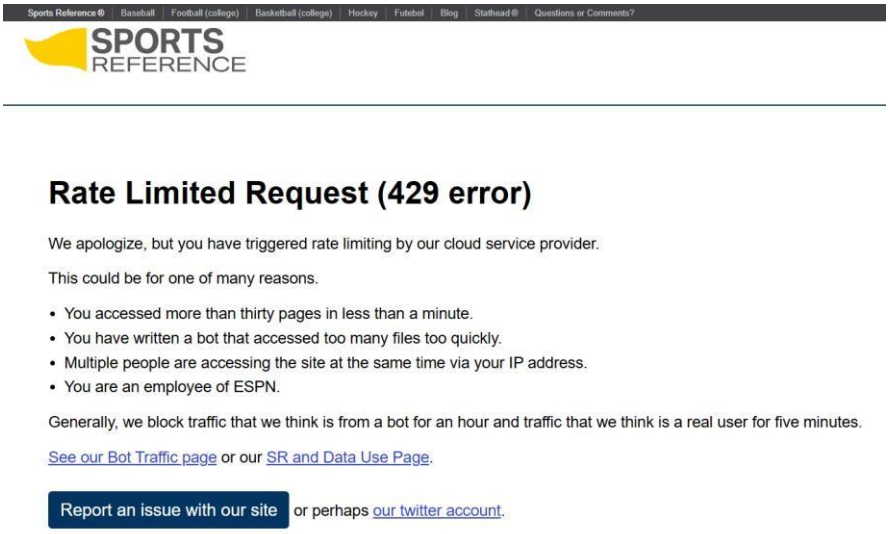
- Do thông tin trên các đường dẫn là tương tự nhau nên ta chỉ cần viết 1 hàm lấy dữ liệu trên 1 đường dẫn và lặp lại 20 lần với 20 đường dẫn tương đương với 20 đội bóng:
    - 1 là lấy dữ liệu từ bảng Standard Stats và lọc dữ liệu lấy các cầu thủ thi đấu trên 90 phút, do bảng này chứ tất cả các cầu thủ thi đấu cho 1 đội trong mùa giải và cho các cầu thủ thỏa mãn yêu cầu vào 1 danh sách.

- 2 là lấy dữ liệu chỉ số của các cầu thủ có trong danh sách trên trong các bảng còn lại, những cầu thủ không có tên trong danh sách trên thì không cần phải lấy.

- Các thư viện được sử dụng trong bài tập này
  - Beautifulsoup
  - Selenium
  - Pandas

2. Một số lỗi thường gặp

- Truy cập vào web lấy dữ liệu liên tục trong 1 thời gian ngắn qua nhiều đường link khiến web dễ hiểu lầm là bot và cấm địa chỉ IP của bạn truy cập trong vài phút khiến việc lấy dữ liệu bị gián đoạn.



- Để xử lý lỗi trên ta sử dụng thư viện selenium thay thế cho thư viện requests để lấy dữ liệu từ web bởi việc set thời gian chờ mỗi khi lấy dữ liệu từ trình duyệt được khởi tạo.

3. Quá trình thực hiện

- Đầu tiên ta tạo 1 list urls chứa các đường dẫn tới trang thống kê kết quả của 20 đội bóng thuộc giải bóng đá ngoại hạng Anh mùa 2023-2024
- Tiếp theo là tạo hàm lấy dữ liệu từ 1 đường dẫn (collect_data(url)), trong hàm này ta khởi tạo 1 trình duyệt web chrome và mở đường dẫn url đã cho.
- Sau đấy ta bắt đầu thu thập dữ liệu từ trình duyệt chrome đã mở trên, ta sẽ lưu tất cả các chỉ số được yêu cầu từ các bảng vào 1 list header và dữ liệu của các cầu thủ được lưu vào 1 list là data_rows. Để truy nhập vào lấy dữ liệu của bảng ta truy vấn thông qua ID bảng đó.
- Ở bảng đầu tiên là Standard stats ta sẽ lấy dữ liệu của tất cả các cầu thủ thi đấu trên 90 phút, ta lấy dữ liệu header cho vào list header. Ta

còn tạo thêm 1 dict có key là tên cầu thủ và value là các dòng tương ứng với cầu thủ đó trên bảng.

