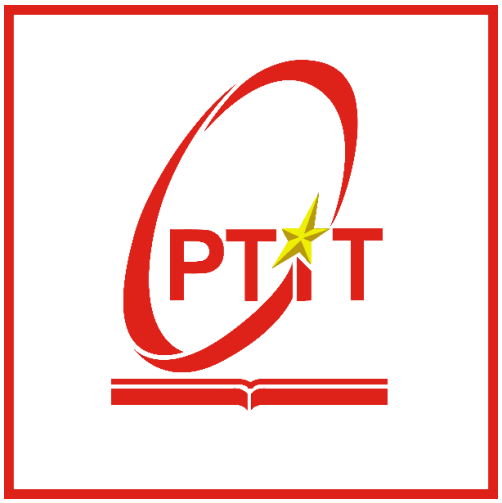
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



BÁO CÁO BÀI TẬP

MÔN HỌC: LẬP TRÌNH VỚI PYTHON

Giảng viên : Thầy Kim Ngọc Bách

Sinh viên : Đinh Quốc Đại

Lớp : D22CQCN07-B

Mã sinh viên : B22DCCN175

Tháng 10/2024

Contents

Bài 1:

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 liều vào dictionary rồi chuyển về dataframe
  - Lưu bảng thành các file .csv
- 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.
- Code:

```python
def crawler(url, id_div, cnt):
    # url = 'https://fbref.com/en/comps/9/2023-2024/stats/2023-2024-Premier-League-Stats'
    print(url,id_div)
    r1 = requests.get(url)
    soup1 = bs(r1.content, 'html.parser')
    table1 = soup1.find('div', {'id': id_div})
    if cnt== 5:
        f= open('output.html','w')
        f.write(table1.prettify())
    comment1 = table1.find_all(string=lambda text: isinstance(text, Comment))
    data1 = bs(comment1[0], 'html.parser').find_all('tr')  # chuyển string về dạng tags
    ans1 = dict()
    for i, g in enumerate(data1[1].find_all('th')):
        if i !=0:
            ans1[g.get('data-stat')] = []
    for i in range(2, len(data1)):
        # tmp chứa tất cả thông tin của player
        tmp1 = data1[i].find_all('td')
        for j, x in enumerate(tmp1):
            if x.get('data-stat') in ans1.keys():
                if x.get('data-stat') == 'nationality':
                    s = x.getText().split(" ")
                    ans1[x.get('data-stat')].append(s[0])
                else:
                    ans1[x.get('data-stat')].append(x.getText())
    df1 = pd.DataFrame(ans1)
    if cnt == 2:
        df1.drop(['gk_games', 'gk_games_starts', 'gk_minutes','minutes_90s'],axis=1,inplace=True)
    df1.to_csv(f'table{cnt}.csv')
```

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

- Các bảng 1 và từ 3 đến 10 đều 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)
- Do tên các thủ môn không trùng nhau nên bảng 2 sẽ ghép với bảng result đựa trên tên cầu thủ.
- Tiếp sẽ đưa chỉ số minutes về dạng số và xóa các cầu thủ có thời gian thi đầu <= 90 phút
- Sắp xếp lại bảng theo tên cầu thủ tăng dần và tuổi giảm dần
- Lưu bàng vào file result.csv
- Code :

```python
def clean_data():
    result = pd.read_csv('table1.csv')
    for x in range(3,11):
        table = pd.read_csv(f"table{x}.csv")
        for x in table.columns:
            if x in result.columns:
                if x!= 'Unnamed: 0':
                    table.drop(x, axis=1,inplace= True)
        result= pd.merge(result,table,on=['Unnamed: 0'],how= 'inner')
    merge = []
    table2 = pd.read_csv("table2.csv")
    for x in table2.columns:
        if x in result.columns:
            merge.append(x)
    merge.pop(0)
    result = pd.merge(result,table2,on=merge,how='left')
    result.drop(['Unnamed: 0_x','Unnamed: 0_y','minutes_90s','birth_year','matches'],axis=1,inplace=True)
    # chuyển trường minutes về dạng số
    result['minutes'] = result['minutes'].apply( lambda x : int(''.join(x.split(','))))
    result = result [result['minutes']>90]
    result.sort_values(by=["player",'age'],ascending=[True,False])
    result.to_csv('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:

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

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

