HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP LỚN

Lập trình với Python

Giảng viên : **Kim Ngọc Bách**
Nhóm : **11**
Tên sinh viên : **Nguyễn Năng Vương**
Mã sinh viên : **B22DCCN923**

Bài 1:

Cấu trúc chương trình:

1. *Đọc HTML của trang web bằng requests và dùng BeautifulSoup để đọc nội dung html:*

```
# URL của trang web
url = 'https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats'

# Gửi yêu cầu và lấy nội dung trang web
r = requests.get(url)
soup = bs(r.content, 'html.parser')
```

2. *Dữ liệu cầu thủ cần lấy có nhiều nhóm chỉ số khác nhau nên cần lấy ra thẻ chứa các link dẫn tới bảng chứa các nhóm chỉ số cần tìm:*

```
# Tìm tất cả các mục <li> trong <ul> sau <p class="listhead">
data = []
for item in soup.find('p', class_='listhead').find_next('ul').find_all('li'):
    title = item.text.strip()  # Lấy tên mục
    link = 'https://fbref.com' + item.find('a')['href']  # Lấy đường dẫn đầy đủ
    data.append({'title': title, 'link': link})  # Lưu vào danh sách dưới dạng từ điển
```

- Kết quả chương trình:

```
Standard Stats: https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats
Goalkeeping: https://fbref.com/en/comps/9/2023-2024/keepers/2023-2024-Premier-League-Stats
Advanced Goalkeeping: https://fbref.com/en/comps/9/2023-2024/keepersadv/2023-2024-Premier-League-Stats
Shooting: https://fbref.com/en/comps/9/2023-2024/shooting/2023-2024-Premier-League-Stats
Passing: https://fbref.com/en/comps/9/2023-2024/passing/2023-2024-Premier-League-Stats
Pass Types: https://fbref.com/en/comps/9/2023-2024/passing_types/2023-2024-Premier-League-Stats
Goal and Shot Creation: https://fbref.com/en/comps/9/2023-2024/gca/2023-2024-Premier-League-Stats
Defensive Actions: https://fbref.com/en/comps/9/2023-2024/defense/2023-2024-Premier-League-Stats
Possession: https://fbref.com/en/comps/9/2023-2024/possession/2023-2024-Premier-League-Stats
Playing Time: https://fbref.com/en/comps/9/2023-2024/playingtime/2023-2024-Premier-League-Stats
Miscellaneous Stats: https://fbref.com/en/comps/9/2023-2024/misc/2023-2024-Premier-League-Stats
```

3. *Vì bảng chứa dữ liệu các cầu thủ bị ẩn trong Comment của HTML nên cần lấy ra Comment trong các link:*

```
def get_url(url):
    # Gửi yêu cầu đến URL và tạo BeautifulSoup từ nội dung HTML
    r = requests.get(url)
    soup = bs(r.content, 'html.parser')
    time.sleep(0.5)
    print('Loading...')
    # Tìm tất cả các comment trong nội dung HTML
    comments = soup.find_all(string=lambda text: isinstance(text, Comment))
    return comments
```

- Vì việc đọc dữ liệu từ trang web quá nhanh có thể làm cho máy chủ chặn nên có thể thêm thời gian chờ.

4. *Để đọc được các chỉ số trong mỗi link cần 1 hàm để phân tích html:*

```
 30 ⊞ def get_goalkeeper_stats(comments):…
105 ⊞ def get_shooting_stats(comments):…
187 ⊞ def get_standard_stats(comments):…
283 ⊞ def get_passing_stats(comments):…
379 ⊞ def get_passing_types_stats(comments):…
548 ⊞ def get_gca_stats(comments):…
636 ⊞ def get_defense_stats(comments):…
723 ⊞ def get_possession_stats(comments):…
821 ⊞ def get_playing_time_stats(comments):…
909 ⊞ def get_misc_stats(comments):…
```

- Vì mỗi nhóm chỉ số được lưu trong các bảng khác nhau nên cần viết các hàm riêng cho từng link, mỗi hàm trả về 1 dataFrame chứa các nhóm chỉ số.

