

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



BÁO CÁO BÀI TẬP LỚN

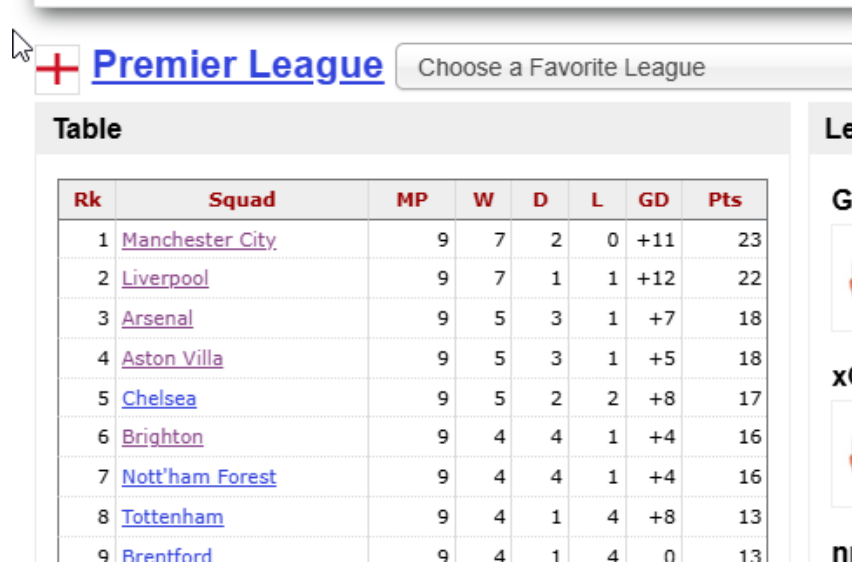
- **Sinh viên thực hiện : Nguyễn Thành Trung**
- **Mã sinh viên : 22DCCN873**

Bài 1

File code : Bai1.py

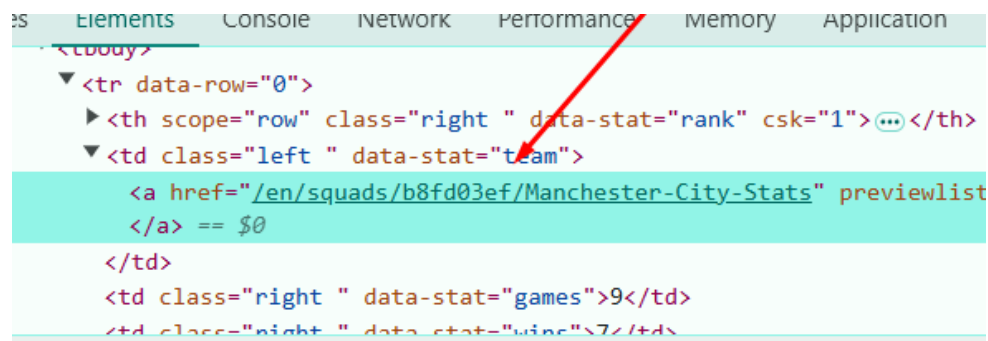
Ý tưởng :

Đầu tiên , ta sẽ truy cập đến bảng bằng id của bảng xếp hạng đó



Rk	Squad	MP	W	D	L	GD	Pts
1	Manchester City	9	7	2	0	+11	23
2	Liverpool	9	7	1	1	+12	22
3	Arsenal	9	5	3	1	+7	18
4	Aston Villa	9	5	3	1	+5	18
5	Chelsea	9	5	2	2	+8	17
6	Brighton	9	4	4	1	+4	16
7	Nott'ham Forest	9	4	4	1	+4	16
8	Tottenham	9	4	1	4	+8	13
9	Brentford	9	4	1	4	0	13

Ta sẽ truy cập đến từng dòng và lấy link của đội bóng



```
<tbody>
  <tr data-row="0">
    <th scope="row" class="right " data-stat="rank" csk="1">...</th>
    <td class="left " data-stat="team">
      <a href="/en/squads/b8fd03ef/Manchester-City-Stats" previewlist
      </a> == $0
    </td>
    <td class="right " data-stat="games">9</td>
    <td class="right " data-stat="wins">7</td>
```