```
[3]: for x in df.columns[5:]:
         df_copy =df.copy()
         if 'gk' in x:
             df_copy= df_copy[df_copy['position']=='GK']
         tmp =df_copy[['player',x]].sort_values(x)
         tmp.dropna(axis=0,inplace=True)
         print(f'3 cầu thủ có {x} cao nhất')
         display(tmp[-3:][::-1])
         print(f'3 cầu thủ có {x} thấp nhất')
         display(tmp[:3])
```

   - Mô tả:
     - Để tránh làm sai dữ liệu đã thu thập được nên em sẽ tạo một bản copy
     - Đối với những chỉ số của riêng thủ môn thì em sẽ chỉ so sánh những thủ môn với nhau
     - Lấy ra từ bảng 2 trường thông tin là tên cầu thủ và chỉ só tương ứng và thực hiện sắp xếp theo chỉ số\
     - Hiển thị kết quả. Lưu ý do sắp xếp tăng dần nên khi lấy top 3 chỉ số cao nhất sẽ phải đảo ngược kết quả
   - Kết quả : (minh họa kết quả)

3 cầu thủ có age cao nhất

	player	age
486	Ashley Young	38
414	Thiago Silva	38
150	Łukasz Fabiański	38

3 cầu thủ có age thấp nhất

	player	age
87	Leon Chiwome	17
299	Lewis Miley	17
347	David Ozoh	18

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
   - Ý tưởng:
     - Đầu tiên thực hiện chuyển các thông tin cần tính về dạng số.
     - Sau đó tạo một dictionary với key là tên các đội và ‘all’
     - Sử dụng hàm groupby để tính các thông số cần thiết.
     - Sau đó thêm cái giá trị vào dictionay theo đúng thứ tự
     - Khi thực hiện xong chuyển dictionary thành dataframe với key sẽ là index
   - Code:

```
df2 = df.copy()
index2 = ['team']
df2.fillna(0,inplace=True)
for x in df2.columns[6:]:
    index2.append(x)
    df2[x] = df2[x].astype(float)
df2= df2[index2]
```