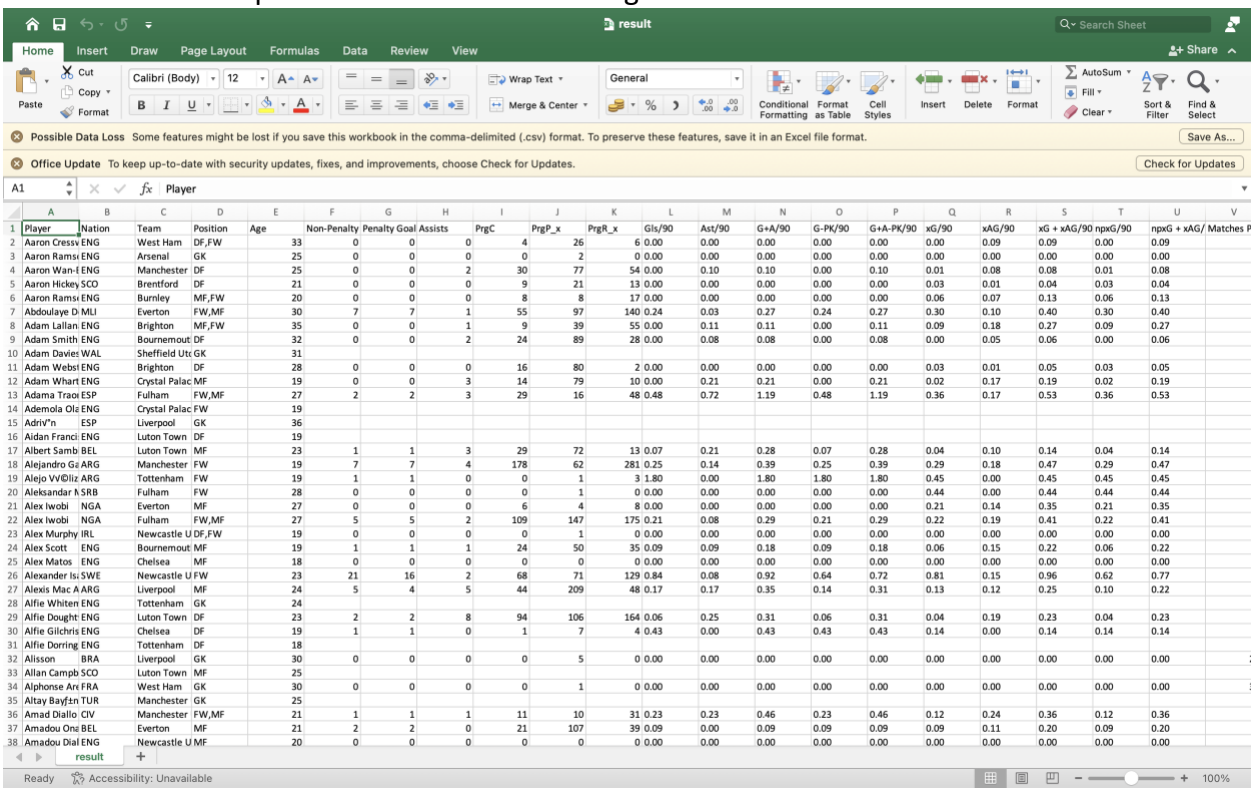
5. *Hợp nhất các dataFrame trong các link tìm được và sắp xếp:*

```
992  dataframes = [
993      get_standard_stats(get_url(data[0]['link'])),
994      get_goalkeeper_stats(get_url(data[1]['link'])),
995      get_shooting_stats(get_url(data[3]['link'])),
996      get_passing_stats(get_url(data[4]['link'])),
997      get_passing_types_stats(get_url(data[5]['link'])),
998      get_gca_stats(get_url(data[6]['link'])),
999      get_defense_stats(get_url(data[7]['link'])),
1000     get_possession_stats(get_url(data[8]['link'])),
1001     get_playing_time_stats(get_url(data[9]['link'])),
1002     get_misc_stats(get_url(data[10]['link']))
1003 ]
1004
1005 # Khởi tạo DataFrame kết quả với DataFrame đầu tiên
1006 df = dataframes[0]
1007
1008 # Gộp tất cả các DataFrame vào df_combined
1009 for df_clone in dataframes[1:]:
1010     df = pd.merge(df, df_clone, on=['Player', 'Nation', 'Team', 'Position', 'Age'], how='outer')
1011 df['First Name'] = df['Player'].apply(lambda x: x.split()[0])  # Lấy tên đầu tiên
1012
1013 df = df.sort_values(by=['First Name', 'Age'], ascending=[True, False])
1014 df = df.drop(columns=['First Name'])
```

6. *Lưu vào file result.csv:*

```
1015 # Kiểm tra kết quả
1016 df.to_csv('/Users/nangvuong/Documents/CODE PTIT/Python/result.csv', sep=';', index=False)
```

- Kết quả khi mở file result.csv bằng excel:



Player	Nation	Team	Position	Age	Non-Penalty	Penalty Goal	Assists	PrgC	PrgP_x	PrgR_x	Gls/90	Ast/90	G+A/90	G-PK/90	G+A-PK/90	xG/90	xAG/90	xG + xAG/90	npxG/90	npxG + xAG/	Matches P
Aaron Cressv	ENG	West Ham	DF,FW	33	0	0	0	4	26	6	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.09	0.00	0.09	
Aaron Rams	ENG	Arsenal	GK	25	0	0	0	0	2	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Aaron Wan-I	ENG	Manchester	DF	25	0	0	2	30	77	54	0.00	0.10	0.10	0.00	0.10	0.01	0.08	0.08	0.01	0.08	
Aaron Hickey	SCO	Brentford	DF	21	0	0	0	9	21	13	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.04	0.03	0.04	
Aaron Rams	ENG	Burnley	MF,FW	20	0	0	0	8	8	17	0.00	0.00	0.00	0.00	0.00	0.06	0.07	0.13	0.06	0.13	
Abdoulaye D	MLI	Everton	FW,MF	30	7	7	1	55	97	140	0.24	0.03	0.27	0.24	0.27	0.30	0.10	0.40	0.30	0.40	
Adam Lallan	ENG	Brighton	MF,FW	35	0	0	1	9	39	55	0.00	0.11	0.11	0.00	0.11	0.09	0.18	0.27	0.09	0.27	
Adam Smith	ENG	Bournemout	DF	32	0	0	2	24	89	28	0.00	0.08	0.08	0.00	0.08	0.00	0.05	0.06	0.00	0.06	
Adam Davies	WAL	Sheffield Utc	GK	31																	
Adam Webst	ENG	Brighton	DF	28	0	0	0	16	80	2	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.05	0.03	0.05	
Adam Whart	ENG	Crystal Palac	MF	19	0	0	3	14	79	10	0.00	0.21	0.21	0.00	0.21	0.02	0.17	0.19	0.02	0.19	
Adama Traor	ESP	Fulham	FW,MF	27	2	2	3	29	16	48	0.48	0.72	1.19	0.48	1.19	0.36	0.17	0.53	0.36	0.53	
Ademola Ola	ENG	Crystal Palac	FW	19																	
Adriv°n	ESP	Liverpool	GK	36																	
Aidan Franci	ENG	Luton Town	DF	19																	
Albert Samb	BEL	Luton Town	MF	23	1	1	3	29	72	13	0.07	0.21	0.28	0.07	0.28	0.04	0.10	0.14	0.04	0.14	
Alejandro Ga	ARG	Manchester	FW	19	7	7	4	178	62	281	0.25	0.14	0.39	0.25	0.39	0.29	0.18	0.47	0.29	0.47	
Alejo VvÐliz	ARG	Tottenham	FW	19	1	1	0	0	1	3	1.80	0.00	1.80	1.80	1.80	0.45	0.00	0.45	0.45	0.45	
Aleksandar N	SRB	Fulham	FW	28	0	0	0	0	1	0	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.44	0.44	0.44	
Alex Iwobi	NGA	Everton	MF	27	0	0	0	6	4	8	0.00	0.00	0.00	0.00	0.00	0.21	0.14	0.35	0.21	0.35	
Alex Iwobi	NGA	Fulham	FW,MF	27	5	5	2	109	147	175	0.21	0.08	0.29	0.21	0.29	0.22	0.19	0.41	0.22	0.41	
Alex Murphy	IRL	Newcastle U	DF,FW	19	0	0	0	0	1	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Alex Scott	ENG	Bournemout	MF	19	1	1	1	24	50	35	0.09	0.09	0.18	0.09	0.18	0.06	0.15	0.22	0.06	0.22	
Alex Matos	ENG	Chelsea	MF	18	0	0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Alexander Is	SWE	Newcastle U	FW	23	21	16	2	68	71	129	0.84	0.08	0.92	0.64	0.72	0.81	0.15	0.96	0.62	0.77	
Alexis Mac A	ARG	Liverpool	MF	24	5	4	5	44	209	48	0.17	0.17	0.35	0.14	0.31	0.13	0.12	0.25	0.10	0.22	
Alfie Whiten	ENG	Tottenham	GK	24																	
Alfie Dought	ENG	Luton Town	DF	23	2	2	8	94	106	164	0.06	0.25	0.31	0.06	0.31	0.04	0.19	0.23	0.04	0.23	
Alfie Gilchris	ENG	Chelsea	DF	19	1	1	0	1	7	4	0.43	0.00	0.43	0.43	0.43	0.14	0.00	0.14	0.14	0.14	
Alfie Dorring	ENG	Tottenham	DF	18																	
Alisson	BRA	Liverpool	GK	30	0	0	0	0	5	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Allan Campb	SCO	Luton Town	MF	25																	
Alphonse Arr	FRA	West Ham	GK	30	0	0	0	0	1	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
Altay Bayfzn	TUR	Manchester	GK	25																	
Amad Diallo	CIV	Manchester	FW,MF	21	1	1	1	11	10	31	0.23	0.23	0.46	0.23	0.46	0.12	0.24	0.36	0.12	0.36	
Amadou Ona	BEL	Everton	MF	21	2	2	0	21	107	39	0.09	0.00	0.09	0.09	0.09	0.09	0.11	0.20	0.09	0.20	
Amadou Dial	ENG	Newcastle U	MF	20	0	0	0	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

