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

**A red circle with a yellow star and a book

Description automatically generated**

**BÁO CÁO BÀI TẬP LỚN**

**LẬP TRÌNH VỚI PYTHON**

|  |  |
| --- | --- |
| ***Giảng viên hướng dẫn:*** | ***GV.Kim Ngọc Bách*** |

**Sinh viên thực hiện : Tạ Quang Chiến**

**Mã sinh viên : B22DCCN109**

**Lớp : D22CQCN01-B**

**Số điện thoại : 0968118125**

**Nhóm : 11**

Mục lục

[**I.** **Cơ sở** 2](#_Toc181390514)

[**1. Yêu cầu** 2](#_Toc181390515)

[**2. Xây dựng cơ sở cho bài toán** 3](#_Toc181390516)

[**a) Cấu trúc thư mục** 3](#_Toc181390517)

[**b) Xây dựng file object** 3](#_Toc181390518)

[**II.** **Tiến hành thực hiện** 5](#_Toc181390519)

[**1.Bài tập 1** 6](#_Toc181390520)

[**a)Chuẩn bị các thư viện cần thiết** 6](#_Toc181390521)

[**b) Vấn đề thu thập dữ liệu từ website** 6](#_Toc181390522)

[**c) Thu thập dữ liệu từ từng trang web** 9](#_Toc181390523)

[**d) Lọc,lưu kết quả và hoàn thành** 11](#_Toc181390524)

[**2. Bài tập 2** 14](#_Toc181390525)

[**a) Chuẩn bị dữ liệu** 14](#_Toc181390526)

[**b)Tìm top 3 cầu thủ có chỉ số cao nhất và thấp nhất ở mọi chỉ số** 15](#_Toc181390527)

[**c) 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** 17](#_Toc181390528)

[**d) Vẽ histogram** 20](#_Toc181390529)

[**e) Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số** 22](#_Toc181390530)

[**3.Bài tập 3** 25](#_Toc181390531)

[**a) Các thư vận cần thiết** 25](#_Toc181390532)

[**b) Lọc và chuẩn hóa dữ liệu** 26](#_Toc181390533)

[**c) Tìm số điểm hội tụ cho thuật toán Kmean** 28](#_Toc181390534)

[**d) Phân cụm Kmean** 30](#_Toc181390535)

[**e) Sử dụng phương pháp giảm chiều của dữ liệu bằng thuật toán PCA** 31](#_Toc181390536)

[**g) So sánh 2 cầu thủ với radarChart** 32](#_Toc181390537)

[**4. Bài tập 4** 37](#_Toc181390538)

[**a) Update file object** 37](#_Toc181390539)

[**b)Thu thập thông tin cầu thủ** 38](#_Toc181390540)

[**c)Đề xuất phương án định giá cầu thủ** 42](#_Toc181390541)

1. **Cơ sở**

## **1. Yêu cầu**

- Thu thập dữ liệu thống kê [\*] của tất cả các cầu thủ có số phút thi đấu nhiều hơn 90 phút tại giải bóng đá ngoại hạng Anh mùa 2023-2024.

- Ghi kết quả ra file 'results.csv'

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

- 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

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

- Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số. Đánh giá xem đội nào có phong độ tốt nhất giải ngoại Hạng Anh mùa 2023-2024

- Sử dụng thuật toán K-means để phân loại các cầu thủ thành các nhóm có chỉ số giống nhau.

- Nên phân loại cầu thủ thành bao nhiêu nhóm? Vì sao? 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.

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

- 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.

## **2. Xây dựng cơ sở cho bài toán**

### **a) Cấu trúc thư mục**

A screen shot of a computer

Description automatically generated

Phần báo cáo được chia thành 8 file chính bao gồm:

+ file bt1,bt2,bt3,bt4 đùng để đáp ứng các yêu cầu của phần yêu cầu đề bài

+ file object dùng để lưu các đối tượng có trong bài(gồm có đối tượng Player(cầu thủ) và Squad(cho đội bóng))

+file title chứa mảng, hàm để lấy tiêu đề của thuộc tính, thuộc tính của các dối tượng sẽ giải thích chi tiết ở phần sau

+ file radarCharPlot chứa hàm dùng để thực hiện vẽ sơ đồ phục vụ cho yêu cầu bài 3

+ file tranferdata dùng để lấy dữ liệu từ trang web tranferfootball phục vụ cho yêu cầu bài 4

### **b) Xây dựng file object**

File này chứa các lớp cơ bản và là cơ sở để xây dựng các bài toán sau này

Chứa các thành phần:

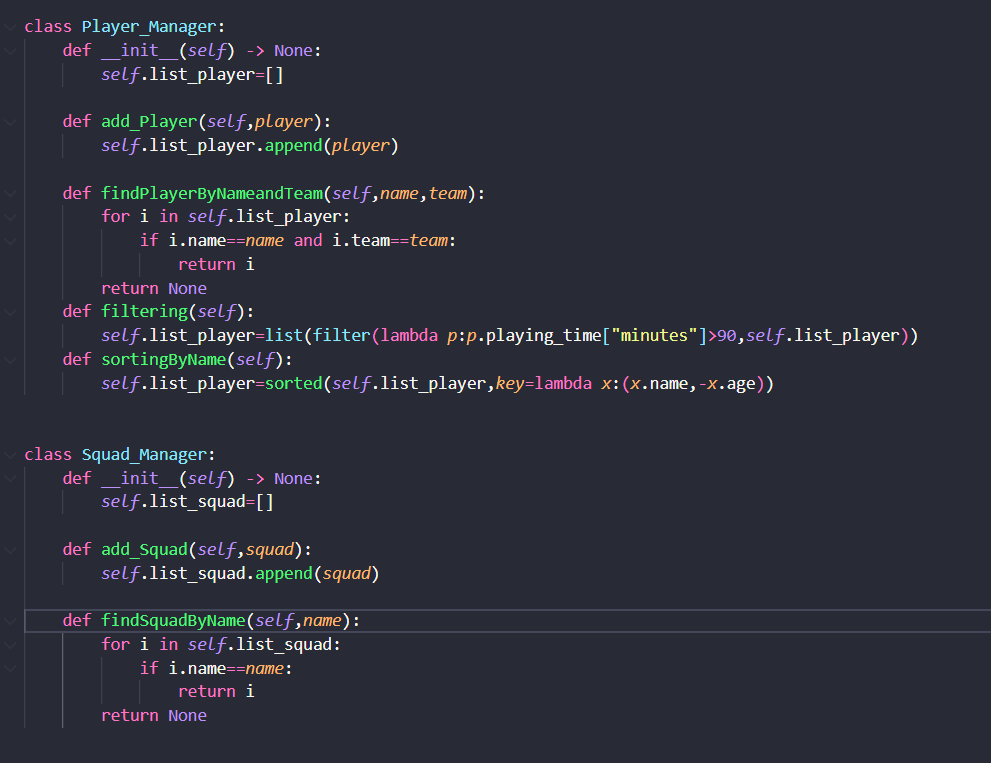
* Class Player(cầu thủ): chứa đầy đủ các thuộc tính mà để bài yêu cầu
  + Ngoài ra còn 1 số hàm set để thuận tiện cho việc thiết lập chỉ số cầu thủ
  + Hàm toString để thuận tiện cho việc debug và đọc ghi file

A screen shot of a computer program

Description automatically generated

Hình 1: một số thuộc tính của class Player

* Class Squad: chứa các thông tin tương tự với Player
* Các lớp quản lý cho 2 class Player và Squad lần lượt là Player\_Manager và Squad\_Manager
  + Cả 2 lớp đều có phương thức add để thêm cầu thủ/đội bóng vào trong danh sách quản lý của chúng
  + Các Squad phân biệt với nhau bởi tên, trong khi đó Player thì được phân biệt bởi Tên và đội, do trong kì chuyển nhượng của mùa giải các cầu thủ được chuyển qua đội tuyển khác( trong trường hợp này coi như là 2 người khác nhau để dễ đàng đánh giá, các cầu thủ đã được kiểm tra để đảm bảo không có 2 cầu thủ trùng tên trong 1 đội)
  + Lớp Player còn có 1 hàm filtering để lọc cầu thủ chơi trên 90 và sortingByName để sắp xếp cầu thủ theo tên



1. **Tiến hành thực hiện**

## **1. Bài tập 1**

### **a) Chuẩn bị các thư viện cần thiết**

A computer screen shot of a program

Description automatically generated

Cần thư viện selenium để lấy file html từ trang web

Thư viện BeautifulSoup để thực hiện phân tích file html thành các thành phần

Import các class từ file object để nhập thông tin

A computer screen shot of a program

Description automatically generated

Hàm này thực hiện chuẩn hóa các dữ liệu không hợp lệ

### **b) Vấn đề thu thập dữ liệu từ website**

- Có tổng cộng 10 trang web để có thể lấy đầy đủ dữ liệu cho 1 đối tượng Player/Squad

- Mỗi trang web sử dụng 1 url riêng, tên các bảng trên web riêng,số lượng thuộc tính của từng đối tượng cũng như tên phân biệt cho các dạng dữ liệu đó.Xong tất cả các trang web đều có các thuộc tính như vậy, ta xây dựng 1 hàm để xử lý việc lấy dữ liệu từ từng url: 

Với các tham số

+ url: Địa chỉ trang web

+*idPlayerTable: id của bảng cầu thủ trong thẻ html của trang web*

+*idSquadTable: : id của bảng đội bóng trong thẻ html của trang web(dữ liệu nay không cần thiểt ở bài tập này nhưng ta vẫn thu thập thông tin để tránh lặp lại việc lấy dữ liệu cho các bài sau)*

+*lengthPlayerData: số lượng thuốc tính của đối tượng Player/squad trong bảng*

+*DataName: tên gọi chung của các thuộc tính có trong trang web*

\* Hàm trả về 2 mảng dữ liệu 1 cho đối tượng Player, 1 cho đối tượng class

**

*Bắt đầu với hàm driver.get(url), để truy cập vào trang web,ta chờ khoảng 3s để tất cả các nội dung trên trang được tải hết,ta chia file html thu được ra thành 2 thành phần của Player và Squad*

* *Bắt đầu với bảng Player*

*A computer screen shot of text

Description automatically generated*

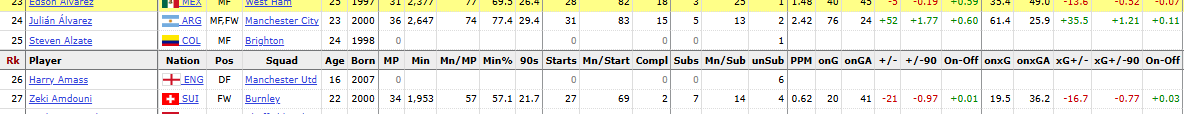
* + *Qua các bước sơ chế, các phương thức của thư viện BeatifulSoup ta thu được thành phẩm là 1 list, mội phần tử trong list đó chính là dãy các thuộc tính của 1 đối tượng Player thu được từ url đó, t bắt đầu lặp hết để xử lý các đoạn dữ liệu*

*A screen shot of a computer program

Description automatically generated*

*Qua xử lý, chuẩn hóa dữ liệu(loại bỏ dấu phẩy của các chỉ số) ta thu được dãy là các thuộc tính của đối tượng Player, 1 dãy các thuộc tính lại được thêm vào mảng resultPlayerData, có nghĩa là mảng resultPlayerData chứa n phần tử tương ứng với n đối tượng Player, mỗi phần tử là thuộc tính của đối tượng Player*