```
df2=df2.set_index('team')
```

```
mean = df2.groupby('team').mean()
median =df2.groupby('team').median()
std = df2.groupby('team').std()
mean_all= df2.mean()
median_all= df2.median()
std_all =df2.std()
result2 = dict()
col_result2 = []
result2['all']=[]
for x in mean.index:
    result2[x]=[]
for x in df.columns[6:]:
    col_result2.append(f'Median of {x}')
    col_result2.append(f'Mean of {x}')
    col_result2.append(f'Std of {x}')
    result2['all'].append(median_all[x])
    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])
result2=pd.DataFrame.from_dict(result2,orient='index',columns=col_result2)
result2
```

- Kết quả thu được:

[6]:

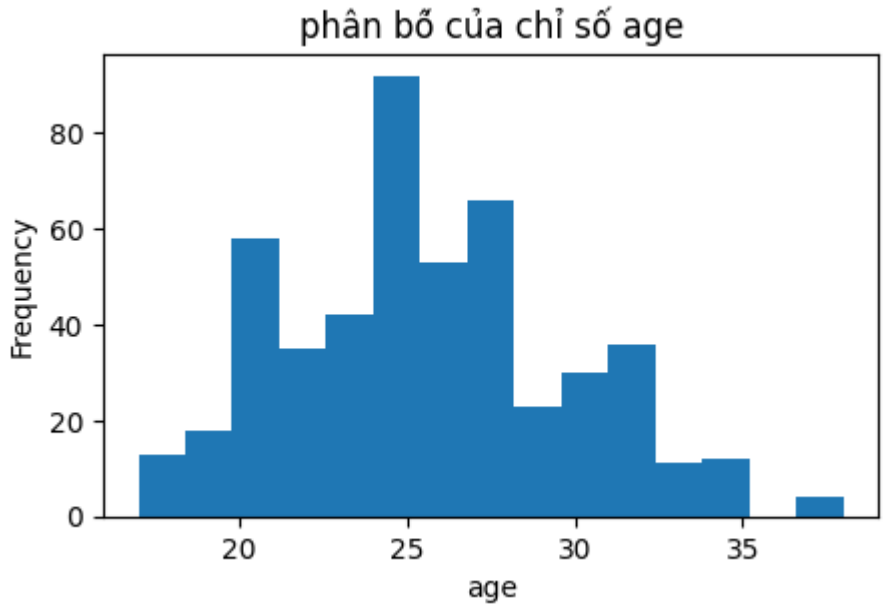
	Median of games	Mean of games	Std of games	Median of games_starts	Mean of games_starts	Std of games_starts	Median of minutes	Mean of minutes	Std of minutes	Median of assists	...	Std of gk_pens_allowed	Median of gk_pens_saved
all	23.0	22.657201	10.136975	16.0	16.941176	11.167179	1419.0	1518.369168	949.241058	1.0	...	0.974582	0.0
Arsenal	27.0	26.809524	10.191266	18.0	19.857143	13.093073	1649.0	1781.571429	1102.745872	2.0	...	0.436436	0.0
Aston Villa	27.0	24.173913	11.109587	20.0	18.130435	12.392462	1652.0	1629.130435	1057.004055	1.0	...	0.208514	0.0
Bournemouth	25.5	22.076923	11.852166	13.0	16.038462	12.732575	1317.5	1438.423077	1074.136832	1.0	...	0.992278	0.0
Brentford	26.0	22.960000	10.346014	15.0	16.720000	10.883933	1321.0	1496.840000	910.122459	1.0	...	0.400000	0.0
Brighton	20.0	20.928571	8.751417	15.0	14.892857	8.603786	1344.5	1338.107143	768.792913	1.0	...	0.786796	0.0
Burnley	16.0	20.392857	9.346575	14.0	14.928571	10.014540	1213.0	1334.857143	851.000706	1.0	...	0.956736	0.0
Chelsea	23.0	21.880000	9.404432	18.0	16.720000	11.066165	1576.0	1495.120000	951.169820	1.0	...	0.439697	0.0
Crystal Palace	22.5	22.458333	9.477567	17.5	17.416667	10.993740	1587.5	1566.000000	910.738831	1.0	...	0.448427	0.0
Everton	28.0	23.304348	11.561829	23.0	18.173913	13.720099	1884.0	1633.173913	1156.480384	0.0	...	1.668115	0.0
Fulham	29.0	27.238095	7.993152	18.0	19.904762	10.084170	1593.0	1773.714286	834.245056	1.0	...	1.745743	0.0
Liverpool	28.0	25.863636	8.993624	17.0	18.954545	8.283531	1671.0	1694.409091	706.077876	2.0	...	0.213201	0.0
Luton Town	23.0	22.840000	9.163696	16.0	16.720000	10.159232	1304.0	1500.840000	865.696188	0.0	...	0.800000	0.0

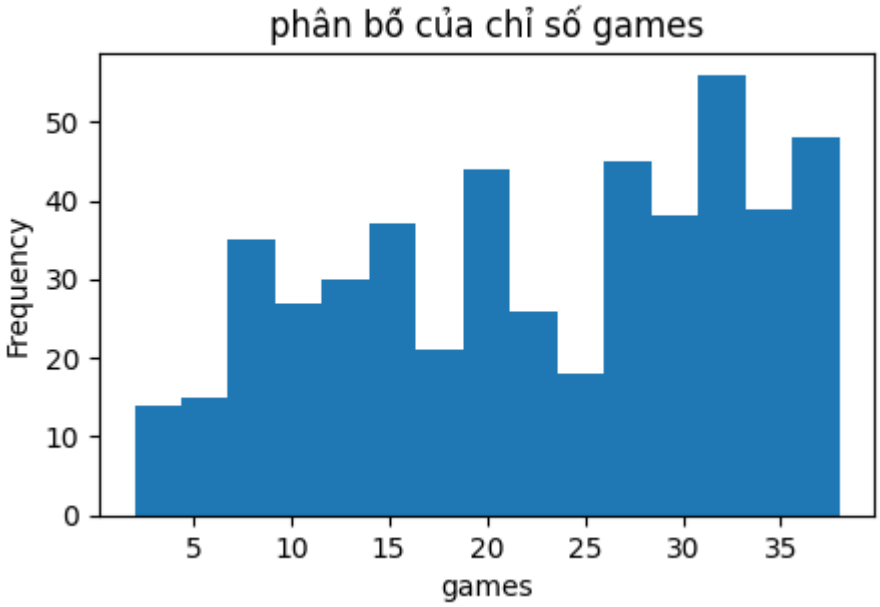
3. Vẽ histogram phân bố của mỗi chỉ số của tất cả các cầu thủ trong toàn giải và mỗi đội
   - Ý tưởng:
     - Lấy dữ liệu ra và loại bỏ giá trị NaN. Đối với những chỉ só của thủ môn thì chỉ lấy dữ liệu của thủ môn
     - Thực hiện vẽ histogram với tất cả các trường của bảng
   - Code:

```python
df3= df.copy()
print("Phân bố của mỗi chỉ số của các cầu thủ trong toàn giải")
print("")
for i,x in enumerate(df3.columns[5:]):
    df_copy =df.copy()
    if 'gk' in x:
        df_copy= df_copy[df_copy['position']=='GK']
    tmp =df_copy[x]
    tmp.dropna(axis=0,inplace=True)
    plt.figure(figsize=(5,3))
    plt.hist(tmp,bins=15)
    plt.title(f"phân bố của chỉ số {x}")
    plt.xlabel(x)
    plt.ylabel("Frequency")
    plt.show()
```

```python
print("Phân bố của mỗi chỉ số của các cầu thủ trong mỗi đội")
print("")
for y in df['team'].unique():
    df_copy = df.copy()
    df_copy = df_copy[df_copy['team']==y]
    for i,x in enumerate(df_copy.columns[5:]):
        if 'gk' in x:
            df_copy= df_copy[df_copy['position']=='GK']
        tmp = df_copy[x]
        tmp.dropna(axis=0,inplace=True)
        plt.figure(figsize=(5,3))
        plt.hist(tmp,bins=15)
        plt.title(f"Phân bố chỉ số {x} của cầu thủ trong đội bóng {y}")
        plt.xlabel(x)
        plt.ylabel("Frequency")
        plt.show()
```

- Kết quả: (kết quả chi tiết xem tại fille bt2.ipynb)





4. Tìm đội bóng có chỉ số cao nhất ở mỗi chỉ số. Theo bạn đội nào có phong độ tốt nhất giải ngoại hạng anh 2023-2024

4.1) Tìm đội bóng có chỉ số cao nhất ở mỗi chỉ số.

- Code:

```
df5=  pd.read_csv('result2.csv')
df5.drop([0],axis=0,inplace=True)

df5_col=df5.columns[1:]
ans =dict()

length= 0
for x in df5_col:
    c=df5[df5[x]==max(df5[x])]['Unnamed: 0']
    ans[x] =list(c)
    length=max(length,len(ans[x]))
for x in ans.keys():
    while len(ans[x])<length:
        ans[x].append('')
ans2= pd.DataFrame.from_dict(ans)
ans2.to_csv('result4.csv')
```

0.2s Python

- Giải thích
  - Đọc dữ liệu từ file result2.csv(file kết quả của phần 2). Xóa bỏ đi dòng dầu tiên(dòng lưu thông tin ‘all’)
  - Lấy các cột chỉ số và tạo dictionary để lưu kết quả
  - Thực hiện truy vấn trong bảng: Lấy ra đội có chỉ số bằng chỉ số max rồi chuyển về list thêm vào dictionary kết quả
  - Điều chỉnh độ dài các mảng trong dictionary sao cho các mảng có độ dài bằng nhau để chuyển kiểu dữ liệu về dataframe
  - Lưu lại kết quả thành file result4.csv
  - Note: Cột ’Unnamed: 0’ chứa tên đội
- Kết quả:

	Median of games	Mean of games	Std of games	Median of games_starts	Mean of games_starts	Std of games_starts	Median of minutes	Mean of minutes	Std of minutes	Median of assists	...	Std of gk_pens_allowed
0	Fulham	Fulham	Wolves	Manchester City	Fulham	Everton	Manchester City	Manchester City	Everton	Arsenal	...	Fulham
1	Manchester City				Manchester City					Liverpool	...	
2											...	
3											...	
4											...	
5											...	

(minh họa kết quả)

4.2) Theo bạn đội nào có phong độ tốt nhất ngoại hạng anh 2023-2024

- Ý tưởng: Đếm xem mỗi đội có bao nhiêu chỉ số đạt max. Đội có nhiều chỉ số đạt max nhất sẽ là đội có phong độ tốt nhất
- Code:

```
### Đếm số lượng chỉ số đạt max của mỗi đội
team = dict()
for x in df5['Unnamed: 0']:
    team[x]=0
for x in ans.keys():
    for y in ans[x]:
        if y!= '':
            team[y]+=1
for x in team.keys():
    print(x,team[x])
```

0.0s Python

- Kết quả:

```
Arsenal 69
Aston Villa 46
Bournemouth 49
Brentford 51
Brighton 33
Burnley 36
Chelsea 45
Crystal Palace 46
Everton 60
Fulham 88
Liverpool 116
Luton Town 50
Manchester City 162
Manchester Utd 41
Newcastle Utd 53
Nott'ham Forest 42
Sheffield Utd 40
Tottenham 52
West Ham 62
Wolves 45
```

⇨ Từ kết quả này thì Manchester City là đội bóng có phong độ tốt nhất mùa 2023-2024

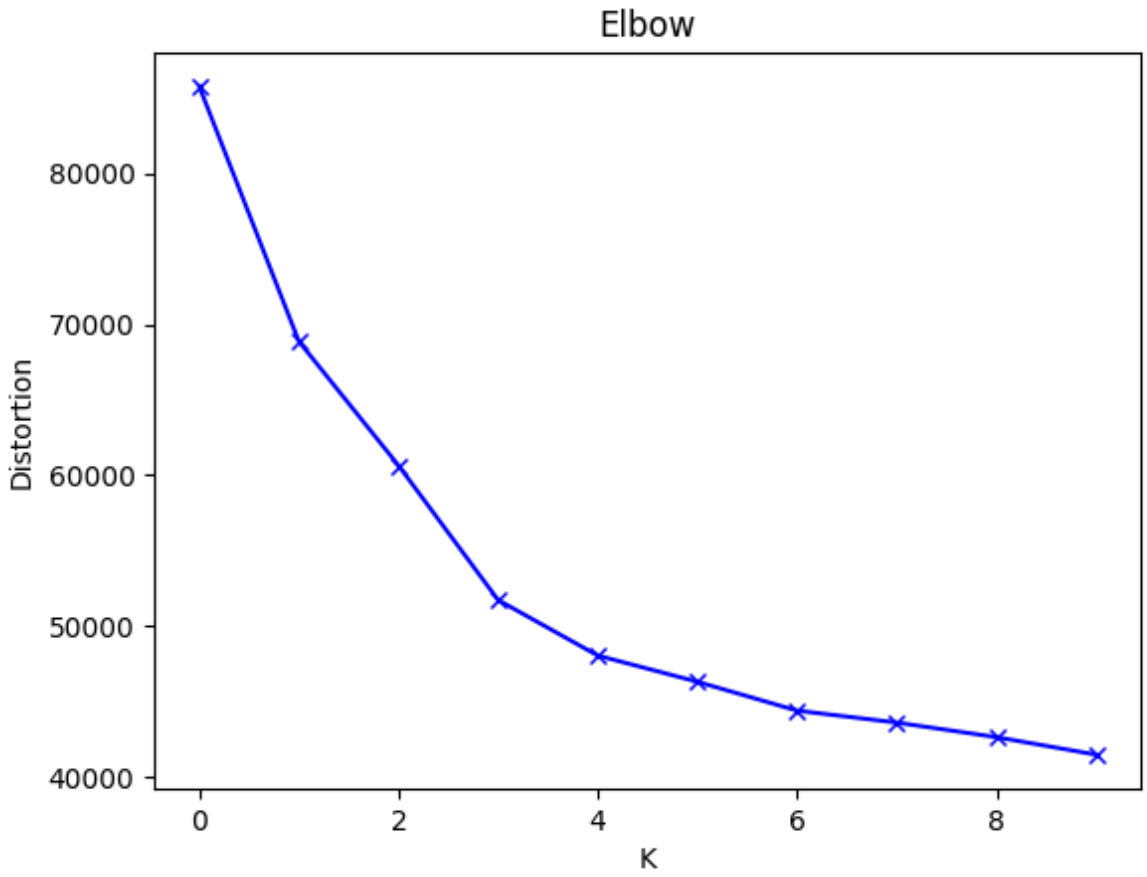
Bài 3:

1. Sử dụng Kmean để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau
   - Code chuẩn hóa dữ liệu và thực hiện thuật toán elbow

```
df1 = df.copy()
for x in df1.columns:
    if type(x) == str:
        lableEncoder = LabelEncoder()
        df1[x] =lableEncoder.fit_transform(df1[x])
    else:
        df1[x].fillna(df1[x].mean(),inplace=True)
standardScaler = StandardScaler()
df1=standardScaler.fit_transform(df1)
```