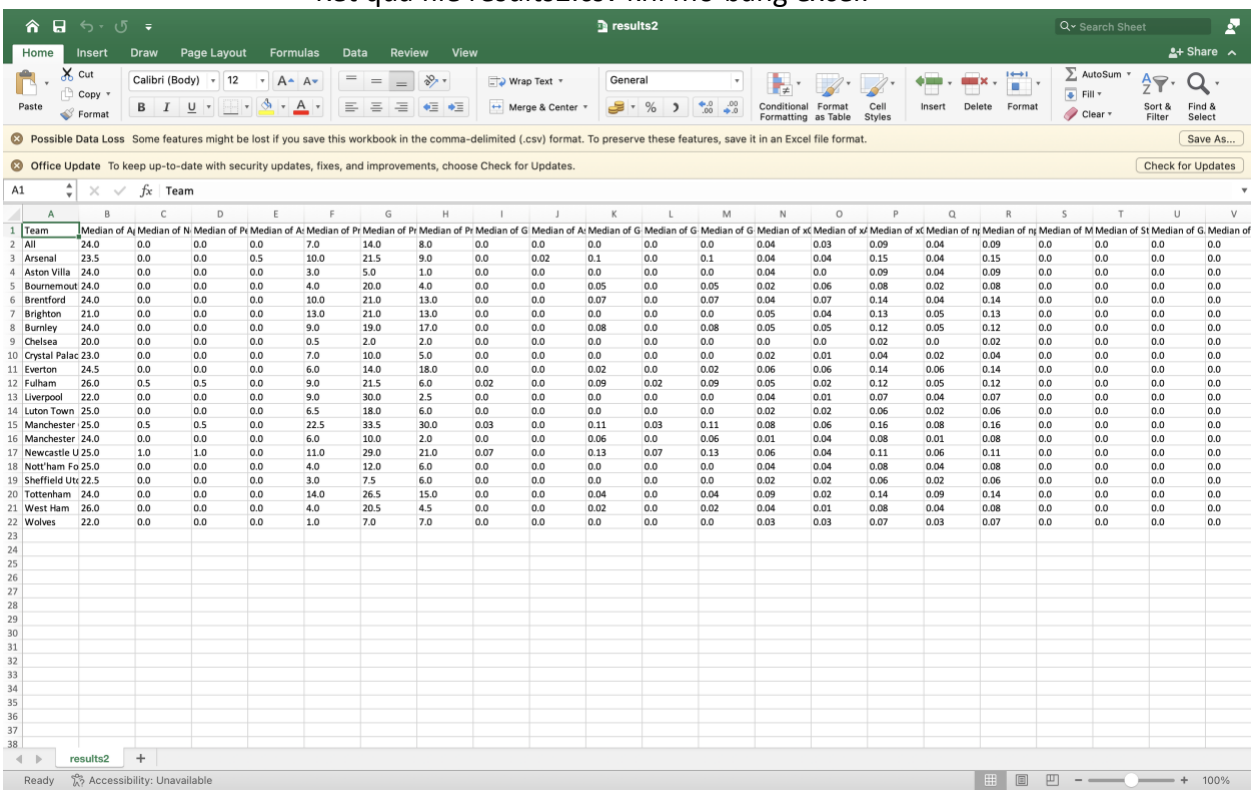
Bài 2:

1. *Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số.*
    a. Lấy dữ liệu các cầu thủ từ file result.csv và đổi các dữ liệu có dạng NaN thành 0.
    b. Dùng phương thức nlargest, smallest có sẵn trong thư viện pandas để tìm ra top 3 cao nhất và thấp nhất ở mỗi chỉ số.
    c. In ra màn hình.