- Ở các bảng tiếp theo ta sẽ lấy header của bảng đó nối vào list chứa header tổng. Còn dữ liệu của bảng sẽ được lưu thành từng dòng vào được lưu vào 1 list. Sau đó ta kiểm tra xem cầu thủ trong dict được tạo ở bước trên có xuất hiện trong bảng hiện tại không, nếu có thì ta sẽ nối dữ liệu của cầu thủ đó với dữ liệu mới, còn nếu không xuất hiện thì sẽ nối dữ liệu của bảng đó với chuỗi 'N/a'. Lặp lại bước này cho đến khi hết tất cả các bảng.
- Sau khi lặp hết tất cả các bảng, ta thu được dữ liệu của các cầu thủ là các dòng dài chứa các chỉ số tương ứng với header đã lọc ra.
- Đấy mới chỉ là dữ liệu của 1 đội bóng vì vậy ta phải lặp lại việc với tất cả đội bóng và nối dữ liệu của các cầu thủ các đội bóng với nhau.
- Sau khi gom được dữ liệu của từng cầu thủ vào 1 danh sách với mỗi phần tử của danh sách là 1 danh sách chứa các chỉ số tương ứng của cầu thủ đó, ta cần sắp xếp lại danh sách đó theo tên của các cầu thủ.
- Sau khi sắp xếp là lấy danh sách trên tạo 1 DataFrame với header là danh sách các header mà ta đã gom từ các bảng.
- Cuối cùng là xuất file csv theo yêu cầu.

III. Kết quả của bài toán

```
import pandas as pd
df = pd.read_csv('D:\\pythonptit\\results.csv')
df
```

	Player	Nation	Team	Pos	Age	MP-Playing_Time-standard_stats	Starts-Playing_Time-standard_stats	Min-Playing_Time-standard_stats	90s-Playing_Time-standard_stats	Gls-Performace-standard_stats	...	Crs-Performance-Miscellaneous_Stats	Int-Perform Miscellaneous
0	Aaron Cresswell	engENG	West-Ham-United	DF,FW	33	11	4	436	4.8	0	...	11	
1	Aaron Hickey	sctSCO	Brentford	DF	21	9	9	713	7.9	0	...	5	
2	Aaron Ramsdale	engENG	Arsenal	GK	25	6	6	540	6.0	0	...	0	
3	Aaron Ramsey	engENG	Burnley	MF,FW	20	14	5	527	5.9	0	...	2	
4	Aaron Wan-Bissaka	engENG	Manchester-United	DF	25	22	20	1,780	19.8	0	...	14	
...	...	...	...	...	...	...	...	...	...	...	...	...	
488	Yves Bissouma	mlMLI	Tottenham-Hotspur	MF	26	28	26	2,068	23.0	0	...	1	
489	Zeki Amdouni	chSUI	Burnley	FW	22	34	27	1,953	21.7	5	...	8	
490	Álex Moreno	esESP	Aston-Villa	DF	30	21	11	1,031	11.5	2	...	37	
491	Đorđe Petrović	rsSRB	Chelsea	GK	23	23	22	1,987	22.1	0	...	0	
492	Łukasz Fabiański	plPOL	West-Ham-United	GK	38	10	7	721	8.0	0	...	0	