*\*\*(Biến miss đùng để đánh dẫu mỗi 25 đối tượng player được xử lí thì sẽ bỏ qua 1 dòng do cấu trúc của bảng trên trang web cứ mỗi 25 cầu thủ lại lặp lại hàng header thuộc tính)*

**

*\*\*(biến lengthPlayerData dùng để đánh dấu thuộc tính cuối cùng của bảng Player do phần cuối của thẻ tr là thẻ link, không liên quan đến thuộc tính của đối tượng)*

* Tương tự Player, phần lấy dữ liệu cho Squad cũng có cấu trúc tương tự như vậy

*A computer screen shot of a program code

Description automatically generated*

Cuối cùng ta in ra dòng chữ finish để đánh dấu đã hoàn thành thu thập dữ liệu tại trang web, đồng thời trả lại 2 mảng kết quà resultPlayerData và resultSquadData

### **c) Thu thập dữ liệu từ từng trang web**

- Khởi đầu với trang web chứa đầy đủ tất cả các cầu thủ nhất là Playing Time, chứa đầy đủ hơn 760 cầu thủ làm gốc cho bài toán

-Bởi vì trong phần quản lý mảng của 2 đối tượng quản lý đang rỗng nên mỗi cầu thủ hoác đội bóng sẽ gọi hàm tạo mới và được lưu vào trong mảng để các lớp quản lý lưu trữ

A screenshot of a computer program

Description automatically generated

Các đối tượng được thiết lập các chỉ số sau đó được lưu vào lớp quản lý mảng đối tượng tương ứng,

Ngoài ra, cầu thủ còn được thêm vào danh sách các cầu thủ có trong một đội bóng là cơ sở để thực hiện các bài toán sau này

Các url được thực hiện lần lượt cho đến hết 10 url mà bài toán yêu cầu

A screen shot of a computer program

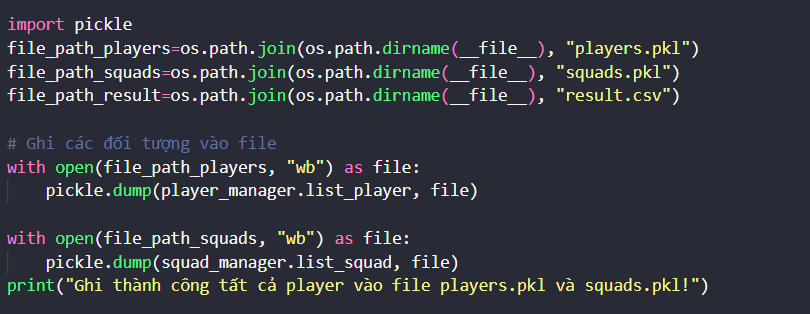
Description automatically generated

Ví dụ thu thập dữ liệu cho url dẫn đến trang web standard

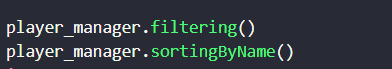
Mỗi url, ta thu được từng loại dữ liệu, mỗi loại này được thiết lập vào trong thuộc tính của đổi tượng Player thông qua các hàm set của lớp, các thuộc tính được set theo mảng dữ liệu tương ứng với nó để tích kiệm thời gian(ví dụ như hàm setPlaying\_time,setPerformance,... ở hình trên. Tương tự với các url còn lại

### **d) Lọc,lưu kết quả và hoàn thành**

- Sau khi thu thập tất cả các đổi tượng Player/Squad tiến hành lưu kết quả vào trong file players.pkl và squads.pkl để phục vụ cho các phần phía sau



* Tiến hành lọc cùng với đó là sắp xếp lại các Player có thời lượng chơi trên 90 trong mùa giải bằng phương thức filtering và sortingByName của lớp quản lý đối tượng Player



* Các đối tượng được lưu vào file csv thông qua thư viện csv

A screen shot of a computer program

Description automatically generated

* Mảng header được import từ file title với từng phần tử là tiêu đề của thuộc tính của các đối tượng Player

A screenshot of a computer program

Description automatically generated

* Sau cùng ta mở file result*.*csv hoàn thành yêu cầu bài 1 của bài toán

A screen shot of a computer

Description automatically generated

## **2. Bài tập 2**

### **a) Chuẩn bị dữ liệu**

A computer screen shot of a program

Description automatically generated

Ta lấy 2 mảng đối tượng Player và Squad từ kết quả của bài 1

NUMBER\_OF\_ATTR là số thuộc tính chính của đối tượng Player/Squad, thực tế là có 172 thuộc tính nhưng ta chỉ lấy 167 thuộc tính(loại bỏ 5 thuộc tính đầu) có thể so sánh được

\* Trong file title, ta xây dựng hàm row, hàm row nhận tham số là 1 đối tượng Player, trả về 1 mảng là giá trị các thuộc tính của đối tượng Player

A screenshot of a computer program

Description automatically generated

### **b) Tìm top 3 cầu thủ có chỉ số cao nhất và thấp nhất ở mọi chỉ số**

- Từ list\_player ta lấy hết tất cả các giá trị thuộc tính của các đối tượng Player sau đó lưu vào mảng to\_compare\_list, các giá trị không hợp lệ của cầu thủ sẽ được chuyển đổi thành 0 để dễ dàng so sánh

A computer screen shot of a program code

Description automatically generated

-Vì số lượng thuộc tính tương đối lớn, việc xem kết quả qua cửa sổ terminal tương đối bất tiện nên thông tin về top chỉ số cao, thấp nhất sẽ được đưa vào file csv để dễ dàng quan sát