Kết quả:

```
Top 3 cầu thủ có Non-Penalty Goals cao nhất:
             Player  Non-Penalty Goals
221  Erling Haaland                27.0
148     Cole Palmer                22.0
24   Alexander Isak                21.0

Top 3 cầu thủ có Non-Penalty Goals thấp nhất:
              Player  Non-Penalty Goals
0    Aaron Cresswell                 0.0
1     Aaron Ramsdale                 0.0
2  Aaron Wan-Bissaka                 0.0
```

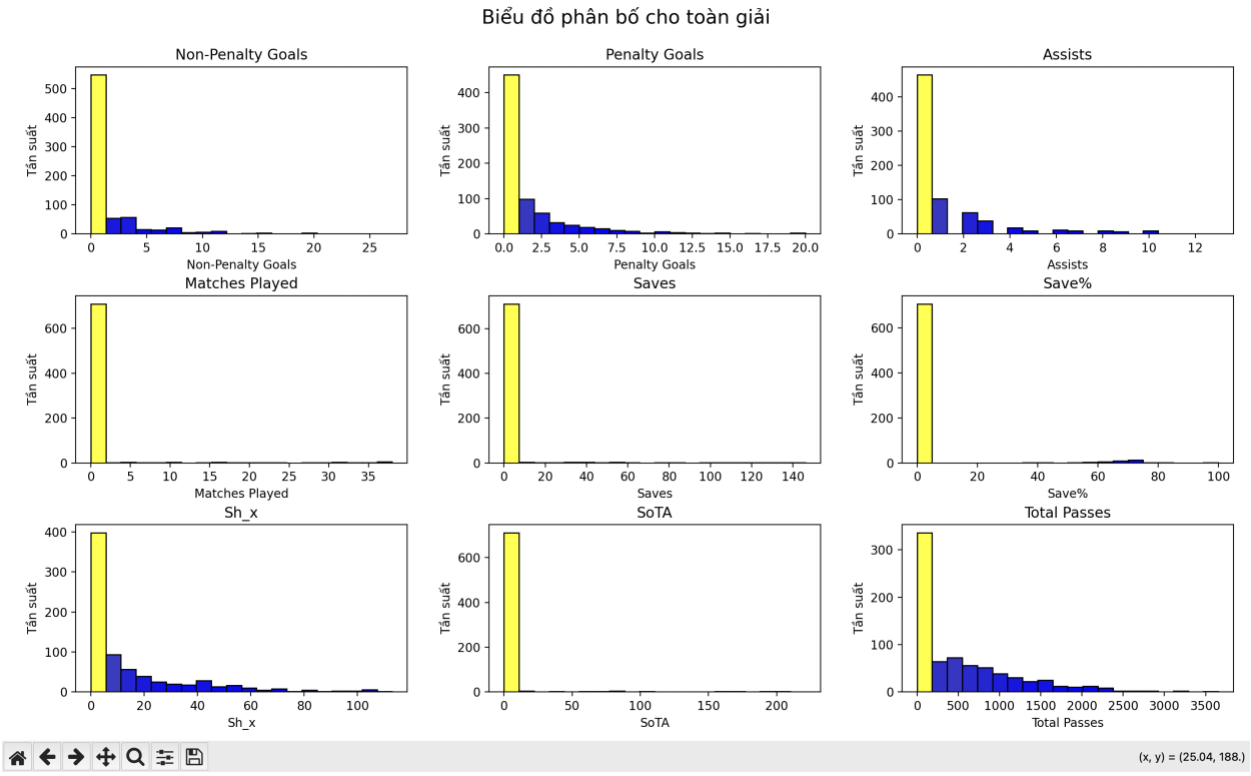
2. *Tìm trung vị, trung bình, độ lệch chuẩn của mỗi chỉ số cho toàn giải và cho mỗi đội:*
    a. Lấy dữ liệu các cầu thủ từ file result.csv và đổi các dữ liệu có dạng NaN thành 0.
    b. Dùng phương thức median, mean, std trong thư viện pandas để tìm trung vị, trung bình, độ lệch chuẩn cho từng chỉ số và dùng hàm round để làm tròn đến 2 chữ số sau dấu phẩy.
    c. Lưu kết quả vào file results2.csv
        - Kết quả file results2.csv khi mở bằng excel:



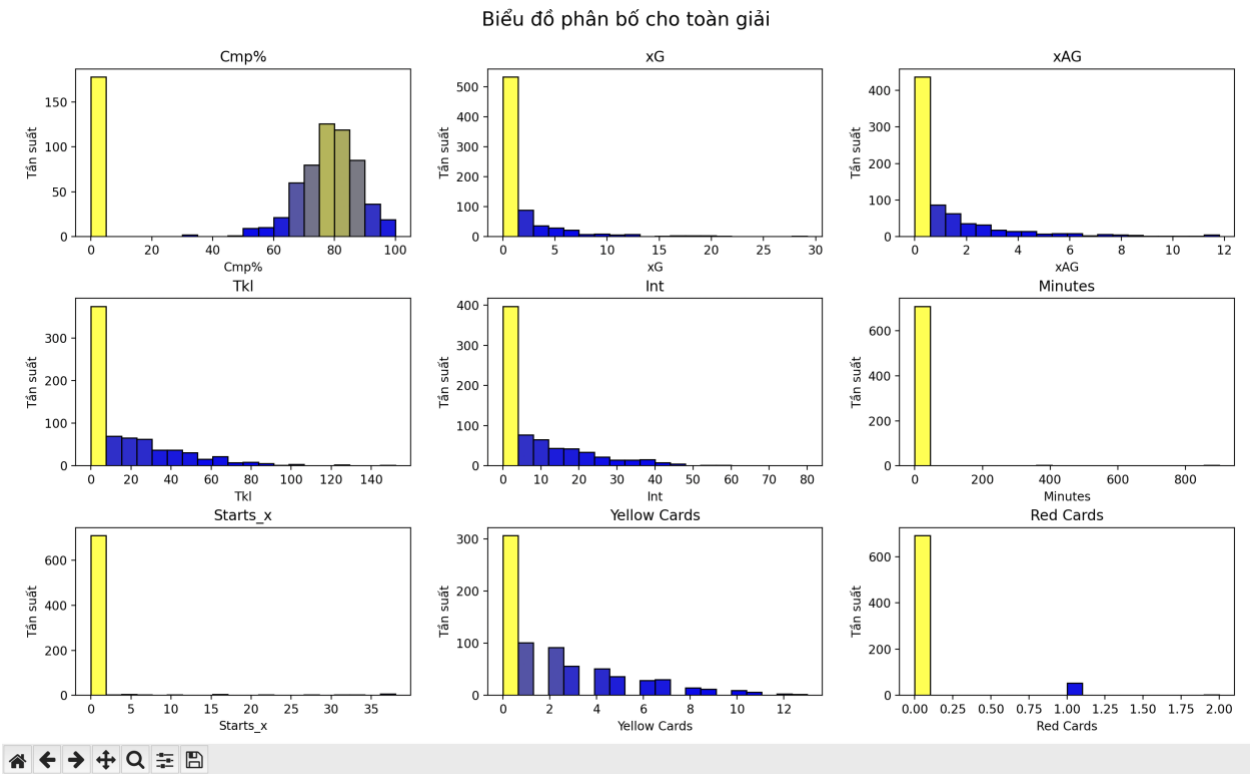
3. *Vẽ historgram phân bố của mỗi chỉ số của các cầu thủ trong toàn giải và mỗi đội:*
    a. Đọc dữ liệu cầu thủ từ file result.csv và đổi các dữ liệu NaN thành 0.
    b. Vì chỉ số của các cầu thủ rất nhiều nên chỉ lấy những giá trị đại diện cho những nhóm chỉ số.

