G.O.A.T. Evaluation Model Based on Analytic Hierarchy Process

IMMC#21510726

Abstract

Nowadays we watch a variety of sports and athletes like Novak Djokovic, Rafael Nadal, etc are really well known among us. Therefore we did this research to find out the G.O.A.T.-*Greatest Of All Time* in one particular sport.

For the 2018 Tennis Grand Slam women's singles, we determined 8 factors which may influence the result of the selection of G.O.A.T. and used AHP-*Analytic Hierarchy Process* model to calculate the weight of each factor. Then we used the evaluation model to find out the score of each athlete and came up with a ranking table, the result shows that Kerber is the G.O.A.T. player of 2018 Grand Slam women's singles.

We also set up another model using the evaluation model. 4 factors are provided in this model to find out the G.O.A.T. player of badminton men's singles and we found out their weight using the AHP model. In this model we collected the athletes' information instead of the information about the competitions. Three most authoritative men's singles badminton competitions are chosen. The result shows that the G.O.A.T player is Lin Dan.

In this study we found out the G.O.A.T. player for 2018 Tennis Grand Slam women's singles and badminton men's singles. through this article we could provide the best way to determine the G.O.A.T. player of one particular sport.

Keywords: Analytic Hierarchy Process, G.O.A.T

Contents

1 Introduction

Sports like tennis and basketball usually appear on our televisions, therefore athletes like Novak Djokovic, Rafael Nadal, etc are really well known among us. So this creates a problem: Who is the best? In this research we will come up with a model that would be able to determine the G.O.A.T - *Greatest Of All Time* in one particular sport.

There are two kinds of sports: Individual sports and Team sports. Individual sports are played by individuals, and the winners are individuals rather than groups. Individual sports could be split into two kinds: Sports with physical contact, such as boxing, wrestling, etc.The others are sports without physical contact, such as tennis, badminton and so on. Team sports are played by teams and the winners are teams instead of individuals. Football, basketball, etc are examples of such sports.

In the first part of the research, we will set up a model to determine the G.O.A.T of 2018 single women Grand Slam. We will use the evaluation model to evaluate the result of 2018 Grand Slam and find out the greatest woman player for that season only.

In the second part of the research, we selected c badminton men's singles as the target individual sport. We chose our indexes depending on the data we could obtain and data that is able to measure. We selected BWF World Championships, All England Open and Olympic Games as the data sources because they are the most authoritative men's badminton single competitions. After setting up of the model, we will also adjust the model to make it suitable for other individual sports.

For the third part of the research, we will also adjust the model above in order to enable it to find out the G.O.A.T. for any team sport.

2 Assumptions and Variables

2.1 Assumptions

- **Assumption 1:** The factors we chose can fully reflect the capability and the level of the athletes.And these factors influence the evaluation significantly.
- **Assumption 2:** Part of the information collected is from the list provided by IMMC, while the other part is from official website about woman tennis. The data obtained is reliable.
- **Assumption 3:** In the model, we do not consider the age of the players for the players are all in a constant range of age. There will not be many differences between the age of players.

2.2 Variables

Table 1: Variables Table

Variables	Description
k	random factor
i	random factor
j	random factor
a_{ik}	the importance of i compared with k
CI	coincidence indicator
RI	random indicator
CR	coincidence rate
λ_{max}	maximum eigenvalue

3 Modeling and Results for Women's tennis singles

In chapter 3, we are going to use AHP-*Analytic Hierarchy Process* model[**?**] and evaluation model to find out the G.O.A.T. for singles women's tennis in 2018 Grand Slam.

3.1 Factors

Firstly, we consider the best woman tennis player in 2018 model mainly from three aspects, including background, performance during competitions and final result. Background(show the past of the player). Performance during competitions(directly show the strength of a player). Final result(most simple way to show a player's ranking,essential while evaluating).

Secondly, the background consists of three factors, country,seed ranking number,competitions attended in 2018. Country is the number of woman tennis champions in one country. Seed ranking number is the ranking made before the competition based on a player's scores before, while competitions attended in 2018 is the number of contests a player joined among the four races in 2018. The performance during competitions contains points achieved in all and average rounds. Points in all is the points a player obtained in all her rounds. Average rounds are the total rounds divided by the total competitions. If the average rounds are smaller, it shows that the player can beat the enemy in fewer rounds. Final ranking is the official output, it has great influence on judging who is the best player. The average difference of points per round show the difference between players. If a player gets minus number in all, it shows that she is worse than other ones. It is the opposite for players get the highest number.