* Sử dụng hàm sort cho từng mảng trong to\_compare\_list, với tiêu chí sắp sếp là giá trị của thuộc tính thứ 5+i hay phần tử thứ 5+i trong từng phần tử của mảng to\_compare\_list
* Mảng header lại tiếp tục được sử dụng để làm đầu đề cho các top 3 về thuộc tính đó

A screen shot of a computer code

Description automatically generated

* Cuối cùng mở file result3.csv ta có thể xem thông tin về kết quả

A computer screen shot of a computer screen

Description automatically generated

### **c) 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**

\*Ý tưởng: xây dựng hàm để gom từng thuộc tính của các đối tượng Player vào 1 mảng

Vậy sẽ có 167 mảng tương ứng với 167 thuộc tính có thể so sánh, 1 mảng sẽ có độ dài bằng số lượng Player hiện có, 167 mảng này sẽ được quản lý bởi mảng all\_attr

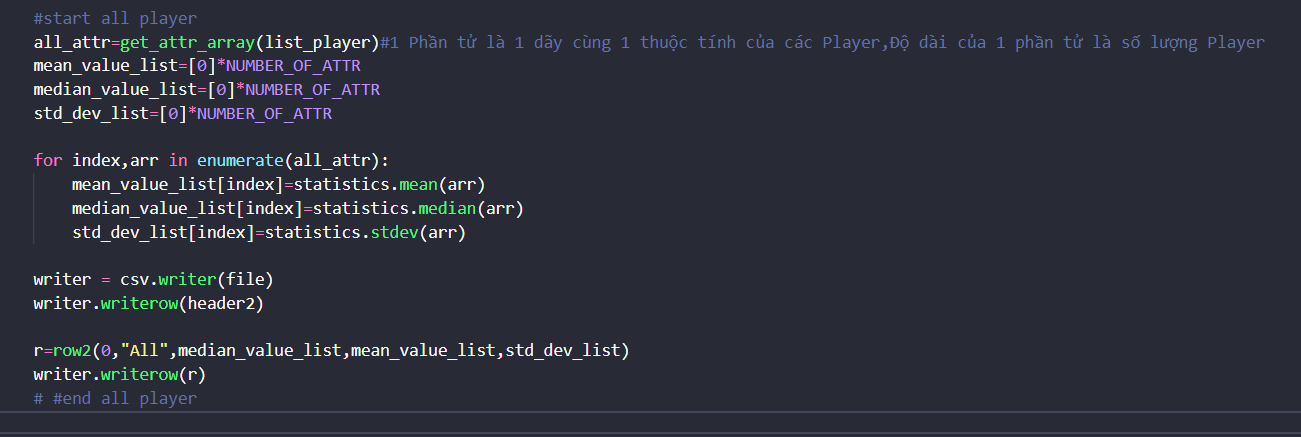
A screenshot of a computer program

Description automatically generated

Hàm def get\_attr\_array(*list\_player*) sẽ thực hiện yêu cầu này bằng cách:

* Lấy 167 thuộc tính của Player lưu thành 1 mảng, mảng này được lưu vào all\_list\_attr\_of\_player
* Từ mảng all\_list\_attr\_of\_player, ta lấy giá trị đầu tiên của tất cả các mảng con trong all\_list\_attr\_of\_player lưu thành 1 mảng và thêm vào mảng all\_attr
* Cứ thế ta lấy giá trị thứ hai, thứ 3,... cho đến hết và tổng cộng có 167 mảng tương ứng được thêm vào mảng all\_attr
* Tất cả các giá trị không hợp lệ sẽ bị loại bỏ thay vì gán chúng bằng 0 để đảm bảo các giá trị trung bình, trung vị, độ lệch chuẩn không bị lệch quá nhiều

\*Tìm trung vị,trung bình, độ lệch chuẩn cho tất cả các Player



- Lấy được mảng all\_attr từ hàm ta mới xây dựng đồng thời khởi tạo 3 mảng mean\_value\_list, median\_value\_list, std\_dev\_list, mỗi phần tử là 1 thuộc tính của dối tượng

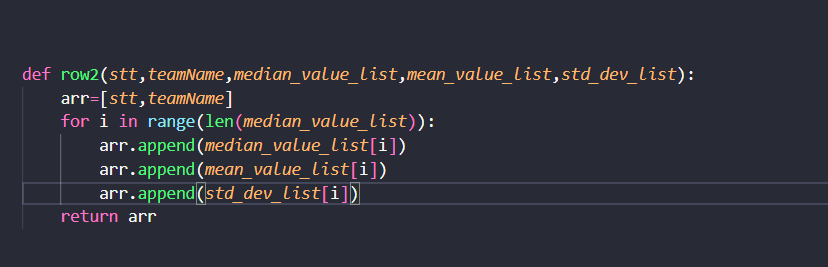
-Có thể tính các giá trị dựa vào thư viện statistics

-header2 là mảng các đầu đề của bảng được định nghĩa ở file title

A screen shot of a computer screen

Description automatically generated

-row2 là hàm trả về mảng giá trị của các thuộc tính,nhận tham số là các mảng trung vị,trung bình, độ lệch chuẩn



