**About code**

Please change the index18 and result18 file location as per your requirement. Means give proper location of the files.

f=pd.read\_csv('C:/Users/ASUS/Desktop/upgrad/2018wc/result18.csv',',',)

and

v=pd.read\_csv('C:/Users/ASUS/Desktop/upgrad/2018wc/index18.csv',',')

**Data selection**

We use two data tables.

1st:-FIFA ranking of all Teams with there points as per FIFA.

Link:-

<https://www.fifa.com/fifa-world-ranking/ranking-table/men/index.html>

Excel file name:-Index18

2nd :-All national football match results of last 4 years including fifawc18 group stage results

Link:-

<https://www.kaggle.com/martj42/international-football-results-from-1872-to-2017/version/32#_=_>

and

<https://www.eurosport.com/football/world-cup/2018/world-cup-2018-final-group-standings-results-and-last-16-draw-fixtures_sto6815861/story.shtml>

Also knockout match table

link:- <https://www.google.com/search?q=worldcup+2018+results&rlz=1C1GIGM_enIN793IN793&oq=worldcup+2018+results&aqs=chrome..69i57j0l5.19283j1j4&sourceid=chrome&ie=UTF-8#sie=lg;/m/06qjc4;2;/m/030q7;mt;fp;1>

**Data cleaning process: -**

In case of index18 file

1st convert the data in excel format by below link

<https://convertio.co/document-converter/>

after that clean the data using replace, find, Remove duplicate data, Text to columns, filter etc.

For changing the point range of every teams, Point divided by 1000.

After that change this excel file to csv file.

In case of result18 file

Filter and select last 4 years of data.

Add world cup 2018 data into it. Find the winning team, loser team or TIE by goals. Add this data in new columns.

**Visualizations and Modelling**

We visualize the data in from of numbers. In Excel if Team1 Goals are greater than Team2 Goals Team1 is winning team if equal then TIE or if less team2 is winner. In Python we calculate a point for every teams which teams point is greater that teams wins against other team.

We use panda and numpy in Python. At first create a list of all countries playing in fifa worldcup 2018,grouping by team1 and team2 for getting total number of goal givens by a team. From that we evaluate the total goal diff as (gd).

Now create a for loop calling every team as a item one by one for calcutating there points. If any team is winning any match against another team we give him points basis of the loser teams ranking and point in index18 chart.Index18 is taken from fifa ranking given by FIFA. IF any team wins against a top-ranking team we give him more points as per the index18 chart. Same logic for draw. For losing a game by a team we decide a parameter basis of winning team, If losing against a team which is in top losing parameter(lp) is less. We have added all winning ,TIE and losing points as p,dm and lp.

Example:-

Spain wins against Belgium (point- 1.298) and Egypt(point- 0.649)

p=1.298+ 0.649 {in code formula is p=p+(h[j][2]))

Spain TIE against Belgium (point- 1.298) and Sweden (point- 0.88)

dm=1.298+0.88 {in code formula is dm=dm+h[j][2]}

Spain lose against France (point- 1.198) and Sweden (point- 0.88)

Lp=1.5 - 1.198+ 1.5- 0.88

{1.5 taken as a range as Germany is in top in rank and his point is almost equle to 1.5

In code formula is lp=lp+1.5-(h[j][2])}

In the end we calculate total points for wining,TIE and losing. We multiply 6 for winning ,2 for TIE , 1 for loss and divided 10 from goal difference. Also divided the total point by total number of matches played by that team for get a avg.

Code:-

Point[i1]= round(((dm\*2+p\*6-lp+(gd/10))/len(play)),2 )

Save this data as a dictionary format.

Now we have all 32 teams along with there points in Point\_new dictionary.We sort the teams by there points.