493 rows × 189 columns

Câu 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
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 thu được là:

```
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.

- Sau khi ép kiểu các dữ liệu về mong muốn, ta sẽ tính toán thống kê cho các đội và tất cả cầu thủ.

- 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	29.0	27.238095	7.993152	18.0	19.904762	10.084170	1593.0	1773.714286	834.245056	1

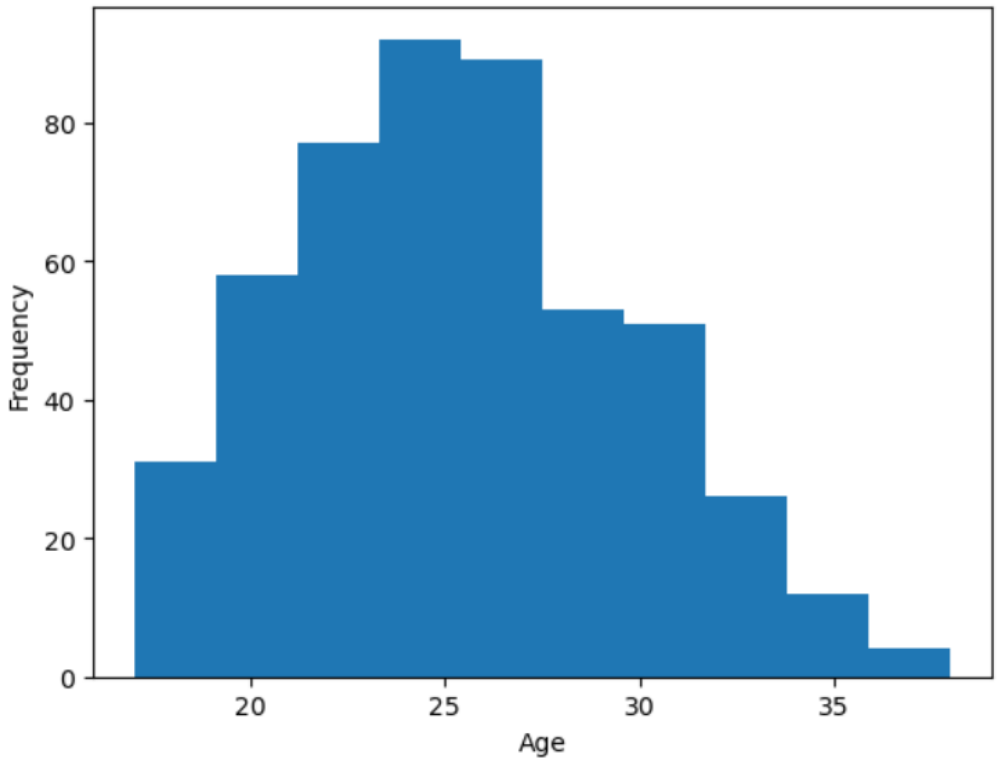
3. Vẽ historgram phân bố của mỗi chỉ số của các cầu thủ trong toàn giải và mỗi đội.

- 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**

Câu 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.

- Đầu tiên ta cũng phải lọc lại dữ liệu như các câu trên

```
# 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)
```

- Tiếp theo là 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)
```

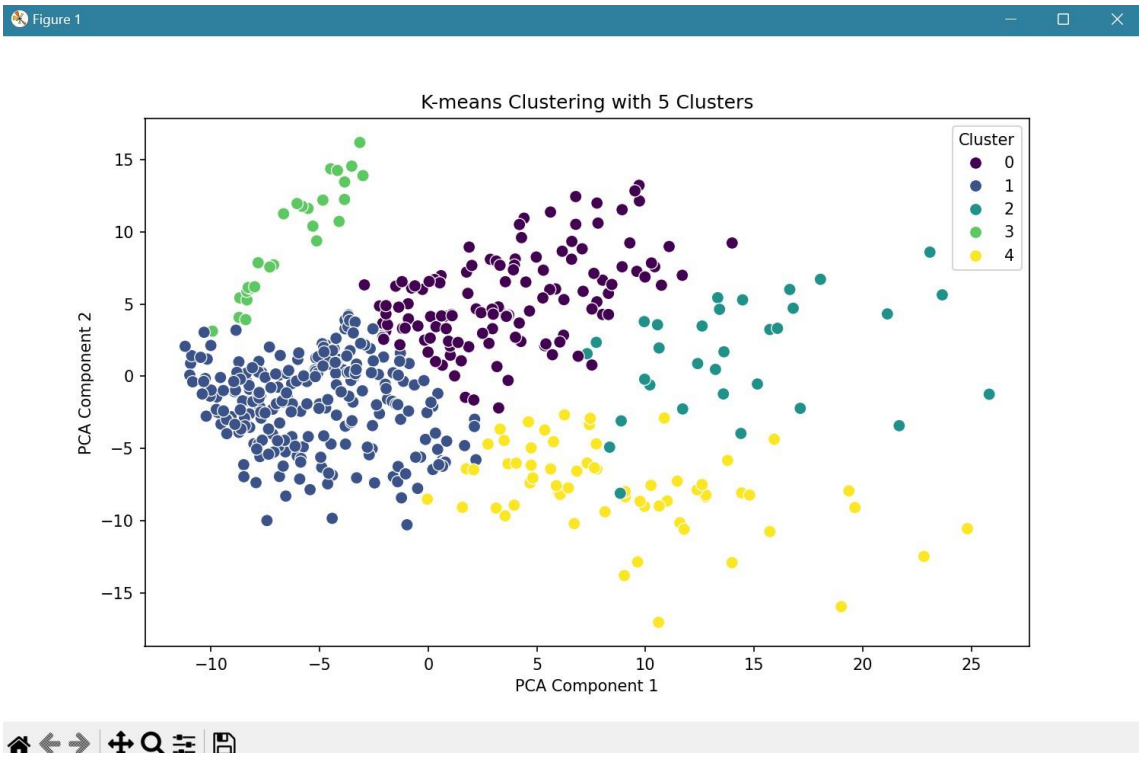
- Phân nhóm và gán nhãn cho các cầu thủ

```
# 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)
```