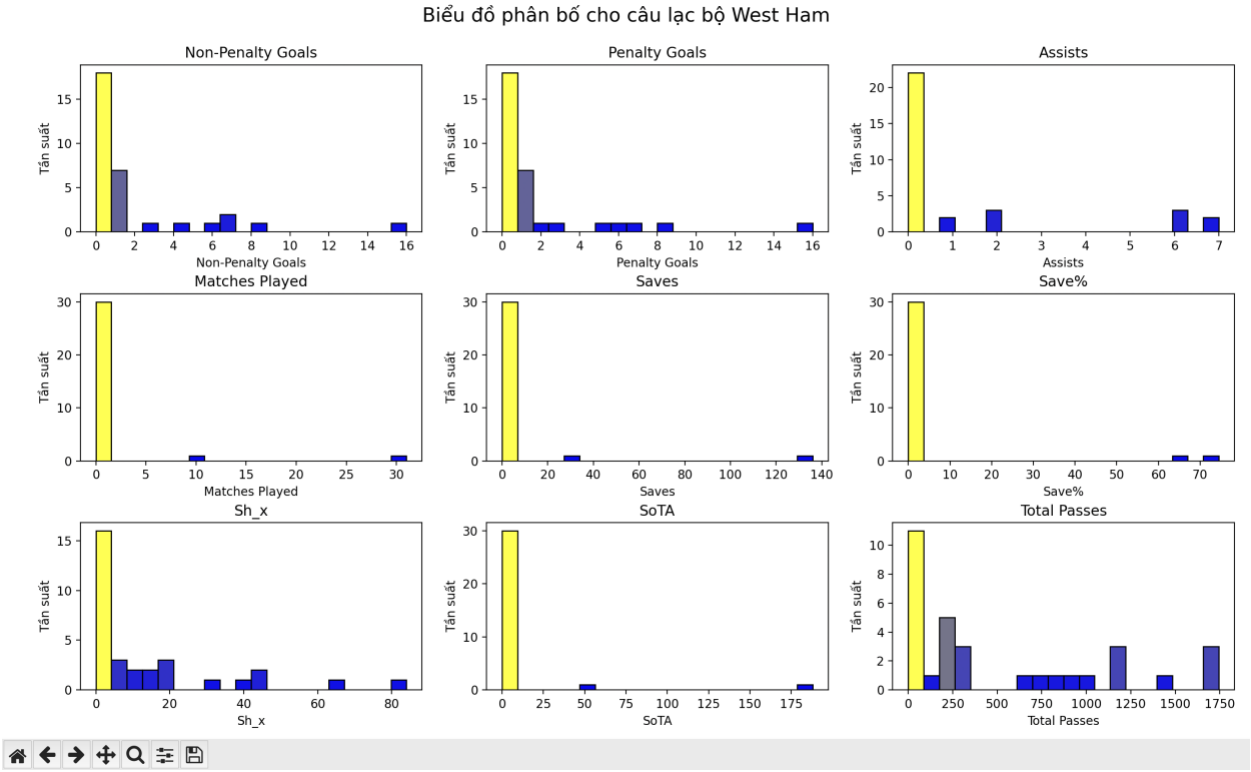
```
columns_to_plot = [
    'Non-Penalty Goals', 'Penalty Goals', 'Assists', 'Matches Played',
    'Saves', 'Save%', 'Sh_x', 'SoTA',
    'Total Passes', 'Cmp%', 'xG', 'xAG',
    'Tkl', 'Int', 'Minutes', 'Starts_x',
    'Yellow Cards', 'Red Cards'
]
```