```
distortions=[]
for K in range(1,11):
    kmeanModel =KMeans(n_clusters=K)
    kmeanModel.fit(df1)
    distortions.append(kmeanModel.inertia_)
plt.plot(range(10),distortions,'bx-')
plt.xlabel('K')
plt.ylabel('Distortion')
plt.title('Elbow')
plt.show()
```

   - Kết quả:



⇨ Từ bảng kết quả trên, Chia các cầu thủ thành 5 nhóm chính.

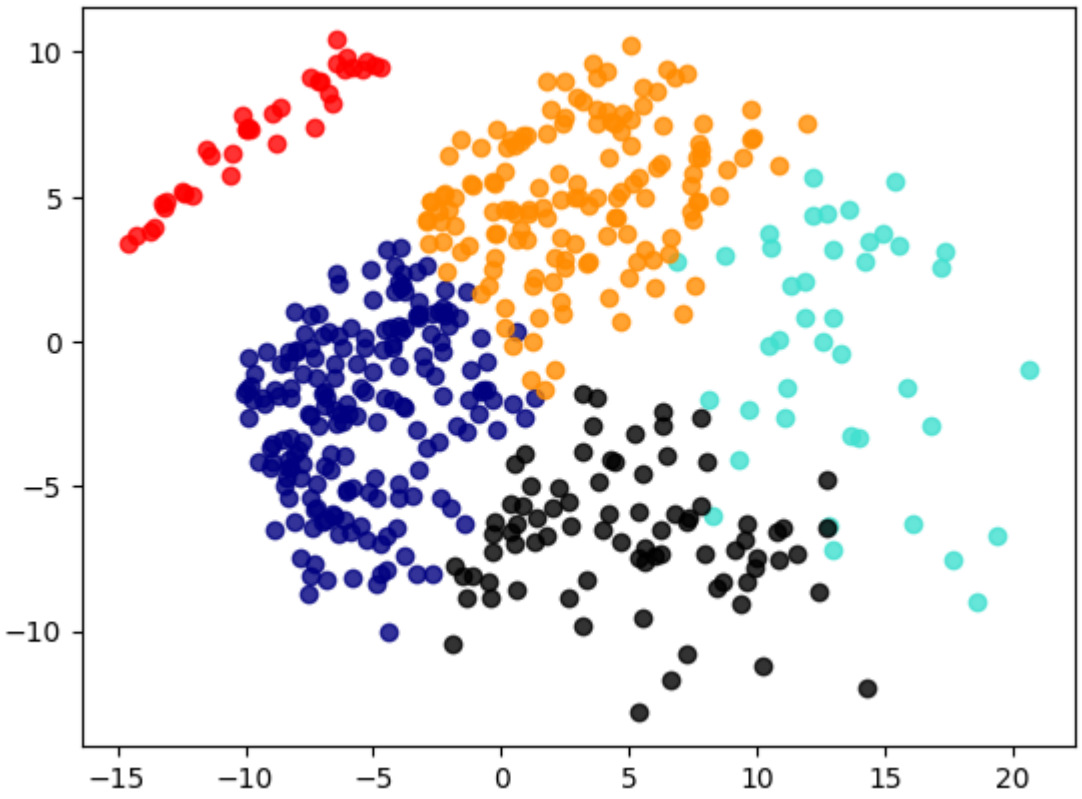
- Nhận xét: Việc chia cầu thủ thành 5 nhóm khá đúng với thực tế. Trong thế giới bóng đá hiện tại, sẽ có 5 nhóm cầu thủ chính: thủ môn, hậu vệ, tiền vệ phòng ngự, tiền vệ tấn công, tiền đạo.

2. Sử dụng thuật toán PCA, giảm số chiều dữ liệu xuống 2 chiều và vẽ hình phân cụm các dữ liệu trên mặt 2D
   - Code:

```
# phân loại thành 5 cụm
model = KMeans( n_clusters= 5)
ans =model.fit_predict(df1)
```

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
X_r = pca.fit(df1).transform(df1)

plt.figure()
colors = ['navy','turquoise','darkorange','red','black']

for color, i in zip(colors, [0,1,2,3,4]):
    plt.scatter(X_r[ans == i, 0], X_r[ans == i, 1], color = color, alpha = 0.8, lw = 1)
```