3. Khi truy cập được sang trang của đội bóng , ta sẽ tìm đường link của mùa giải trước (2023-2024) của đội bóng

4. Sau khi đến được trang cần lấy dữ liệu, chúng ta sẽ chọn từng table , bỏ qua những bảng không cần thiết và xử lý dữ liệu theo yêu cầu

```
all_dataframe=[]
for i in team_urls:
    team_url = i
    # lấy tên của đội bóng
    teamName= team_url.split("/")[-1].replace("-Stats","").replace("-"," ")
    print(f"Đang crawl dữ liệu team {teamName}")
    r=requests.get(team_url)
    soup = bs(r.content, 'html.parser')
    #link đội bóng mùa 2023-2024
    previousSeasonUrl ="https://fbref.com/"+soup.find('div',attrs={'id':'meta'}).find('a').get('href')
    r=requests.get(previousSeasonUrl)
    soup = bs(r.content, 'html.parser')
    table_teams=soup.find_all('table')
    all_table=[]
    #xử lý các bảng
    for table in table_teams:
        cols=[]
        cols.append(['Info', 'Team'])
        tableId=table.get("id")
```

5. Cuối cùng chúng ta sẽ xuất ra file excel.

Bài 2

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

File code : Bai2Top3.py

Ý tưởng : ta sẽ duyệt qua từng cột và lấy ra 3 giá trị thấp nhất và 3 giá trị cao nhất

```
df = pd.read_csv("result.csv", header=[0, 1])

print("Tìm top 3 cầu thủ có điểm cao nhất và thấp nhất ở mỗi chỉ số.")
# Khởi tạo dictionary để lưu kết quả
top_and_bottom = {}

# Duyệt qua từng cột của DataFrame để tìm top và bottom 3 cầu thủ
for col in df.columns:
    # Kiểm tra nếu cột là kiểu số
    if pd.api.types.is_numeric_dtype(df[col]):
        top_players = df.nlargest(3, col) # Top 3 cầu thủ có điểm cao nhất
        bottom_players = df.nsmallest(3, col) # Top 3 cầu thủ có điểm thấp nhất
        top_and_bottom[col] = {
            'top_3': top_players,
            'bottom_3': bottom_players
        }
# print(top_and_bottom)
for col, results in top_and_bottom.items():
    print(f"\nChỉ số: {col}")
    print("Top 3 cầu thủ cao nhất:")
    # in ra thông tin cầu thủ
    print(results['top_3'][[('Info', 'Player'), ('Info', 'Age'), ('Info', 'Team'), ('Info', 'Nation')]]) # Hiển thị tên cầu thủ và
    print("Top 3 cầu thủ thấp nhất:")
    print(results['bottom_3'][[('Info', 'Player'), ('Info', 'Age'), ('Info', 'Team'), ('Info', 'Nation')]])
```

Kết quả :

Vì có nhiều chỉ số nên ta sẽ chỉ lấy 3 chỉ số minh họa :

Chỉ số ('Info', 'Age') , ('Playing Time', 'MP'), ('Playing Time', 'Starts'):

```

Chỉ số: ('Info', 'Age')
Top 3 cầu thủ cao nhất:
Info
Player Age Team Nation
40 Ashley Young 38 Everton eng ENG
442 Thiago Silva 38 Chelsea br BRA
486 Łukasz Fabiański 38 West Ham United pl POL
Top 3 cầu thủ thấp nhất:
Info
Player Age Team Nation
276 Leon Chiwome 17 Wolverhampton Wanderers eng ENG
283 Lewis Miley 17 Newcastle United eng ENG
111 David Ozoh 18 Crystal Palace eng ENG

Chỉ số: ('Playing Time', 'MP')
Top 3 cầu thủ cao nhất:
Info
Player Age Team Nation
6 Adam Armstrong 26 Southampton eng ENG
433 Stephy Mavididi 25 Leicester City eng ENG
466 Václav Hladký 32 Ipswich Town cz CZE
Top 3 cầu thủ thấp nhất:
Info
Player Age Team Nation
14 Alex Iwobi 27 Everton ng NGA
183 Ionuț Radu 26 Bournemouth ro ROU
197 Jakub Stolarczyk 22 Leicester City pl POL

Chỉ số: ('Playing Time', 'Starts')
Top 3 cầu thủ cao nhất:
Info
Player Age Team Nation
466 Václav Hladký 32 Ipswich Town cz CZE
172 Harry Winks 27 Leicester City eng ENG
6 Adam Armstrong 26 Southampton eng ENG
Top 3 cầu thủ thấp nhất:
Info
Player Age Team Nation
101 Dane Scarlett 19 Ipswich Town eng ENG
111 David Ozoh 18 Crystal Palace eng ENG
185 Ivan Perišić 34 Tottenham Hotspur hr CRO