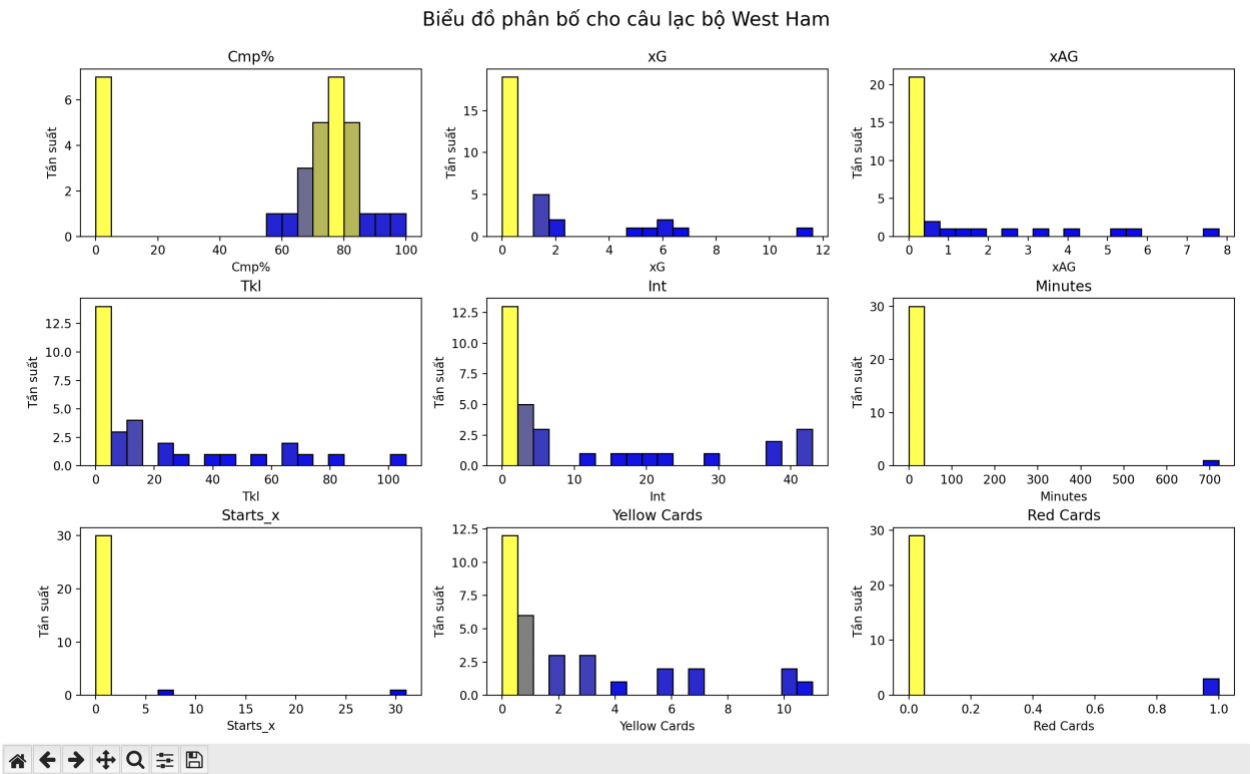
    c. Dùng thư viện matplotlib để vẽ histograms.
    d. Kết quả:


Biểu đồ phân bố cho toàn giải
Non-Penalty Goals
Penalty Goals
Assists
Matches Played
Saves
Save%
Sh_x
SoTA
Total Passes
Tần suất

(x, y) = (25.04, 188.)


Biểu đồ phân bố cho toàn giải
Cmp%
xG
xAG
Tkl
Int
Minutes
Starts_x
Yellow Cards
Red Cards
Tần suất





4. *Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số.*
    a. Lấy dữ liệu được lưu trong file results2.csv.
    b. Sử dụng phương thức max() để tìm đội có trung bình chỉ số cao nhất.

c. In kết quả ra màn hình.

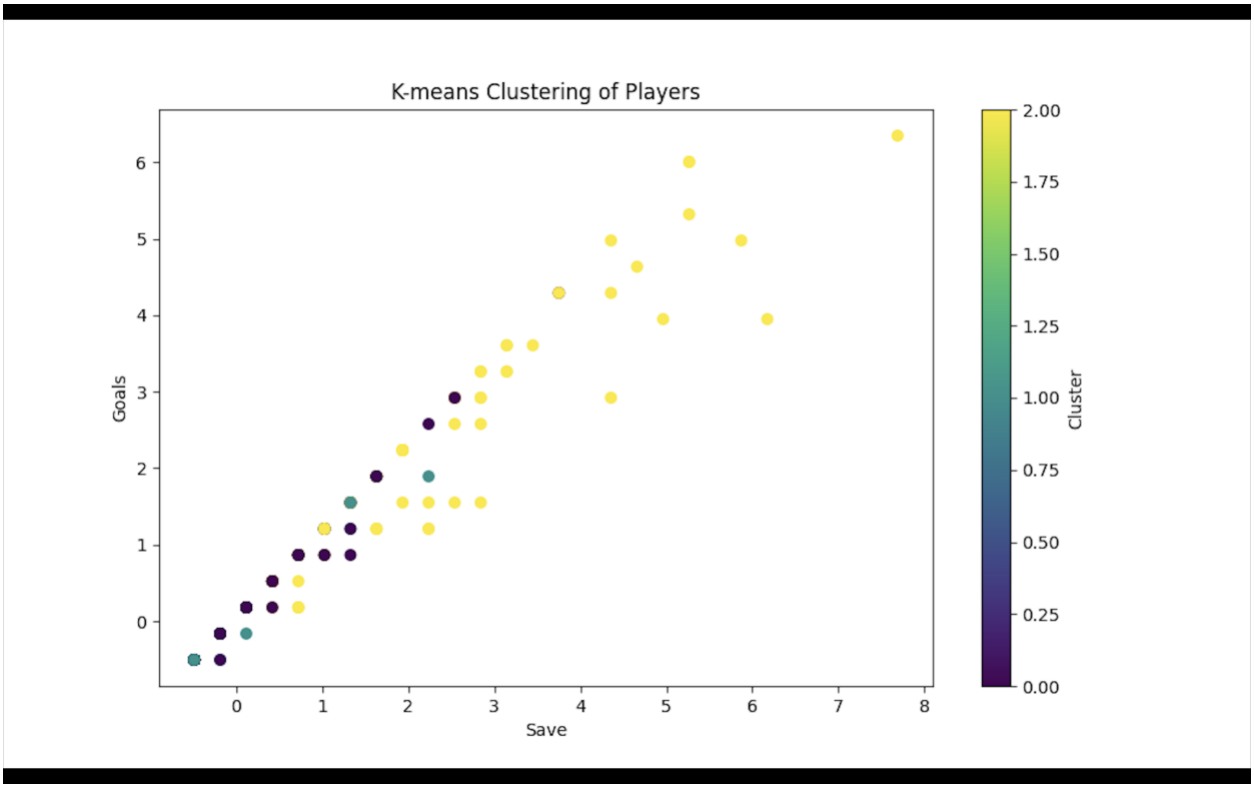
```
Yellow Cards: Wolves
Red Cards: Burnley
Yellow-Red Cards: Burnley
Fls: Everton
Fld: Tottenham
Off_y: Everton
Crs_y: Everton
Won: Everton
Lost_y: Burnley
Won%: Nott'ham Forest
Recov: Everton

Đội bóng có phong độ tốt nhất giải Ngoại Hạng Anh mùa 2023-2024:
Manchester City với 80 chỉ số cao nhất.
```

⇨ Vậy đội có phong độ tốt nhất giải là Manchester City.