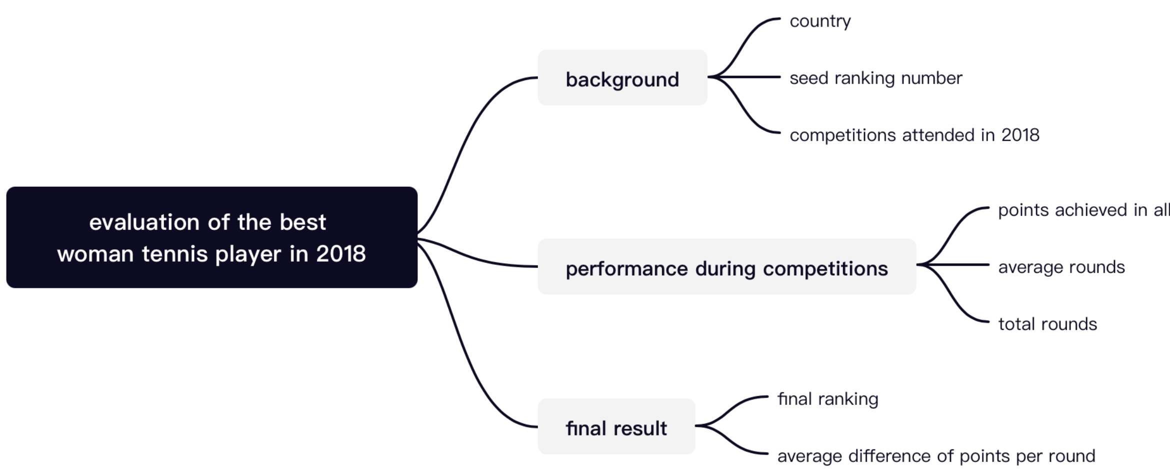

Figure 1: factors of the greatest woman tennis player in 2018

The factors are divided into three three bigger categories: background, performance and result. The reason for choosing these factors are mentioned in the assumption. The factors are categorized to make the model building less sophisticated.In this case, a big model can be divided into two simplier ones.

3.2 AHP Model

To get the equation of the greatest woman athlete in 2018, it is necessary to know how much the factors influence the result, which is the weight of the factors. We choose to use the AHP to calculate the weight, it is because there are not many factors(less than nine) so the results are probably accurate using AHP.

1.At the beginning, we make a list to compare the importance between factors. The list includes the importance of factors in comparison, which is divided into groups of two.The importance ranges from 1 to 9. We use a_ij to represent the importance of i compared with j.

$$a_{ik} * a_{kj} = a_{ij} \tag{1}$$

2.We calculate the maximum eigenvalue and eigenvector of judgment matrix.Then we normalize the eigenvector to obtain the weight.

3.After making the list, we need to prove that the list is reliable. Firstly, we calculate the coincidence indicatorCI using equation (2).

$$CI = (\lambda_{max} - n)/(n - 1) \tag{2}$$

Then, we need to know RI. It can be achieved through a specific list. After that, we need to calculate CR using equation (3).

$$CR = CI/RI \tag{3}$$

if CR < 0.1, then the list is reliable. After that, we use Eigenvalue method to calculate the weights of factors ranging from 0 to 1.

Table 2: Weight Table

Variables	Weight
average rounds	0.104
total score	0.108
total round	0.107
seed	0.148
country	0.132
overall result	0.137
attending time	0.114
score difference	0.150

3.3 Data processing

We mainly use the normalization method when we process the data, for tennis we have factors like credits and rounds number,etc, the more these factors are, the better one athlete is. For these data, we used

$$x_{new} = \frac{x}{x_{max}} \tag{4}$$

to calculate their normalization result, x represents the data of each athlete.

Some other factors are the opposite, just like the ranking of the seed, the smaller the ranking is, the better the athlete is. We used

$$x_{new} = 1 - \frac{x}{x_{max}} \tag{5}$$

to calculate the normalization result.

Factors like the gap of the score will sometimes be negative and therefore we used

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \times 2 - 1 \tag{6}$$

to calculate the result.