```

- 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.

File code : Bai2MedianMeanStd.py

Ý tưởng :

Đầu tiên ta sẽ lọc ra các bảng có dữ liệu không phải là kiểu số

Nếu muốn tính các giá trị cần tìm của mỗi đội thì ta sẽ dùng groupby theo cột chứa thông tin mà cầu thủ đó nằm trong cột nào

Nếu trong toàn giải thì tính như bình thường

Hợp kết quả của bước 2 và bước 3 lại và xuất excel

```
print(" trung vị của mỗi chỉ số. Tìm trung bình và độ lệch chuẩn của mỗi chỉ số trong toàn giải và của mỗi đội")
df1=df
df1.columns = ['_'.join(filter(None, col)).strip() for col in df.columns]
total_scores = df.select_dtypes(include='number')
total_scores['Team'] = df['Info_Team']
#tính theo mỗi đội
total_scores=total_scores.groupby('Team').agg(['median', 'mean', 'std'])
#làm lại tên cột cho giống mẫu
total_scores.columns = ['_'.join(col).strip() for col in total_scores.columns]
total_scores.reset_index()
#tính toàn giải
all_stats = df.select_dtypes(include='number').agg(['median', 'mean', 'std'])
all_row = []
for col in all_stats.columns:
    all_row.append(all_stats[col]['median'])
    all_row.append(all_stats[col]['mean'])
    all_row.append(all_stats[col]['std'])
#thêm kết quả của toàn giải vào
total_scores.loc['All'] = all_row
#cho kết quả toàn giải lên đầu
total_scores = total_scores.sort_index(ascending=True)
print(total_scores)
total_scores.to_csv('result2.csv')
print("Đã xuất Excel result2 thành công")
```

Kết quả ta được file result2.csv

Team				
A	B	C	D	E
Team	Info_Age_median	Info_Age_mean	Info_Age_std	Playing
All	25	25.49075975	4.146918242	
Arsenal	24	24.76190476	2.547641299	
Aston Villa	26	25.95652174	3.548088613	
Bournemouth	24.5	25.03846154	3.538143798	2
Brentford	26	25.8	3.593976442	
Brighton and Hove Albion	23.5	24.78571429	5.698324269	
Chelsea	22	23	3.905124838	
Crystal Palace	25.5	25.16666667	4.280051469	2
Everton	26	26.34782609	4.858064482	
Fulham	27	27.9047619	3.360130383	

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

File code : Bai2_his.py

- 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

File code : Bai2_team.py

Ý tưởng : ta sẽ groupby theo cột chứa tên của đội bóng và lấy ra đội bóng có chỉ số trung bình cao nhất

```

total_scores=df.select_dtypes(include='number')
total_scores[('Info','Team')] = df[('Info','Team')]
total_scores=total_scores.groupby(('Info','Team')).mean()
best_teams = {}
# Duyệt qua từng cột của DataFrame để tìm đội bóng có điểm cao nhất
for col in df.columns:
    if pd.api.types.is_numeric_dtype(df[col]):
        # Tìm đội bóng có tổng điểm cao nhất
        best_team = total_scores[col].idxmax()
        max_score = total_scores[col].max()
        print(f" - Đội bóng có chỉ số: {col} cao nhất: {best_team} với tổng điểm: {max_score}")