Bài 3:

1. *Sử dụng thuật toán K-means để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau.*
    a. Lấy dữ liệu cầu thủ từ file result.csv, đổi các dữ liệu NaN thành 0
    b. Sử dụng thư viện có sẵn Sklearn để phân loại các cầu thủ bằng thuật toán Kmeans, sử dụng thư viện matplotlib để vẽ hình dữ liệu trên mặt 2D.
    c. Chọn ra 2 chỉ số của cầu thủ để phân loại so sánh.

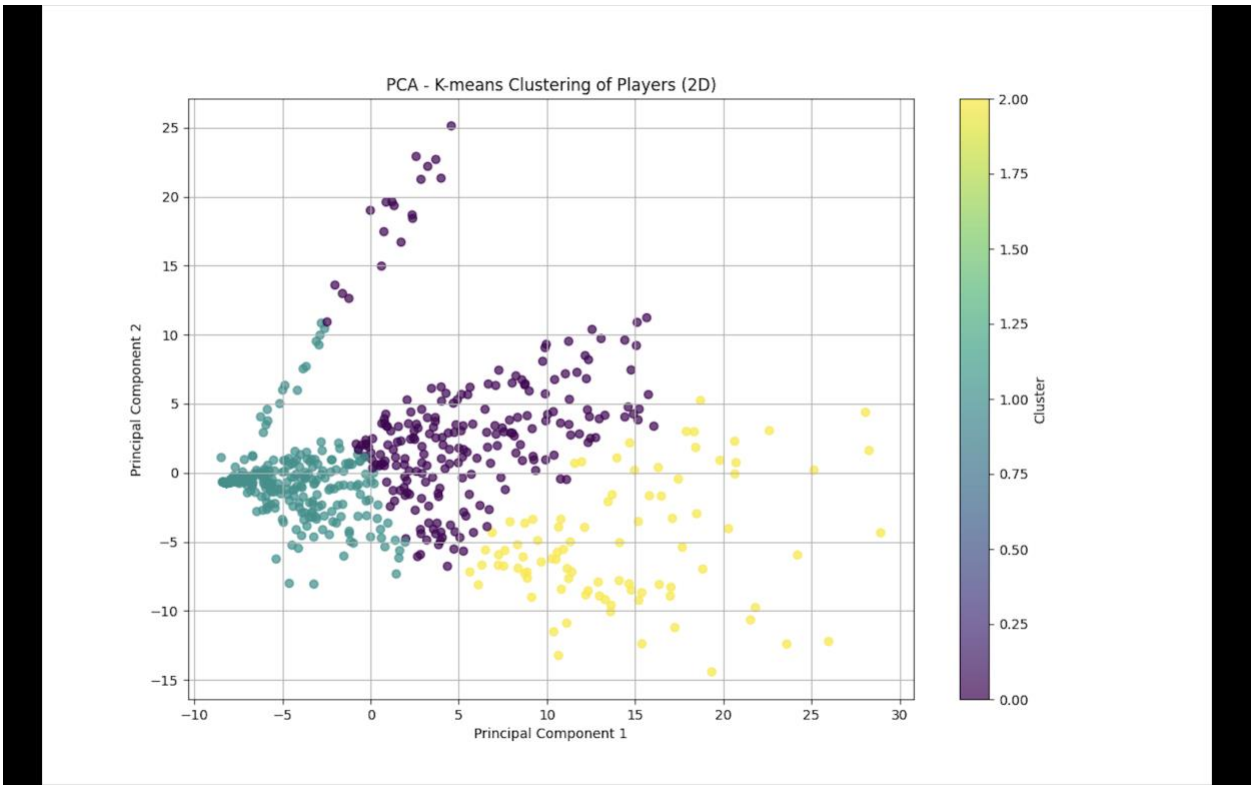
Kết quả với 2 chỉ số là Goals, Save:



2. *Theo em, nên phân loại các cầu thủ thành 4 nhóm. Vì có 4 vị trí trên sân cỏ là thủ môn, hậu vệ, tiền vệ, tiền đạo tương ứng với các chỉ số sẽ phân bố khác nhau với từng vị trí.*

3. *Sử dụng thuật toán PCA, giảm số chiều dữ liệu xuống 2 chiều, vẽ hình phân cụm các điểm dữ liệu trên mặt 2D.*
    a. Đọc dữ liệu các cầu thủ từ file result.csv, chuyển các dữ liệu NaN thành 0.
    b. Sử dụng thư viện Sklearn để phân loại các cầu thủ bằng thuật toán PCA và Kmeans.
    c. Sử dụng thuật toán PCA để giảm chiều dữ liệu các chỉ số cầu thủ thành 2.
    d. Biểu diễn các dữ liệu cầu thủ trên mặt 2D và chọn số cụm là 4.