3.4 Ranking Results and Analysis

As a result, we have Angelique Kerber at the first place, Serena Williams as the runner-up and Madison Keys as the second runner-up. The information of the top 3 is listed in the business card below.

Figure 2: Information of Angelique Kerber

Figure 3: Information of Madison Keys

Figure 4: Information of Serena Williams

There were in total 40 athletes taking part in the competition of 2018 Tennis Grand Slam women's singles, their ranking are in the table below. The scores in the table are obtained by multiplying the weight of each index by the normalized data and then add them together.

Kerber has a high credit after the normalization, which is 0.96, this is because she not only won the champion, she was also the top 4 of France Open and runner-up of Australia Open. Because she competed in three of the four major competitions and placed well in all of them, her final score became the first which is also what we expected.

Though using our model we have Kerber as the G.O.A.T player, it might not be admitted by the public due to the fact that we only analyzed the data for 2018 Grand Slam. Maybe if we collects the data from 2000 to 2020, we might receive a different result.

Therefore the G.O.A.T. player of 2018 Tennis Grand Slam is Angelique Kerber.

Table 3: Ranking of 2018 Tennis Grand Slam women's singles

Name	Score	Ranking	Name	Score	Ranking
Kerber	1.154024161	1	Ostapenko	0.471164951	21
Williams	1.151384535	2	Barty	0.449370144	22
Keys	0.991050069	3	Svitolina	0.443580305	23
Halep	0.961409074	4	Tsurenko	0.363998001	24
Stephens	0.781751347	5	Rybarikova	0.354769156	25
Navarro	0.695742577	6	Muguruza	0.34764468	26
Wozniacki	0.672470294	7	Garcia	0.343437727	27
Cibulkova	0.642183829	8	Giorgi	0.336882758	28
Pliskova	0.621289615	9	Hsieh	0.332995136	29
Kasatkina	0.60068995	10	Putintseva	0.305018019	30
Kontaveit	0.57744653	11	Vondrousova	0.26875992	31
Strycova	0.567564123	12	Kanepi	0.248398943	32
Sevastova	0.523451098	13	Uytvanck	0.248123907	33
Sabalenka	0.522438966	14	Martic	0.233145458	34
Osaka	0.511497426	15	Bencic	0.223865466	35
Bertens	0.498901032	16	Mertens	0.217748764	36
Gorges	0.497575385	17	Makarova	0.214585473	37
Rondina	0.487584771	18	Sasnovich	0.201428461	38
Buzarnescu	0.477158542	19	Vekic	0.199627469	39
Sharapova	0.473686888	20	Allertova	0.184669466	40

4 Modeling and results for Men's badminton singles

We chose badminton as the individual sport because it is a really popular sport and nearly everyone knows how to play it. We selected top 6 well known badminton player and researched their winning records instead of collecting the information of each competition.

4.1 Factor

Firstly, we consider the G.O.A.T. model mainly from three aspects, which is the same as the evaluation of the greatest woman tennis player in 2018, including background, performance during competitions and final result. Background(show the past of the player). Performance during competitions(directly show the strength of a player). Final result(most simple way to show a player's ranking,essential while evaluating).

Secondly, the background consists of three factors, country,career length,competitions attended in history. Country is the number of badminton champions in one country. career length and competitions both show the experience of a player. the final results includes the final ranking in BWF World Championships, Summer Olympic Games and All England Open indicate the scores a player achieved in history. The performance during competition contains rate of victory. It shows that if the player is usually winning, which is important when evaluating.

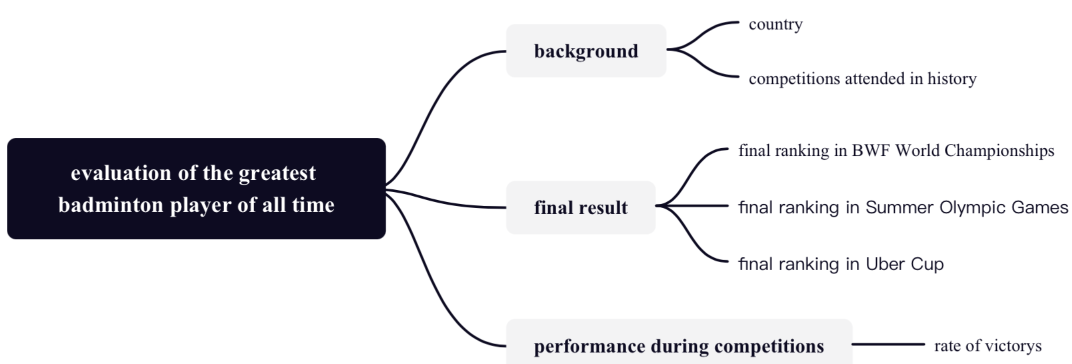

Figure 5: factors of the greatest badminton player

After using the AHP method, we could get the weight result in Table 4 and the pie chart of weight is showed in Figure 6.