- Cuối cùng là 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 của phần này sẽ là:



Theo tôi thì 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)
```

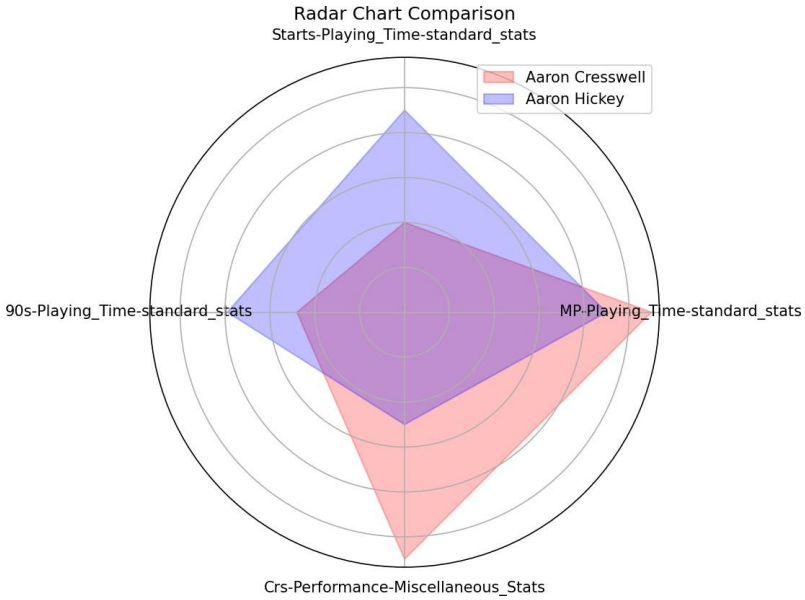
- Khi muốn vẽ biểu đồ rada so sánh giữa cầu thủ 1 và cầu thủ 2 ta chỉ cần gõ lệnh có dạng như sau ở terminal:

```
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 là:



Câu 4:

I. 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.

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

II. Xử lý và giải quyết bài toán

1. Đầu tiên ta sẽ lấy dữ liệu trên trang web về và lưu vào file 'results3.csv'.

- Bước này tương tự như câu 1, nhưng thay vì sử dụng selenium ta sử dụng requests. Ta lấy dữ liệu theo từng đội bằng cách tìm kiếm theo các thẻ chứ dữ liệu cần được biết ở Elements của web. Trong từng đội ta lại lấy dữ liệu về cầu thủ.

- Khi đó dữ liệu lấy về sẽ gồm têm các cầu, đội tham gia và giá chuyển nhượng mùa giải 2023-2024 của ngoại hạng Anh.

- Kết quả khi đó sẽ là:

	Player	Team	Cost
0	Ederson	Man City	€43.2M
1	Scott Carson	Man City	€0.1M
2	Stefan Ortega	Man City	€1.8M
3	Josh Wilson-Esbrand	Man City	€4.5M
4	Kyle Walker	Man City	€18.8M
...	...	...	...

2. Đề xuất phương pháp: Dựa trên dữ liệu đã thu thập, sinh viên cần đề xuất một phương pháp để 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.