OrderedDict([('SaudiArabia', 0.55), ('Panama', 0.86), ('Tunisia', 1.07), ('Egypt', 1.17), ('Nigeria', 1.2), ('CostaRica', 1.24), ('SouthKorea', 1.26), ('Iceland', 1.3), ('Serbia', 1.3), ('Morocco', 1.32), ('Australia', 1.33), ('Russia', 1.33), ('Japan', 1.45), ('Senegal', 1.66), ('Iran', 1.77), ('Sweden', 1.88), ('Switzerland', 2.27), ('Denmark', 2.29), ('Uruguay', 2.31), ('Poland', 2.31), ('Mexico', 2.38), ('Peru', 2.49), ('Croatia', 2.51), ('Colombia', 2.62), ('Germany', 2.73), ('Portugal', 2.79), ('England', 3.01), ('Argentina', 3.04), ('Belgium', 3.14), ('Spain', 3.23), ('France', 3.73), ('Brazil', 4.19)])

If point if higher winning chance is higher.

**Results:-**

We already get round of 16 teams. Using for loop we compare each teams by there points and get the semi-finalist ,finalist and champion.

***Our Semi-finalist are***

***['France', 'Brazil', 'Spain', 'England']***

***Our finalist are***

***['Brazil', 'Spain']***

***2018 worldcup champion is***

***Brazil***

**Code and Output: -**

import pandas as pd

import numpy as np

from collections import OrderedDict

f=pd.read\_csv('C:/Users/ASUS/Desktop/upgrad/2018wc/result18.csv',',',)

v=pd.read\_csv('C:/Users/ASUS/Desktop/upgrad/2018wc/index18.csv',',')

h=np.array(v)

Point={}

#dic={"Brazil":2,"England":3,"Portugal":4,"France":1,"Iran":4,"Sweden":4,"Denmark":7,"Australia":4,"Russia":4,"Paraguay":4,"Colombia":4}

list1=["Egypt","Spain","Belgium","Brazil","England","Portugal","France","Iran","Sweden","Denmark","Australia","Russia","Colombia","Morocco","Mexico","Uruguay","Germany","SaudiArabia","Morocco","Peru","Croatia","Nigeria","Iceland","Argentina","Switzerland","Serbia","CostaRica","SouthKorea","Tunisia","Panama","Japan","Senegal","Poland"]

groupingsasTeam1 = f.groupby(['Team1']) [['Team1Goals','Team2Goals']].sum()

#print(groupingsasTeam1)

groupingsasTeam2 = f.groupby(['Team2']) [['Team1Goals','Team2Goals']].sum()

#print(groupingsasTeam2)

for item in list1:

play=f[f.Team2==item]+f[f.Team1==item]

# print("number of play matchs of",item,len(play))

draw=f[f.Winner==item]+f[f.Losser==item]

d=len(play)-len(draw)

# print("number of draw matchs of",item,d)

win=f[f.Winner==item]

dm=0

#print("number of Win matchs of",item,len(win))

los=f[f.Losser==item]

p=1

# print("number of losses matchs of",item,len(los))

i1=item

hoo=f[f.Winner==f.Losser]

ho1=hoo[hoo.Team1==item]

ho=hoo[hoo.Team2==item]

listofdraw=list(ho.Team1)+list(ho1.Team2)

# print(listofdraw)

# Point[i1]= round((len(win)\*6+y\*2)/len(play),2 )

goalsgiven=groupingsasTeam1.Team1Goals[groupingsasTeam1.index==item] + groupingsasTeam2.Team2Goals[groupingsasTeam2.index==item]

# print(goalsgiven.dtypes)

# goalsgiven=goalsgiven.astype('int')

# print(goalsgiven.dtypes)

# print(goalsgiven+4)

goalsagainst=groupingsasTeam1.Team2Goals[groupingsasTeam1.index==item] + groupingsasTeam2.Team1Goals[groupingsasTeam2.index==item]

goaldiff=goalsgiven-goalsagainst

# print("number of goaldiff of",goaldiff)

l=list(win.Losser)

lost=list(los.Winner)

gd=goaldiff.item()

lp=0

u=t=y=0

for i in range(0,len(l)):

for j in range(0,206):

if (l[i]==(h[j][1])):

p=p+(h[j][2])

for i in range(0,len(listofdraw)):

for j in range(0,206):

if (listofdraw[i]==(h[j][1])):

dm=dm+h[j][2]

for i in range(0,len(lost)):

for j in range(0,206):

if (lost[i]==(h[j][1])):

lp=lp+1.5-(h[j][2])