Table 4: Weight Table

Variables	Weight
total round	0.151
rate of victory	0.568
competition attended in history	0.06
final result	0.221

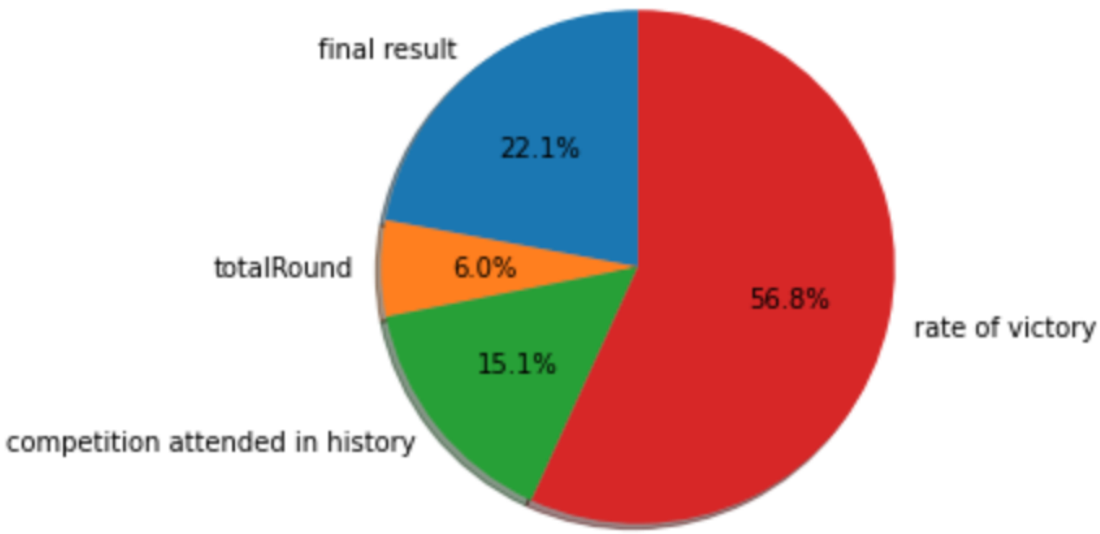

Figure 6: weight of pie chart for badminton

4.2 Candidates information

Below are the top 6 men's badminton athletes in the world which are the G.O.A.T. player candidates.

Figure 7: Information of Lin Dan

Figure 8: Information of Chen Long

Figure 9: Information of Lee Chong Wei

Figure 10: Information of Peter Gade

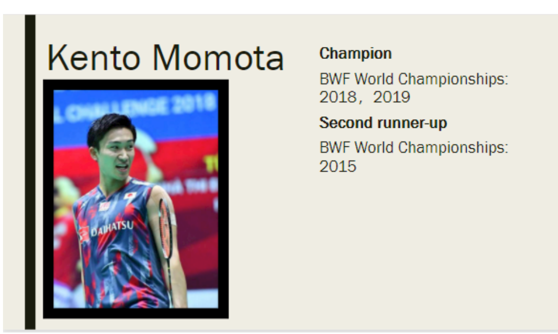

Figure 11: Information of Kento Momota

Figure 12: Information of Taufik Hidayat

4.3 Ranking Results and Analysis

The result in the table below shows that Lin Dan is the G.O.A.T. player for men's badminton singles. The first 4 column shows the normalized numbers for credits, total games career length and win rate, using the same method as above, we multiplying the weight of each index by the normalized data and then add them together to get the second last column of the table, which is he final score and the last column contains the ranking of the final score from the largest to the smallest.