Ta gọi hàm row2 sau đó viết vào file csv

\*Tìm trung vị,trung bình, độ lệch chuẩn cho tất cả các Player trong từng team

-Tương tự như tất cả các Player

A screenshot of a computer program

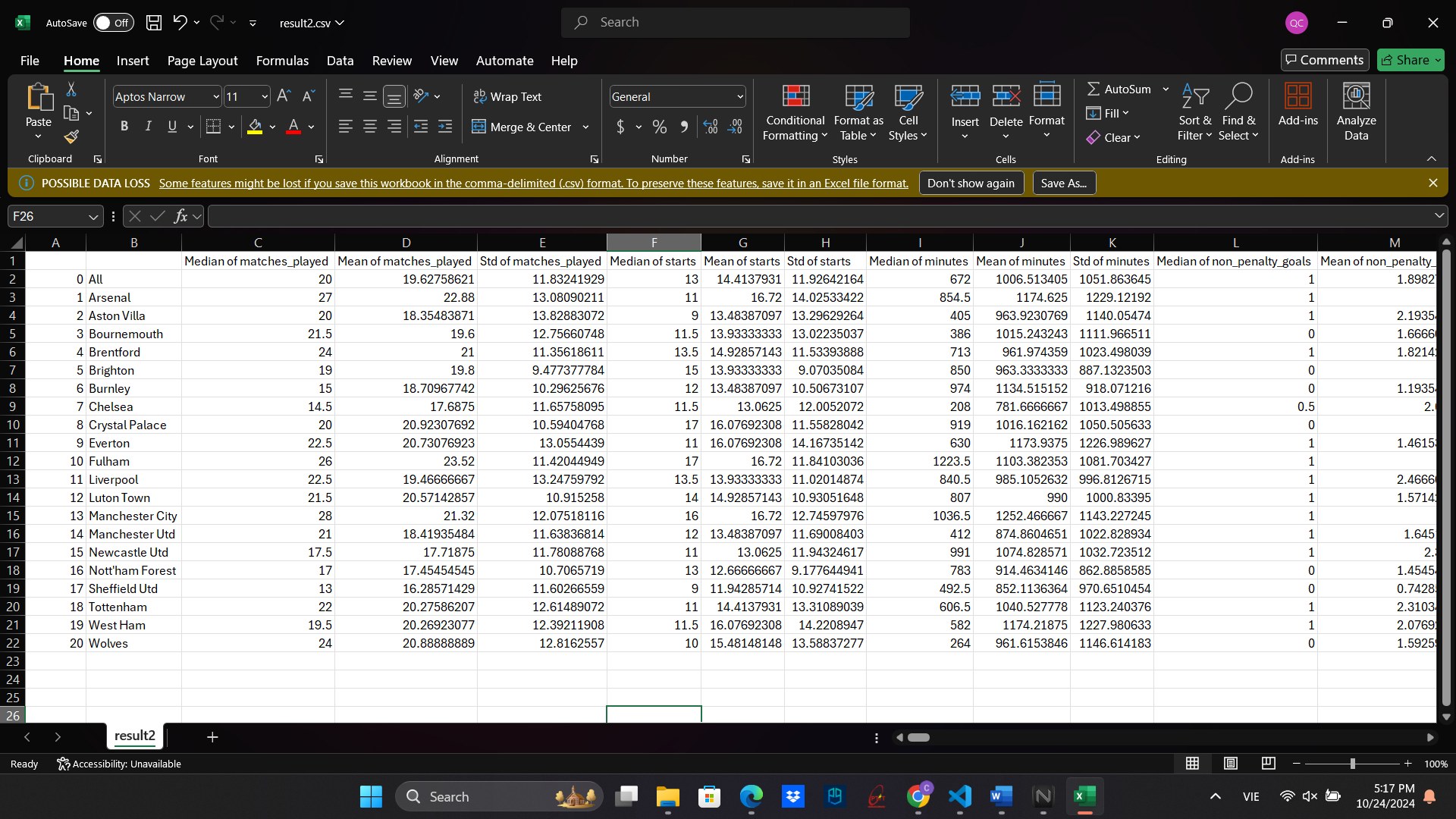
Description automatically generated

Chú ý: tham số truyền vào hàm get\_attr\_array không phải tất cả Player mà chỉ là các Player có trong team đó thôi, có thêm điều kiện độ dài của mảng nhở hơn 2 do số lượng Player trong team quá ít nên không đủ số lượng để tính độ lệch chuẩn(1 đội nào đó chỉ có 1 thủ môn, nên các chỉ số của thủ môn chỉ có 1 nên không thể tính độ lệch chuẩn)  
A screen shot of a computer code

Description automatically generated

Ta tính được trung vị,trung bình, độ lệch chuẩn cho tất cả 20 team sau đó lưu vào file tương tự như phần trên

Cuối cùng mở file và hoàn thành



### **d) Vẽ histogram**

Ta tận dụng hàm get\_attr\_array của phần trên để tập hợp các dữ liệu cùng 1 loại phục vụ cho việc vẽ histogram

Do số lượng histogram khá lớn, ta lưu vào 1 file pdf để tiện theo dõi

**-Vẽ histogram cho tất cả player**  
A screen shot of a computer program

Description automatically generated

**- Vẽ histogram cho từng đội**

A screenshot of a computer program

Description automatically generated

Ta thu được 1 file chứa biểu đồ hítogram của tất cả các cầu thủ và các file chứa histogram của từng đội tuyển như hình bên dưới

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

**\*\*Chú ý**:Các biểu đồ histogram đã được vẽ sẵn bằng hàm drawHistogram() tạo ra các file pdf nên không nhất thiết phải chạy lại hàm này nữa nên hàm này sẽ bị comment  
A blue and green text