# print(h[j][1])

Point[i1]= round(((dm\*2+p\*6-lp+(gd/10))/len(play)),2 )

#print(Point)

Point\_new = OrderedDict(sorted(Point.items(), key=lambda x: x[1]))

print(Point\_new)

tndg1=[]

sndg1= []

tndg2=[]

sndg2= []

group1=["Uruguay","Portugal","France","Argentina","Brazil","Mexico","Belgium","Japan"]

group2=["Spain","Russia","Croatia","Denmark","Sweden","Switzerland","Colombia","England"]

for i in range(0,8,2):

if Point\_new[group1[i]] > Point\_new[group1[i+1]]:

sndg1.append(group1[i])

else:

sndg1.append(group1[i+1])

for i in range(0,4,2):

if Point\_new[sndg1[i]] > Point\_new[sndg1[i+1]]:

tndg1.append(sndg1[i])

else:

tndg1.append(sndg1[i+1])

for i in range(0,8,2):

if Point\_new[group2[i]] > Point\_new[group2[i+1]]:

sndg2.append(group2[i])

else:

sndg2.append(group2[i+1])

for i in range(0,4,2):

if Point\_new[sndg2[i]] > Point\_new[sndg2[i+1]]:

tndg2.append(sndg2[i])

else:

tndg2.append(sndg2[i+1])

print("\n After round of 16 winning teams are:\n",sndg1+sndg2)

print("\n After round of 8 winning teams are:\n",tndg1+tndg2)

print("\n\n Our Semi-finalist are \n",tndg1+tndg2)

semi=tndg1+tndg2

final=[]

for i in range(0,4,2):

if Point\_new[semi[i]] > Point\_new[semi[i+1]]:

final.append(semi[i])

else:

final.append(semi[i+1])

print("\n\n Our finalist are \n ",final)

if Point\_new[final[0]] > Point\_new[final[1]]:

print("\n 2018 worldcup champion is\n ",final[0])

else:

print("\n 2018 worldcup champion is\n ",final[1])

#from heapq import nlargest

#two\_largest = nlargest(2, Point\_new, key=Point\_new.get)

#print("As this teams are not intersect before semi-final so 2 semi-finalists are",two\_largest)

**OUTPUT:-**

runfile('C:/Users/ASUS/Desktop/upgradwc2018.py', wdir='C:/Users/ASUS/Desktop')

OrderedDict([('SaudiArabia', 0.55), ('Panama', 0.86), ('Tunisia', 1.07), ('Egypt', 1.17), ('Nigeria', 1.2), ('CostaRica', 1.24), ('SouthKorea', 1.26), ('Iceland', 1.3), ('Serbia', 1.3), ('Morocco', 1.32), ('Australia', 1.33), ('Russia', 1.33), ('Japan', 1.45), ('Senegal', 1.66), ('Iran', 1.77), ('Sweden', 1.88), ('Switzerland', 2.27), ('Denmark', 2.29), ('Uruguay', 2.31), ('Poland', 2.31), ('Mexico', 2.38), ('Peru', 2.49), ('Croatia', 2.51), ('Colombia', 2.62), ('Germany', 2.73), ('Portugal', 2.79), ('England', 3.01), ('Argentina', 3.04), ('Belgium', 3.14), ('Spain', 3.23), ('France', 3.73), ('Brazil', 4.19)])

After round of 16 winning teams are:

['Portugal', 'France', 'Brazil', 'Belgium', 'Spain', 'Croatia', 'Switzerland', 'England']

After round of 8 winning teams are:

['France', 'Brazil', 'Spain', 'England']

Our Semi-finalist are

['France', 'Brazil', 'Spain', 'England']

Our finalist are

['Brazil', 'Spain']

2018 worldcup champion is

Brazil