The result was the same as what we expected to see because the score Lin Dan has in the first 4 column are higher than others which clearly shows that he must be the G.O.A.T. player in men's badminton single. Though someone might disagree with the result and might think Taufik Hidayat is the best. However the data we found ends till 2020, so due to the data we have now, Lin Dan is still the G.O.A.T. player of men's singles badminton

Table 5: Final Ranking

	Credits	Total games	Career length	Win rate	Final score	Ranking
Lin Dan	0.2208779	0.052442347	0.1506199	0.56822363	0.992163777	1
Lee Chong Wei	0.081724823	0.06027856	0.143088905	0.556859157	0.841951445	2
Chen Long	0.072889707	0.034961565	0.10543393	0.534130212	0.747415414	3
Taufik Hidayat	0.08835116	0.047620062	0.10543393	0.500036794	0.741441947	4
Peter Gade	0.053010696	0.033153208	0.143088905	0.482990085	0.712242895	5
Momota Kento	0.037549243	0.022905853	0.04518597	0.52276574	0.628406805	6

5 Modeling and Results for Team Sports

5.1 Factor

So as to fit the demand of evaluating a player in a team sport, we add one extra standard of evaluation, which is the cooperation skills of a player under the main index of the player performance. In team sports, not only a player's performance is important, but it is also important to cooperate so as to score for the team as a whole. A factor we considered to be very important is how the players understand his mates. For example, in those sports like diving, the fitness of the two players are one of the main factor even in the current evaluation system. This can be measured through the score of given by the coach during the competition. For other team sport like the football or the basketball, times of cooperation shows a lot of in terms of the cooperation skills of a player, this can also be measured by the index 'passing ball' for every competition. For the background, information, we also decide to add the measure index of the history of playing for players. In fact, it is easier to cooperate with someone that the player knows very will than cooperate with someone who is the first time to cooperate with. Therefore, it is not surprising that the player who get a good cooperation index even when it is the first time when they cooperate is greater. Thus, the history of two players cooperating is important.

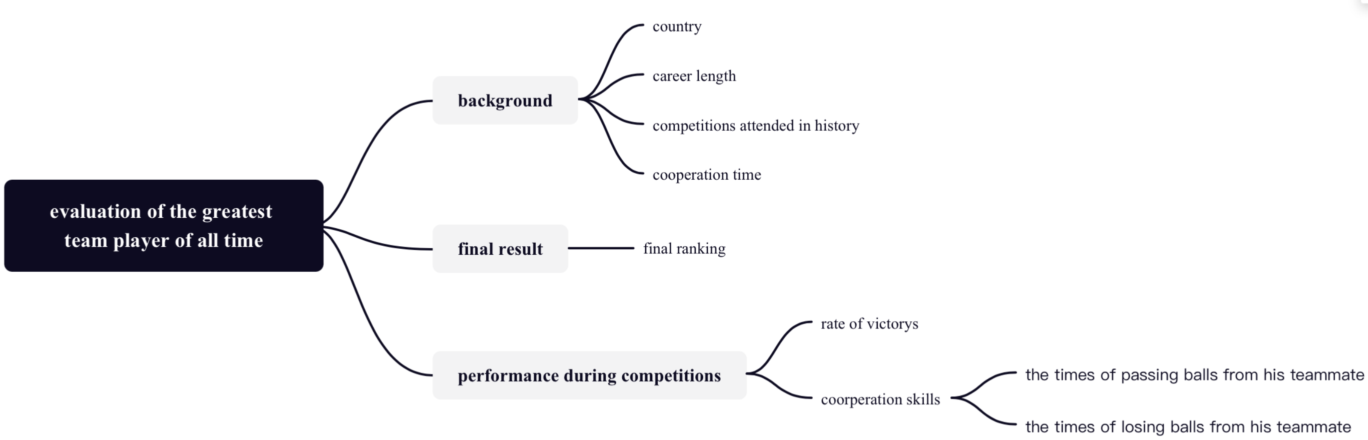

Figure 13: factors in team sport.png

6 Strengths and Weaknesses

6.1 Strengths

1.We can not only use the model to solve the problem of selecting the best woman tennis player in 2018, but also solve any evaluation of individual athlete as long as there is correct data.

2.The data can be easily found in official websites, so it is simple to select the best player in any sport.

6.2 Weaknesses

1.The model is not completely objective because of the method we choose. We choose to use the AHP. It depends on our personal minds to decide which factor is more important briefly. Then we examine if the model is reliable. In this way, our model may not be accurate when the data is much larger.