```

Output: ta chỉ lấy 1 vài kết quả minh họa

```

- Đội bóng có chỉ số: ('Info', 'Age') cao nhất: West Ham United với tổng điểm: 28.272727272727273
- Đội bóng có chỉ số: ('Playing Time', 'MP') cao nhất: Southampton với tổng điểm: 29.541666666666668
- Đội bóng có chỉ số: ('Playing Time', 'Starts') cao nhất: Southampton với tổng điểm: 21.0
- Đội bóng có chỉ số: ('Playing Time', 'Min') cao nhất: Southampton với tổng điểm: 1889.0
- Đội bóng có chỉ số: ('Performance', 'Ast') cao nhất: Manchester City với tổng điểm: 3.238095238095238
- Đội bóng có chỉ số: ('Performance', 'G-PK') cao nhất: Manchester City với tổng điểm: 4.0476190476190474
- Đội bóng có chỉ số: ('Performance', 'PK') cao nhất: Arsenal với tổng điểm: 0.47619047619047616
- Đội bóng có chỉ số: ('Performance', 'CrdY') cao nhất: Chelsea với tổng điểm: 4.32
- Đội bóng có chỉ số: ('Performance', 'CrdR') cao nhất: Liverpool với tổng điểm: 0.22727272727272727
- Đội bóng có chỉ số: ('Expected', 'xG') cao nhất: Liverpool với tổng điểm: 4.1000000000000005
- Đội bóng có chỉ số: ('Expected', 'npxG') cao nhất: Liverpool với tổng điểm: 3.7545454545454544
- Đội bóng có chỉ số: ('Expected', 'xAG') cao nhất: Liverpool với tổng điểm: 2.9318181818181817
- Đội bóng có chỉ số: ('Progression', 'PrgC') cao nhất: Manchester City với tổng điểm: 53.476190476190474
- Đội bóng có chỉ số: ('Progression', 'PrgP') cao nhất: Southampton với tổng điểm: 105.91666666666667
- Đội bóng có chỉ số: ('Progression', 'PrgR') cao nhất: Southampton với tổng điểm: 104.875

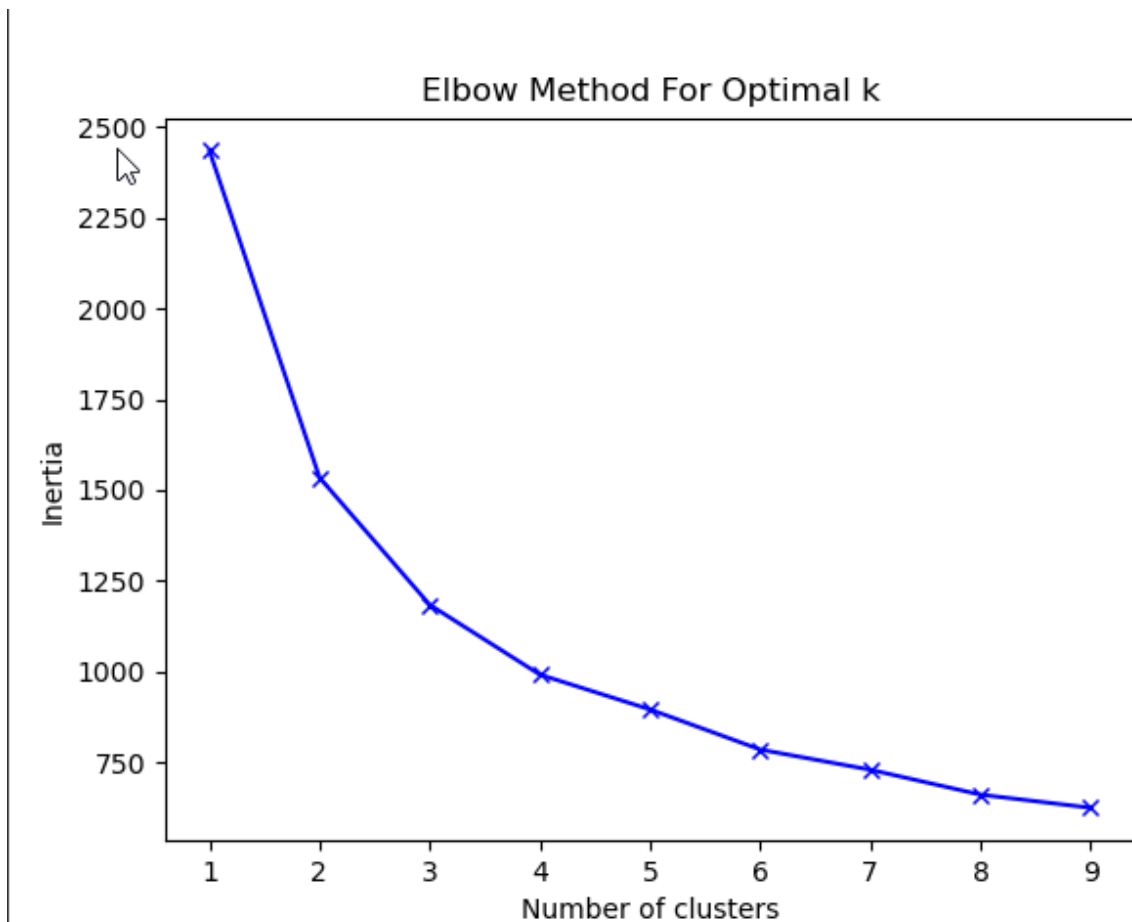
```

-Theo em , đội Manchester City sẽ có phong độ tốt nhất vì đội đó có chỉ số ghi bàn và hỗ trợ cao nhất

Bài 3

File code : Bai3.py

Ý tưởng : Để có thể chia ra được số nhóm , ta sẽ dùng thuật toán elbow để chọn



Ta thấy đến $k=4$ thì khoảng cách tăng không đáng kể, nên ta sẽ chọn $k=4$

-Sau đó ta sẽ chọn ra 4 điểm ngẫu nhiên và dùng thuật toán Kmean để phân loại các cầu thủ, ta sẽ chọn các chỉ tiêu: Tuổi, số trận đá, thời gian chơi, số bàn thắng và kiến tạo để đánh giá và làm chuẩn các giá trị về khoảng 1 đến 11

```
players = pd.read_csv("result.csv", header=[0, 1])
players.columns = ['_'.join(filter(None, col)).strip() for col in players.columns]
#các chỉ số để đánh giá
features = ["Info_Age", "Playing Time_MP", "Playing Time_Min", "Performance_G-PK", "Performance_Ast"]
players=players.dropna(subset=features)
data = players[features].copy()
#chuẩn dữ liệu về từ khoảng 1 đến 11
data = ((data - data.min()) / (data.max() - data.min())) * 10 + 1
```

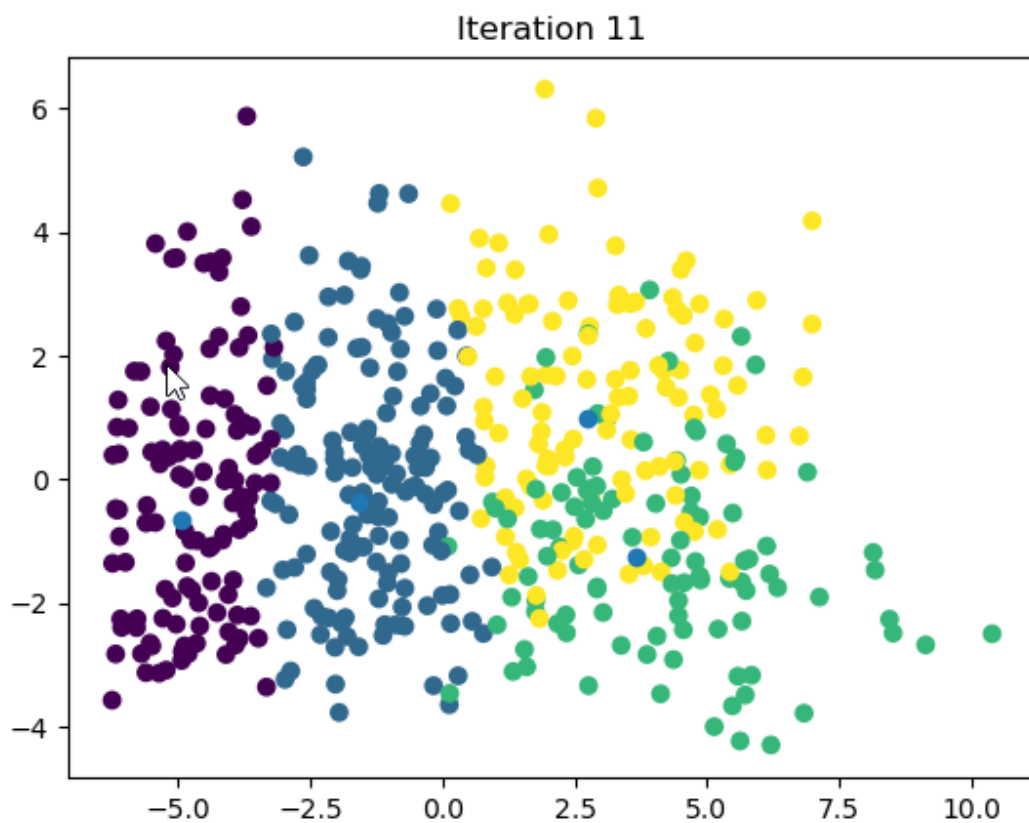
-Vì ta sẽ chọn 4 điểm ngẫu nhiên nên số bước lặp sẽ khác nhau
Ví dụ trong trường hợp ta chọn 4 điểm như sau:


```
centroid = data.apply(lambda x: float(x.sample()))
```

	0	1	2	3
Info_Age	4.809524	4.333333	5.285714	4.809524
Playing Time_MP	2.590909	5.090909	5.772727	8.500000
Playing Time_Min	3.464162	1.069204	5.676223	4.633218
Performance_G-PK	1.500000	2.000000	4.500000	3.000000
Performance_Ast	1.000000	4.333333	1.000000	1.000000

←[2K

Ta sẽ mất 11 lần lặp để được kết quả như sau :
 (ảnh mỗi lần lặp) sẽ được lưu ở file Bai3Image trong thư mục ảnh



Ta sẽ lấy ra các 5 cầu thủ ở đầu mỗi nhóm , ta được 4 bảng như sau:

	Performance_Ast	1.000000	4.333333	1.000000	1.000000			
	Info_Player	Info_Age	Playing Time_MP	Playing Time_Min	Performance_G-PK	Performance_Ast		
0	Aaron Cresswell	33	11	436.0	0	0		
1	Aaron Hickey	21	9	713.0	0	0		
2	Aaron Ramsdale	25	6	540.0	0	0		
14	Alex Iwobi	27	2	140.0	0	0		
15	Alex McCarthy	33	5	450.0	0	0		
	Info_Player	Info_Age	Playing Time_MP	Playing Time_Min	Performance_G-PK	Performance_Ast		
3	Aaron Wan-Bissaka	25	22	1780.0	0	2		
7	Adam Lallana	35	25	850.0	0	1		
9	Adam Webster	28	15	1144.0	0	0		
10	Adam Wharton	19	16	1297.0	0	3		
11	Adama Traoré	27	17	377.0	2	3		
	Info_Player	Info_Age	Playing Time_MP	Playing Time_Min	Performance_G-PK	Performance_Ast		
5	Abdul Fatawu Issahaku	19	40	2814.0	6	13		
6	Adam Armstrong	26	46	3745.0	17	13		
12	Alejandro Garnacho	19	36	2565.0	7	4		
17	Alexander Isak	23	30	2255.0	16	2		
18	Alexis Mac Allister	24	33	2599.0	4	5		
	Info_Player	Info_Age	Playing Time_MP	Playing Time_Min	Performance_G-PK	Performance_Ast		
4	Abdoulaye Doucouré	30	32	2629.0	7	1		
8	Adam Smith	32	28	2150.0	0	2		
13	Alex Iwobi	27	30	2192.0	5	2		
21	Alisson	30	28	2520.0	0	0		
22	Alphonse Areola	30	31	2699.0	0	0		