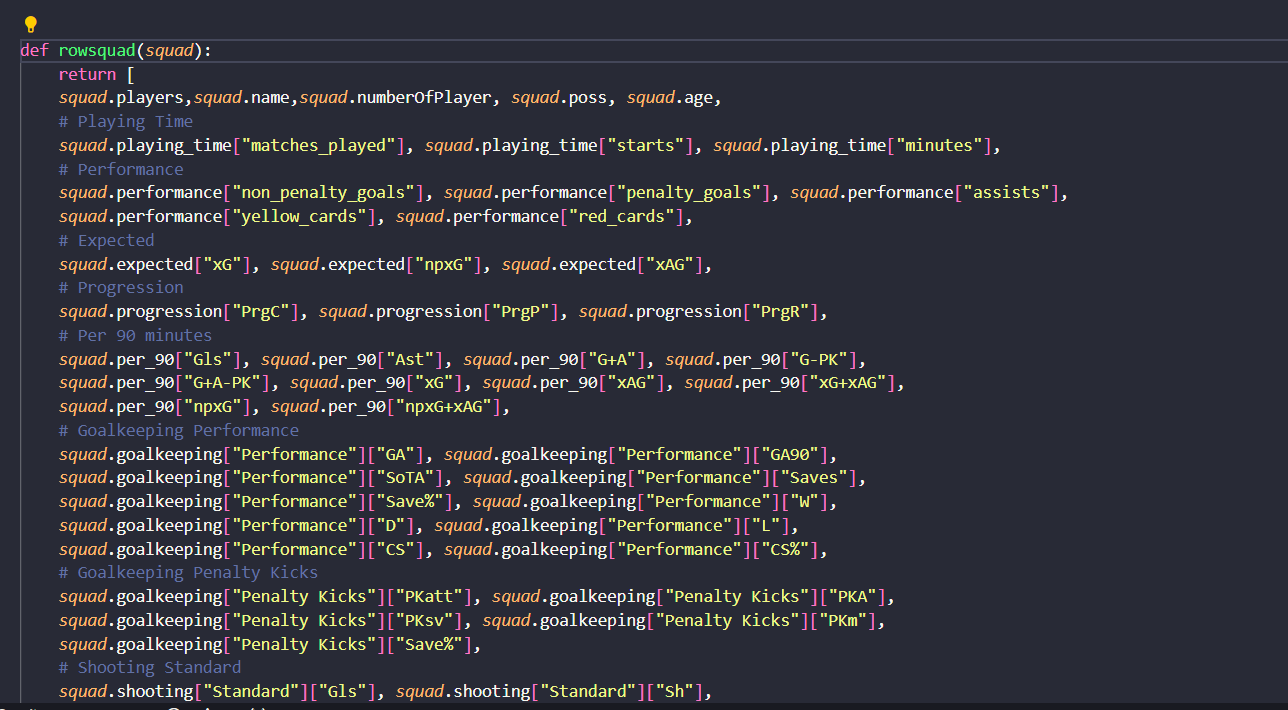
Description automatically generated

### **e) Tìm đội bóng có chỉ số điểm số cao nhất ở mỗi chỉ số**

Ý tưởng: xây dựng 2 mảng, 1 mảng chứa chỉ số cao nhất ở mỗi thuộc tính, 1 mảng chứa tên đội có giá trị thuộc tính cao nhất ứng với chỉ số của mảng đầu tiên  
A computer code with colorful text

Description automatically generated with medium confidence

Như vậy, cả 2 mảng sẽ có độ dài là 167 tương ứng với 167 thuộc tính có thể so sánh được

Hàm rowsquad được định nghĩa ở file title, nhận tham số là 1 đối tượng Squad, trả về mảng là giá trị các thuộc tính của đối tượng đó  


Thực hiện lặp qua từng đội,sau đó lặp qua từng thuộc tính của đội nếu thuộc tính hiện tại của đội này lớn hơn giá trị trong mảng biggest\_attr\_value tương ứng thì ta thực hiện thay đổi giá trị trong mảng biggest\_attr\_value đồng thời cũng thay đổi tên của đội có điểm cao nhất tại vị trí đó luôn

A screen shot of a computer code

Description automatically generated

Thêm 1 trường hợp xử lý nếu chưa có giá trị cao nhất nào được thiết lập trong mảng thì đội hiện tại sẽ

Cuối cùng thực hiện in kết quả

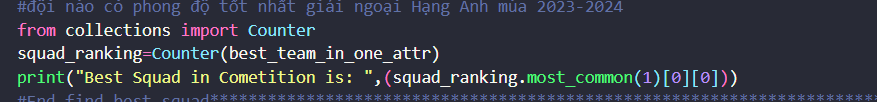
A computer code with colorful text

Description automatically generated with medium confidence

\*Đánh giá đội có phong độ tốt nhất mùa giải

Ta thực hiện đếm số lần mà đội đó xuất hiện trong mảng best\_team\_in\_one\_attr

Đội nào xuất hiện nhiều nhất thì có nghĩa là đội đó dẫn đầu nhiều mặt chỉ số nên có phong độ rất cao



Ở đây ta dùng Counter để đếm số lần mỗi đội xuẩt hiện, ta tìm được tên của đội có phong độ tốt nhất  


Kết quả cho thấy Manchester City có mặt ở nhiều chỉ số cao nhất, thực tế cũng chứng minh rằng đội bóng này là đội bóng vô địch trong mùa giải

## **3. Bài tập 3**

### **a) Các thư vận cần thiết**

Các thư viện cần dùng  
A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Ta sử dụng mảng đối tượng Player đã được lưu ở bài 1

Mảng list\_player chứa những cầu thủ chơi trên 90 phút

Đồng thời lấy luôn mảng giá trị trung bình đã được tính toán ở phần bài 2



### **b) Lọc và chuẩn hóa dữ liệu**

-Ta bắt đầu bằng cách lọc các thuộc tính của từng đối tượng Player rồi lưu vào 1 mảng, tại đây, các giá trị không hợp lệ của đối tượng ta sẽ thay chúng bằng giá trị trung bình của các thuộc tính dựa trên mảng các thuộc tính đã khai báo ở trên

-Các thuộc tính có giá trị âm sẽ sẽ loại bỏ(chỉ cần 1 thuộc tính nào đó của 1 cầu thủ có giá trị âm, thì giá trị đó sẽ bị loại bỏ không đưa vào vẽ biểu đồ bởi nó chủ yếu là các giá trị tính bằng 1 hay nhiều giá trị thuộc tính khác (Ví dụ: per90\_G+A tính bằng tổng per90\_Gls và per90\_Ast) nên chúng không quan trọng