2.There is one factor which may be influential that our model does not include. That is the year of an athlete get a prize. It may be important as the difficulty is different in different times. The reason is that as the theory and training skills of sports improve, it is easier for younger athletes to win prizes. In this case, it is unfair for those who join earlier competitions. If there is enough time, we can divide the prizes into different categories based on time to get a more accurate result. In our model, we do not consider this as a factor.

Letter

Dear Director,

Thank you for inviting us to build a model in order to find out the G.O.A.T.- Greatest Of All time for each individual sport. To make it easier to understand, we will use man's singles badminton as an example to explain our model.

In order to determine the best player, we researched the top 6 famous badminton player (all male) and recorded their winning records. Factors are also needed. Therefore we firstly thought from 3 aspects: background, final result and performance during competitions. Which could help us determine the factors. Then we did some further study and came up with 4 key factors: country, rate of victory, competitions attended in history and career length. The reason why we choose these factors is because these are available data and they could be changed into numbers that are easy to calculate.

After the selection of factors, we used AHP-Analytic Hierarchy Process model to calculate the weight of each factor. The more important a factor is, the more the weight is. Then we used evaluation model to calculate the G.O.A.T. player. The result shows that Lin Dan is the G.O.A.T. player of badminton men's singles because after the calculation of weight for each factor, the total score he earned is 0.992 which is in the first place.

The way to use our model is pretty easy, you just need to determine an individual sport and find out the factor that may influence the result and put them into the model to do the calculation and the G.O.A.T. player of that sport will be found out.

The advantage of our model is that when you want to know the best athletes in a sport, you don't have to compile all the data for that sport, you just have to collect information about the athletes to get the results.

Above is the model we want to introduce and thank you again for inviting us to set up the model to find G.O.A.T.,hope you find it useful!