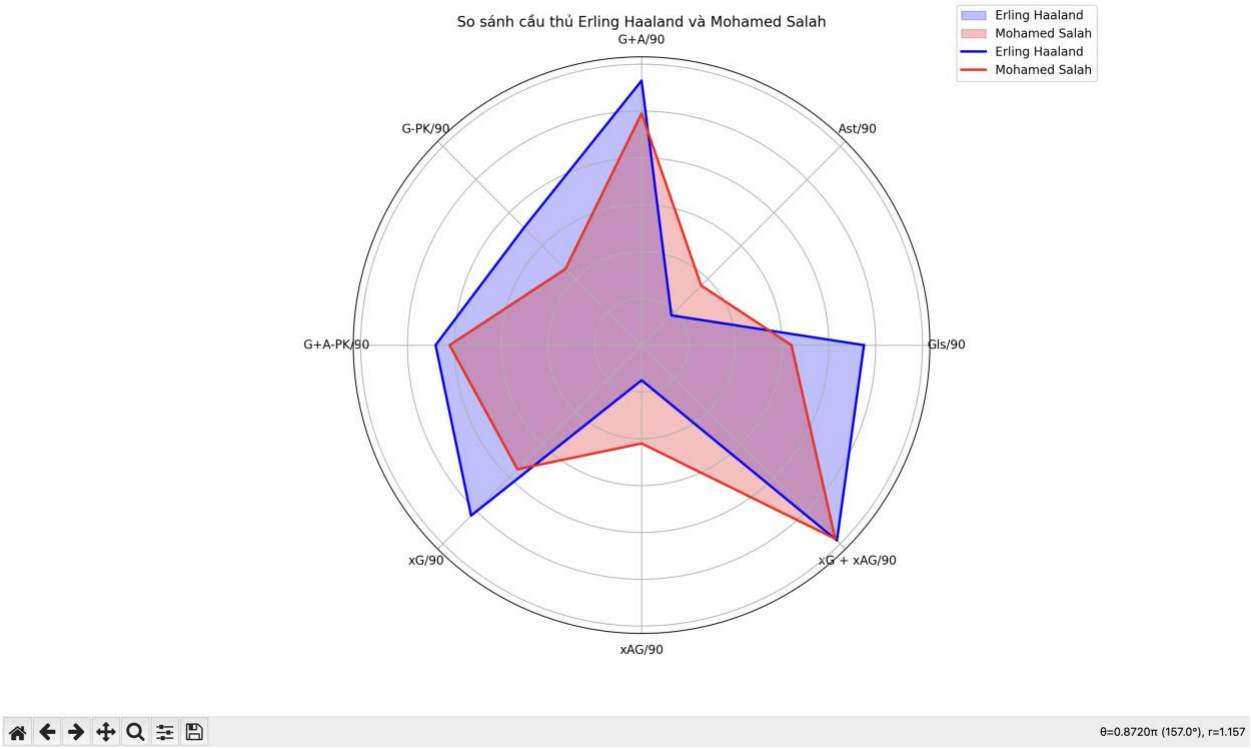
Kết quả:



4. *Viết chương trình python vẽ biểu đồ rada (radar chart) so sánh cầu thủ*
    a. Lấy dữ liệu cầu thủ từ file result.csv để so sánh, đổi các dữ liệu NaN thành 0.
    b. Dùng thư viện matplotlib để vẽ radar chart.
    c. Viết hàm chính với thư viện argparse để nhận input và xử lý dữ liệu.

Kết quả khi so sánh cầu thủ Erling Haaland và Mohamed Salah với các chỉ số Gls/90, Ast/90, G+A/90, G-PK/90, G+A-PK/90, xG/90, xAG/90, xG + xAG/90 bằng câu lệnh:

```
python3 "/Users/nangvuong/Documents/CODE PTIT/Python/BÀI3(rada chart).py" --p1 "``Erling Haaland``" --p2 "Mohamed Salah" --Attribute "Gls/90;Ast/90;G+A/90;G-PK/90;G+A-PK/90;xG/90;xAG/90;xG + xAG/90"
```



Bài 4: ***Lấy dữ liệu chuyển nhượng các cầu thủ từ trang https://www.footballtransfers.com.***

1. *Vì trang https://www.footballtransfers.com sử dụng JavaScript để tải dữ liệu cầu thủ khi truy cập, vì vậy sử dụng `requests.get` sẽ chỉ lấy html tĩnh, không có nội dung tải động như dữ liệu cầu thủ. Để lấy dữ liệu cầu thủ trên, em sử dụng thư viện **Selenium** để giả lập trình duyệt và tải hết nội dung.*

```
6   def read_url(url):
7       driver = webdriver.Safari()
8       driver.get(url)
9       time.sleep(5)  # Đợi trang tải xong
10      page_source = driver.page_source
11      driver.quit()
12      soup = bs(page_source, 'html.parser')
13      return soup
```

Có thể thay bằng webdriver của các trình duyệt khác như Chorme,…

2. *Url của chứa dữ liệu các cầu thủ là https://www.footballtransfers.com/en/values/players/most-valuable-players/playing-in-uk-premier-league và các page của url này.*

```python
# Đọc dữ liệu từ trang 1
url_page_1 = 'https://www.footballtransfers.com/en/values/players/most-valuable-players/playing-in-uk-premier-league'
soup_page_1 = read_url(url_page_1)
players_data_page_1 = extract_player_data(soup_page_1)
all_players_data = pd.concat([all_players_data, pd.DataFrame(players_data_page_1)], ignore_index=True)

# Đọc dữ liệu từ trang 2 đến trang 24
for page in range(2, 25):
    url = f'https://www.footballtransfers.com/en/values/players/most-valuable-players/playing-in-uk-premier-league/{page}'
    soup = read_url(url)
    players_data = extract_player_data(soup)
    all_players_data = pd.concat([all_players_data, pd.DataFrame(players_data)], ignore_index=True)
    print(f"Data from page {page} added to DataFrame.")
```

3. *Sau đó đọc dữ liệu các cầu thủ trong thẻ tr và lấy dữ liệu cần tìm và lưu và file test.csv.*

Kết quả khi mở file test.csv:

```
Player Name;Team;Transfer Value
Erling Haaland;Man City;€149.3M
Bukayo Saka;Arsenal;€117.5M
Phil Foden;Man City;€116.1M
Rodri;Man City;€100.4M
Kai Havertz;Arsenal;€95M
William Saliba;Arsenal;€79.5M
Martin Ødegaard;Arsenal;€73.2M
Rúben Dias;Man City;€70.5M
Alexander Isak;Newcastle Utd.;€69.8M
Declan Rice;Arsenal;€69.1M
Cole Palmer;Chelsea;€67.3M
Bruno Guimarães;Newcastle Utd.;€65.3M
Josko Gvardiol;Man City;€65M
```

Phương pháp định giá cầu thủ:

⇨ Dựa vào bảng giá chuyển nhượng các cầu thủ, ta có thể định giá các cầu thủ dựa trên những tiêu chí:

- Phân tích hiệu suất cầu thủ
- Yếu tố tuổi tác và tiềm năng phát triển
- Yếu tố hợp đồng và điều khoản chuyển nhượng
- Phân tích thương mại và giá trị truyền thông
- Sử dụng mô hình định giá hiện đại