A screenshot of a computer program

Description automatically generated

-Hàm normalizedata thực hiện điều này với mảng players\_attr\_list lưu các thuộc tính của cầu thủ sau khi sửa các giá trị N/a, mảng AttrToDelete lưu các thuộc tính sẽ bị xóa vì thuộc tính đó có cầu thủ có giá trị nhỏ hơn 0

-Các giá trị sẽ được lọc lại một lần nữa cuối cùng ra được mảng non\_negative\_data, mảng này sẽ được đem đi chuẩn hóa làm dữ liệu cho biểu đồ

-Dữ liệu được chuẩn hóa 2 lần

+ Lần đầu tiên là chuẩn hóa bằng log do dữ liệu có độ lệch tương đối lớn( số phút có thể lên đến hàng nghìn nhưng số trận chỉ đến hàng chục,số bàn thắng cũng chỉ đến hàng chục, 1 số dữ liệu khác chỉ đến hàng đơn vị), ta cộng thêm 1 số rất nhỏ(1e-18) do dữ liệu có chứa giá trị 0

+ Lần thứ 2 là chuẩn hóa bằng thuật toán MinMaxScaler, thuật toán đưa các giá trị trong dữ liệu về khoảng 0-1

Hàm normalizedata trả về dữ liệu sau khi xử lý, chuẩn hóa cho việc vẽ sơ đồ

### **c) Tìm số điểm hội tụ cho thuật toán Kmean**

Ta có thể thực hiện đánh giá bằng phương pháp Elbow, số điểm phân cụm được đánh giá tại nơi mà ta thấy rõ điểm khửu tay trên sơ đồ

A computer screen shot of a program code

Description automatically generated

A screen shot of a graph

Description automatically generated

Biểu đồ cho thấy gấp khúc ở đoạn 2 và 3

Ta có thể tiếp tục thử với phương pháp Silhouette

A screen shot of a graph

Description automatically generated

Dựa vào sơ đồ ta thấy điểm tại vị tri 2 đạt điểm cao nhất cho thấy sự tối ưu cao nhất nếu chia cụm thành 2 phần

\*Nhận xét: Dựa vào số phần phân cụm đã các định chia cầu thủ thành 2 nhóm, ta thấy rõ được sự hợp lý khi 1 phần là cầu thủ kém nổi bật, các cầu thủ còn lại là các cầu thủ có thành tích và kĩ năng vượt trội hơn các cầu thủ còn lại

### **d) Phân cụm Kmean**

-Thực hiện vẽ biểu đồ thông qua hàm drawKmean nhận 2 tham số là dữ liệu và số cụm muốn phân  
A computer code on a black background

Description automatically generated

A screenshot of a computer

Description automatically generated

Phân cụm bằng Kmean

-Ta chưa thể thấy được sự rõ ràng trong phân cụm do dữ liệu đầu vào có rất nhiều chiều(167 chiều tương ứng với 167 thuộc tính)

### **e) Sử dụng phương pháp giảm chiều của dữ liệu bằng thuật toán PCA**

A computer screen shot of a program

Description automatically generated

A screen shot of a graph

Description automatically generated

### **g) So sánh 2 cầu thủ với radarChart**

A computer screen shot of a program

Description automatically generated

Ở đây ta lấy thông tin về tên của cầu thủ cùng với đội bóng của cầu thủ đó thi đấu, nếu thông tin không hợp lệ, ta yêu cầu nhập lại

Do thuộc tính của các cầu thủ tương đối phức tạp để nhớ cũng như nhập đúng, ta đặt cho mỗi thuộc tính 1 id và yêu cầu người nhập liệu nhập id vào thay vì tên của thuộc tính

Khi nhập thuộc tính, dữ liệu nhập được đặt trong vòng lặp vô hạn cho đến khi nhập đúng id thì thôi, các id được in ra trước để người dùng có thể theo dõi và nhập khi cần

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Kết quả thu được từ dữ liệu đầu vào là 2 cầu thủ player1 và player2 cùng với đó là dãy các id thuộc tính cần so sánh được dùng làm tham số cho hàm drawRadarChart

\* drawRadarChart được định nghĩa trong file radarChartplot đã nói ở phần đầu tiên,file này chỉ chứa duy nhất 1 hàm dùng để vẽ radarchart với tham số đầu vào là 2 đối tượng Player và mảng id thuộc tính cần so sánh

- Trong bước tiền xử lý, ta sẽ xử lý bằng cách lấy ra giá trị của các thuộc tính của đối tượng Player p1 và p2, tiếp theo lấy giá trị của các id thuộc tính có trong mảng tham số đầu, với những giá trị không hợp lệ,ta đặt nó bằng 0, kết quả cuối cùng là mảng values\_player1 và values\_player2 là 2 mảng chứa giá trị của các thuộc tính yêu cầu.

A screen shot of a computer program

Description automatically generated

Hoàn thành việc lọc dữ liệu ta tiến hành vẽ biểu đồ

A screenshot of a computer program

Description automatically generated

**\*\*Demo kết quả:**

So sánh 2 cầu thủ Aaron Cresswell và Aaron Hickey ở các chỉ số có id 0,1,11,12,13,47,67,68,69,71 ta thu được biểu đồ như hình dưới

A screen shot of a computer screen

Description automatically generated

## **4. Bài tập 4**

### **a) Update file object**

- Do làm việc với 1 url khác, ta phải xây dựng 1 class đặc trưng cho việc chuyển nhượng cầu thủ

Do vậy, ta thêm vào trong file object.py một lớp nữa đặt tên là Tranfer là cơ sở để thực hiện đánh giá giá trị của cầu thủ sau này  
A screenshot of a computer program

Description automatically generated

- Lớp có các thành phần cơ bản là tên cầu thủ(name), giá trị chuyển nhượng của cầu thủ(price)

- Các giá trị còn lại là thành phần để dựa vào để định giá cầu thủ bao gồm:skill(kĩ năng), pot(tiềm năng),mathches(số ván chơi), minutes\_play(số phút đã chơi),goal(số bàn thắng), assists(kiến tạo) và các loại thẻ đỏ vàng đã nhận, trong đó quan trọng nhất đó chính là averageETV là khoảng giá trị ước tính chuyển nhượng trung bình của một cầu thủ đóng vai trò quan trọng trong việc định giá cầu thủ

- Lớp còn có 1 hàm setAttr để thiết lập các thuộc tính cho cầu thủ

-Thuộc tính statLink chỉ là dường dẫn đến trang thông tin chi tiết về số liệu cầu thủ để sau này có thế lấy data về

### **b) Thu thập thông tin cầu thủ**

Ta tạo thêm 1 file mới tên là tranferdata.py chỉ để lấy thông tin từ trang web

A screenshot of a computer

Description automatically generated

Trong file này ta thực hiện như sau:

-Do trang web có thể dùng hậu tố như ‘K,’M’,.. để thể hiện giá trị chuyển nhượng   
A screenshot of a computer

Description automatically generated

Ta cần 1 hàm để xử lý vấn đề này bằng cách biến đổi các kí hiệu đó thành 1 số cụ thể, hàm changePriceToNumber được ra đời để xử lý vấn đề này

A computer screen shot of a code

Description automatically generated

Hàm changePriceToNumber

-Do số lượng page khá lớn(18 trang) chứa thông tin về giá chuyển nhượng của các cầu thủ, ta sử dụng đa luồng để thực hiện lấy data một cách tối ưu nhất  
A screen shot of a computer program

Description automatically generated

-Các trang chứa data đều nằm trong thẻ có id=”player-table-body”, nên idTable sẽ là biến truyền vào hàm getTranferFromWeb lấy data sau này sẽ giải thích chi tiết  
-Số trang là 18, chạy mỗi luồng 3 trang như vậy sẽ có 6 luồng, kết quả được lưu vào mảng list\_result chứa các thành phần là đối tượng Tranfer

\*Hàm getTranferFromWeb nhận 3 tham số là mảng các trang cần duyệt,id của bảng data và mảng kết quả

Dùng vòng lặp để chạy hết các trang có trong mảng các trang cần duyệt lấy được các tr\_list chứa thông tin chuyển nhượng các cầu thủ có trong trang

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

Duyệt từng phần tử trong tr\_list, qua 1 vài bước xử lý, ta lấy được thông tin về tên, giá chuyển nhượng và dường dẫn đén trang chứa thông tin cầu thủ

-Sau khi có mảng kết quả, từ đường dẫn trong thuộc tính của đối tượng Tranfer, ta duyệt qua từng link để thiết lập các thuộc tính còn lại, việc này cũng được làm tương tự như trên, cũng chạy đa luồng với mỗi luồng chạy 85 trang thông qua hàm getStatPlayer nhận tham số là trang bắt đầu, trang kết thúc và mảng kết quả

-Cuối cùng lưu vào 1 file để sau này tiện làm việc mà không cần phải chạy lại những đường link trên để tránh mất thời gian cho mỗi lần khảo sát

A computer screen shot of text

Description automatically generated

Thu thập dữ liệu và lưu vào file tranfers.pkl

Ngoài ra, ta còn thu thập thêm thông tin về độ tuổi của các cầu thủ, ở đây ta tận dụng api của web trả về thay vì truy cập vào từng link gây tốn thời gian.

Thu thập thông tin về độ tuổi và player\_slug, player\_slug giống như id của cầu thủ, mỗi cầu thủ có 1 slug riêng, slug này giống với đuôi cuối trong đường dẫn đến bảng thông tin chi tiết về cầu thủ(Thuộc tính statLink của đối tượng Tranfer)

Từng cầu thủ được lưu theo cặp tuổi, slug, cuối cùng lưu vào file **slugAndAge.pkl** phục vụ cho phần dự đoán cầu thủ sau

A screen shot of a computer program

Description automatically generated

### **c) Đề xuất phương án định giá cầu thủ**

-Sau khi đã có file dữ liệu từ phần trước, ta thực hiện import vào trong file bt4.py, thực hiện thiết lập giá trị cho các mảng thuộc tính

A computer screen shot of a program code

Description automatically generated

Chuẩn bị các vector là thuộc tính của cầu thủ

A computer screen shot of a program code

Description automatically generated

Đồng thời ta cũng lưu giá trị các thuộc tính của từng cầu thủ vào trong mảng player để tiện cho việc test sau này  
A screen shot of a computer program

Description automatically generated

\*Thực hiện hồi quy tuyến tính để đề ra phương pháp định giá cầu thủ

A computer screen shot of a program code

Description automatically generated

Mảng y chính già mảng giá trị cầu thủ, Xbar là ma trận tạo bởi các giá trị thuộc tính của đồi tượng Tranfer, qua quá trình huấn luyện ta thu được kết quà cuối cùng.Sau đây là demo 1 số kết quả

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Có thể thấy có 1 số dữ liệu mô hình dự đoán gần đúng nhưng có một số thi độ lệch rất lớn hệ số xác định R^2 chỉ đạt giá trị cực hạn là 0.9431488656941758 nên mô hình có vẻ như dự đoán có sự sai khác khá lớn nhưng tổng quan thì vẫn có thể chấp nhận được

Ta có thể thực hiện kiểm tra xem độ lệch của mô hình như thế nào thông qua số lượng lệch ở từng mức

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Qua số liệu ta có thể đánh giá độ tin cậy cho phương pháp này