Yours sincerely,

Team#21510726

Appendix

```
import pandas as pd
import numpy as np

class Read:
    def __init__(self, path, name):
        self.path = path # file path
        self.name = name # player's name
        self.data = pd.read_excel(self.path)  # read the data
        self.totalscore = 0 # set initial total score
        self.ScoreList = []
        self.totalDiffer = 0



    def FilterData(self,data,index):  # convert str to list
        return [int(i) for i in data[index][1:-1].replace("]","").replace(
            "[","").split(",")]

    def Decode(self): # find data related to the player
        return self.data[(self.data['Player1'] == self.name) | (self.data[
            'Player2'] == self.name)]

    def Average(self): #
        return np.mean(Read(self.path, self.name).Decode()['GamePerRound'
            ])

    def OneScore(self, PlayerName, index):
        scoreList = self.FilterData(self.Decode()['Score'].tolist(), index
            )
        if PlayerName == self.Decode()["Player1"].iloc[index]:
            if len(scoreList) ==4:
                return scoreList[0] + scoreList[2]
            else:
                return scoreList[0] + scoreList[2] + scoreList[4]
        else:
            if len(scoreList) ==4:
                return scoreList[1] + scoreList[3]
            else:
                return scoreList[1] + scoreList[3] + scoreList[5]
```

```
    def OneDiffer(self, PlayerName, index):
        ScoreList = self.FilterData(self.Decode()['Score'].tolist(), index)
        if PlayerName == self.Decode()["Player1"].iloc[index]:
            if len(ScoreList) == 4:
                return ScoreList[0] - ScoreList[1] + ScoreList[2] - ScoreList[3]
            else:
                return ScoreList[0] - ScoreList[1] + ScoreList[2] - ScoreList[3] + ScoreList[4] - ScoreList[5]
        else:
            if len(ScoreList) == 4:
                return ScoreList[1] - ScoreList[0] + ScoreList[3] - ScoreList[2]
            else:
                return ScoreList[1] - ScoreList[0] + ScoreList[3] - ScoreList[2] + ScoreList[5] - ScoreList[4]


    def TotalScore(self):
        for i in range(len(self.Decode()['Player1'].tolist())):
            self.totalscore += self.OneScore(self.name, i)
        return self.totalscore

    def TotalDiffer(self):
        for i in range(len(self.Decode()['Player1'].tolist())):
            self.totalDiffer += self.OneDiffer(self.name, i)
        return self.totalDiffer

    def checkWinCondition(self):
        return True

    def checkAttendTime(self):
        return (len(self.Decode()["Player1"]) + len(self.Decode()["Player2"])) / 2

    def Grade(self, State):
        if State == "FourthRound":
            if checkWinCondition():
```

```
                return 4
            else:
                return 3

        elif State == "Quarterfinals":
            if checkWinCondition():
                return 3
            else:
                return 2

        elif State == "Semifinals":
            if checkWinCondition():
                return 2
            else:
                return 1

        elif State == "Final":
            if checkWinCondition():
                return 1
            else:
                return 0

    def FinalRating(self):
        name = self.Decode()["Round"]
        return Grade(name)

    def GetAttendenceCondition(self):
        self.nameList.append(self.Decode()["Competition"])
        return len(list(set(self.nameList)))



if __name__ == "__main__":
    R = Read("data.xlsx", "Mertens")
    c = R.TotalScore()
    print(c)
```

```
import numpy as np

'''
```

```
                         AvgRound|totalScore|TotalRound|Seed|Country|
                            Overall Result|AttendingTime|ScoreDifference
        Avg Round            1       4/5        9/10    15/10    3/5
           5/3           7/10           8/10
        totalScore           5/4     1          10/11   14/10    3/10
           14/10         8/10           7/10
        TotalRound           10/9    11/10      1       14/10    5/10
           13/10         8/10           6/10
        Seed                 10/15   10/14      10/19   1        19/10
           26/10         15/10          12/10
        Country              5/3     10/3       10/3    10/19    1
           3/10          7/10           3/10
        Overall Result       3/5     10/14      10/13   10/26    10/3       1
                        15/10          13/10
        AttendingTime        10/7    10/8       10/8    10/15    10/7
           10/15         1              6/10
        ScoreDifference      8/10    10/6       10/6    10/12    10/3
           10/13         10/6           1

'''
'''
A = [[1,5,2,0.2],
    [0.2,1,0.25,0.2],
    [0.5,3,0.8,0.4],
    [3.5,6.5,4.5,1]]
'''
A = [[1,4/5,9/10,15/10,3/5,4/3,6/10,6/10],
     [5/4,1,10/11,14/10,4/10,13/10,9/10,6/10],
     [10/9,9/10,1,14/10,5/10,13/10,8/10,6/10],
     [10/15,10/14,10/16,1,19/10,26/10,15/10,12/10],
     [5/3,10/3,8/3,10/19,1,3/10,6/10,3/10],
     [3/5,10/14,10/13,10/26,10/3,1,15/10,13/10],
     [10/7,10/8,10/8,12/15,10/7,10/15,1,6/10],
     [8/10,10/6,10/15,10/12,10/3,10/13,10/6,1]]

lamb,v = np.linalg.eig(A)
print(lamb)

lambda_max = max(abs(lamb))
loc = np.where(lamb == lambda_max)
```

```
weight = abs(v[0:len(A),loc[0][0]])
weight = weight/sum(weight)
print(weight)
```

```
import numpy as np
lamb,v = np.linalg.eig(A)
lambda_max = max(abs(lamb))
loc = np.where(lamb == lambda_max)
weight = abs(v[0:len(A),loc[0][0]])
weight = weight/sum(weight)
RI_list = [0 ,0 ,0.58,0.9,1.12,1.24,1.32,1.41,1.45,1.46]
RI = RI_list[len(A)-1]
CI = (lambda_max - len(A))/(len(A)-1)
CR = CI / RI
print('lambda_max=',lambda_max)
print('w=',weight)
print('CI=',CI)
print('RI=',RI)
print('CR=',CR)
```

```
import matplotlib.pyplot as plt

labels = 'AvgRound','totalScore','TotalRound','Seed','Country','Overall␣
   Result','AttendingTime','ScoreDifference'
sizes =
   [0.1039187710391877,0.10805948108059481,0.1065222210652222,0.1477181514771815,0.132


0.1368856513688565,0.11411126114111261,0.1504672715046727]
explode = [0 for i in range(len(labels))]

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')

plt.show()
```

```
import matplotlib.pyplot as plt
```

```
labels = 'score','totalRound','playtime','rate␣of␣win'
sizes = [0.2208779, 0.06027856, 0.1506199, 0.56822363]

explode = [0 for i in range(len(labels))]

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')

plt.show()
```