   - Giải thích code:
     - Thực hiện phân các cầu thủ thành 5 cụm như đã phân tích ở trên
     - Sử dụng PCA để giảm số chiều dữ liệu xuống còn 2
     - Sử dụng vòng lặp để vẽ từng cụm dữ liệu
   - Kết quả:



3. Viết chương trình vẽ biểu đồ rada so sánh cầu thủ
   - Ý tưởng:

- Tìm giá trị min và max của các chỉ số cần so sánh để thực hiện chuẩn hóa min-max
- Việc chuẩn hóa sẽ giúp hình khi được vẽ ra được tương sứng không bị lệch khi so sánh.
- Thực hiện vẽ biểu đồ. Circle_close được sử dụng để tạo hình khép kín khi vẽ chỉ số của các cầu thủ

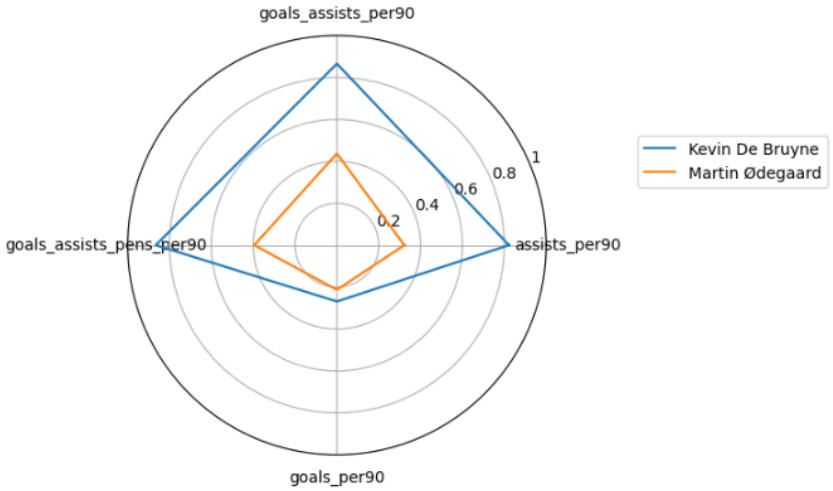
- Code

```
import string

# chuẩn hóa min max
def normalize(value:list, max_val:list,min_val:list) -> list:
    target= []
    for min_v,v,max_v in zip(min_val,value,max_val):
        target.append((v-min_v)/(max_v-min_v))
    return target
def draw_radar_chart(player1:string, player2:string, att:list) -> None:

    # Lấy giá trị min max các cột
    min_att=[]
    max_att=[]
    for x in att:
        min_att.append(df[x].min())
        max_att.append(df[x].max())
    min_att.append(min_att[0])
    max_att.append(max_att[0])
    # Lấy chỉ số 2 cầu thủ và chuẩn hóa chỉ số
    att_p1= np.array(df[df['player']== player1][att])[0]
    att_p1= np.append(att_p1,att_p1[0])
    att1= normalize(att_p1.tolist(),max_att,min_att)

    att_p2= np.array(df[df['player']== player2][att])[0]
    att_p2= np.append(att_p2,att_p2[0])
    att2= normalize(att_p2.tolist(),max_att,min_att)

    # vẽ biểu đồ
    circle = np.linspace(0,2*np.pi,len(att),endpoint=False)
    circle_close= np.append(circle,0)
    ax = plt.subplot(polar=True)
    plt.yticks([0.2,0.4,0.6,0.8,1],['0.2','0.4','0.6','0.8','1'])
    plt.ylim(0,1)
    plt.xticks(circle,att)
    ax.plot(circle_close,att1,label = player1)
    ax.plot(circle_close,att2,label = player2)
    ax.legend(loc='center left',bbox_to_anchor=(1.2,0.7))
    plt.show()
```

- Kết quả:

```
draw_radar_chart('Kevin De Bruyne','Martin Ødegaard',['assists_per90', 'goals_assists_per90', 'goals_assists_pens_per90','goals_per90'])
```

```
[np.float64(0.0), np.float64(0.0), np.float64(0.0), np.float64(0.0), np.float64(0.0)] [np.float64(0.9), np.float64(1.19), np.float64(1.19), np.float64(1.
08), np.float64(0.9)]
[0.74 1.03 1.03 0.29 0.74] [0.29 0.52 0.47 0.23 0.29]
```



Bài 4: