빅데이터 프로그래밍

pasted-image.tiff프로젝트 결과보고서

컴퓨터과학과

201111235 신동규

1.기존 개발 기획

데이터를 총 3가지 분류로 가져올 예정입니다. 첫번째는 팀의 5년간 승률 및 평균스코어

두번째는 상대팀과의 최근 5년간의 전적 의 평균스코어와 승률 , 세번째는 팀선수의 개개인의 상대팀 포메이션 별 평점을 가져올 예정이다.

데이터는 <http://www.scoreboard.com> 혹은 naver sport 에서 팀별 데이터들을 가져올 예정이다.

또한 세번째 데이터를 위해 예를들어 손흥민의 상대팀 포메이션 별 평점을 구하기위해 손흥민선수가 출전한 경기의 상대방 의 포메이션 과 그경기의 평점을 <https://www.premierleague.com/> 사이트에서 크롤링은해서 데이터 정리를 할 예정이다.

팀의 주력선수들의 상대팀 포메이션 별 평점이 높을 수록 상대팀의 포메이션에 강하다는것이 증명되므로 승부예측에 포함시킬수있다고 생각한다.

처음 개발을 10팀을 기준으로 할 예정이다. 한 팀당 9팀과 경기를 할 수 있으므로 각 팀마다 세가지 분류의 데이터 분석 값 들의 합을 9개로 mongodb에 저장할 예정이다. 자신이 선택한 팀과 상대팀을 정한다면 그 두 팀 사이의 데이터들을 보임으로서 예측 결과를 보여줄수있다.

2.변경된 개발 기획

<https://www.premierleague.com/> 사이트를 크롤링 하면서 기존 기획을 했던점을 만족하는 프로그램을 만들수 없다는것을 판단하여 기획을 변경했다.

각팀의 선수별 선수 번호, 선수 이름, 득점 수(골키퍼의 경우는 클린 시트 수),선수의 appearance,자신이 출장하여 승리한 경기수,패배한 경기수를 크롤링 한다. 그 후 선수 별 승률을 출장 횟수와 승리한 경기 수로 계산한다.

유저들은 프로그램이 시작되면 2개의팀을 입력 할 수 있고 선택 후 첫번쨰로 입력한 팀에 대한 현재 등록된 선수들의 능력치(크롤링한 정보들) 목록을 볼 수 있고. 각 선수 별 승률을 그래프로 볼 수 있다. 그 후 자신이 원하는 선수 11명의 번호를 입력하면 입력된 선수들 의 승률만 볼 수 있다.

그 후 두번째 팀도 같은 방법으로 진행한다. 두번째 팀이 끝나면 승률 비교 그래프를 확인 할 수 있다.

3. 개발 코드

# coding: utf-8

import requests

import urllib

from scrapy.selector import Selector

##################################

url = 'https://www.premierleague.com/players/4999/Son-Heung-Min/stats'

r=requests.get(url)

\_html=r.text

stats=["Appearances","Goals","Wins","Losses"]

cal=[None]\*4

target=open('data/tottenham/playerlist.txt','w')

nodes=Selector(text=\_html).xpath("//\*[@class='playerDetails']")

for node in nodes:

    number=node.css("div[class='number']::text").extract()

    name=node.css("h1>div::text").extract()

    if number and name:

        number=number[0].strip()

        name=name[0].strip()

    target.write(number+'\t')

    target.write(name+'\t')

    print number,name

nodes=Selector(text=\_html).xpath("//\*[@class='topStat']")

i=0

for node in nodes:

    value = node.css("span[class='stat']>span::text").extract()

    if value and stats:

        value = value[0].strip()

    target.write(value+'\t')

    print stats[i], value

    cal[i]=value

    i=i+1

total=int(cal[2])+int(cal[3])

target.write(str(total)+'\t')

if total!=0:

    winper=(float(cal[2])/total)\*100

    target.write(str(winper))

else:

    winper=0

    target.write(str(winper))

target.write('\n')

print "출장횟수:",total

print "출장 횟수 대비 승률:",winper,"%"

print number,"txt파일로 저장완료\n"

이 코드 는 프리미어리그 홈페이지에 등록되어있는 선수들의 능력치 를 가져오는 코드이다 . 선수 능력 정보는 한 페이지에 있지 않으므로 이 코드를 팀의 선수 수 별로 이어 붙여 선수들의 정보를 가져온다. (토트넘23명,아스널25명,맨체스터시티 22명,첼시 20명 -> 각 선수들의 수만큼 이 코드들이 반복된다.) 가져온 후 출전 횟수와 승률도 구한다.

받아온 데이터들은 각각의 팀텍스트파일에 저장되고 선수한명당 한줄씩 저장된다. 또한 tsv 형태의 파일로 만들어서 데이터를 활용하기위해 선수의 크롤링 한 능력치 들을 탭으로 구분한다.

Tottenham.txt예시

7 Son Heung-Min 62 18 39 9 48 81.25

2 Kyle Walker 198 5 107 39 146 73.2876712329

3 Danny Rose 135 9 66 34 100 66.0

4 Toby Alderweireld 94 6 54 18 72 75.0

5 Jan Vertonghen 151 4 85 32 117 72.6495726496

16 Kieran Trippier 56 1 22 21 43 51.1627906977

27 Kevin Wimmer 15 0 9 2 11 81.8181818182

33 Ben Davies 125 4 55 37 92 59.7826086957

11 Erik Lamela 85 8 42 20 62 67.7419354839

15 Eric Dier 101 7 57 18 75 76.0

19 Mousa Dembele 205 12 107 47 154 69.4805194805

20 Dele Alli 70 28 45 8 53 84.9056603774

23 Christian Eriksen 134 31 75 28 103 72.8155339806

10 Harry Kane 116 78 66 23 89 74.1573033708

12 Victor Wanyama 121 8 63 31 94 67.0212765957

9 Vincent Janssen 27 2 17 3 20 85.0

25 Josh Onomah 13 0 8 2 10 80.0

29 Harry Winks 21 1 14 3 17 82.3529411765

30 Pau Lopez 0 0 0 0 0 0

17 Moussa Sissoko 143 11 58 59 117 49.5726495726

14 Georges Kevin-Nkoudou 8 0 5 2 7 71.4285714286

1 Hugo Lloris 170 59 98 34 132 74.2424242424

13 Michel Vorm 99 31 32 37 69 46.3768115942

이런식으로 현존하는 BIG 4 팀의 선수들의 정보를 가져왔다.

다음으로 이것들을 활용하기위한 메인 코드이다.

Projectteamvsteam.py

# coding: utf-8

import os

home=os.path.expanduser("~")

import findspark

findspark.init(os.path.join(home,"Downloads","spark-2.0.0-bin-hadoop2.6"))

import pyspark

myConf=pyspark.SparkConf()

spark = pyspark.sql.SparkSession.builder.master("local").appName("myApp").config(conf=myConf).config('spark.sql.warehouse.dir','file:///C:/Users/DongkyuShin/Desktop/sbigdata/s\_201111235/201111235/ProjectAssignment').getOrCreate()

import os

import sys

os.environ["SPARK\_HOME"]=os.path.join(os.path.expanduser("~"),'Downloads','spark-2.0.0-bin-hadoop2.6')

os.environ["PYLIB"]=os.path.join(os.environ["SPARK\_HOME"],'python','lib')

sys.path.insert(0,os.path.join(os.environ["PYLIB"],'py4j-0.10.1-src.zip'))

sys.path.insert(0,os.path.join(os.environ["PYLIB"],'pyspark.zip'))

from pyspark.sql import Row

import numpy as np

import matplotlib.pyplot as plt

team=[None]\*2

vsodd=[None]\*2

print ("Insert Tow Team.")

for i in range(0,2):

    team[i]=raw\_input("")

for a in range(0,2):

    if team[a]=='tottenham':

        cfile= os.path.join(os.path.expanduser("~"),\

                   "Desktop/sbigdata/s\_201111235/201111235/ProjectAssignment/data/tottenham/playerlist.txt")

        rdd = spark.sparkContext.textFile(cfile)

        parts = rdd.map(lambda x:x.split("\t"))

        player = parts.map(lambda p: Row(PlayerNumber=int(p[0].strip()),Name=p[1],Appearance=int(p[2].strip()),Goal=int(p[3].strip()),Win=int(p[4].strip()), Lose=int(p[5].strip()),Play=int(p[6].strip()),Winper=float(p[7].strip())))

        tDf = spark.createDataFrame(player)

        \_AppearanceRdd=tDf.rdd.map(lambda fields:fields[0]).collect()

        \_goalRdd=tDf.rdd.map(lambda fields:fields[1]).collect()

        \_loseRdd=tDf.rdd.map(lambda fields:fields[2]).collect()

        \_nameRdd=tDf.rdd.map(lambda fields:fields[3]).collect()

        \_playRdd=tDf.rdd.map(lambda fields:fields[4]).collect()

        \_numberRdd=tDf.rdd.map(lambda fields:fields[5]).collect()

        \_winRdd=tDf.rdd.map(lambda fields:fields[6]).collect()

        \_winperRdd=tDf.rdd.map(lambda fields:fields[7]).collect()

        value=[None]\*11

        odd=0

        print ("-------------------------------------------------------------------")

        print ("They are currently playing at EPL.")

        print ("PlayerNumber, PlayerName, Goal(CleanSheet), Number of Wins, Number of Losses.")

        for i in range(0,23):

            print np.array(\_numberRdd)[i],np.array(\_nameRdd)[i],np.array(\_goalRdd)[i],np.array(\_winRdd)[i],np.array(\_loseRdd)[i]

        print ("-------------------------------------------------------------------")

        def barchart(data,labels):

            num\_bars = len(data)

            positions = range(1, num\_bars + 1)

            plt.barh(positions, data, align='center')

            plt.yticks(positions, labels)

            plt.xlabel('Win Odd')

            plt.ylabel('Player')

            plt.title('Number of players won per player')

            plt.grid()

            plt.show()

        barchart(\_winperRdd,\_nameRdd)

        ##그래프 표현

        print ("Please enter 11 player numbers.")

        for i in range(0,11):

            playerselect=input("")

            value[i]=playerselect

        print ("-------------------------------------------------------------------")

        print ("The odds of the selected player.")

        for i in range(0,11):

            for j in range(0,23):

                if value[i]==np.array(\_numberRdd)[j]:

                    odd=odd+float(np.array(\_winperRdd)[i])

                    print np.array(\_nameRdd)[i],np.array(\_numberRdd)[i],np.array(\_winperRdd)[i],"%"

        oddsum=odd/11

        vsodd[a]=float(oddsum)

        print ("-------------------------------------------------------------------")

        print ("The odds average of 11 players.")

        print oddsum,"%"

        print ("-------------------------------------------------------------------")

        odd=0

        oddsum=0

    elif team[a]=='arsenal':

        cfile= os.path.join(os.path.expanduser("~"),\

                   "Desktop/sbigdata/s\_201111235/201111235/ProjectAssignment/data/arsenal/playerlist.txt")

        rdd = spark.sparkContext.textFile(cfile)

        parts = rdd.map(lambda x:x.split("\t"))

        player = parts.map(lambda p: Row(PlayerNumber=int(p[0].strip()),Name=p[1],Appearance=int(p[2].strip()),Goal=int(p[3].strip()),Win=int(p[4].strip()), Lose=int(p[5].strip()),Play=int(p[6].strip()),Winper=float(p[7].strip())))

        tDf = spark.createDataFrame(player)

        \_AppearanceRdd=tDf.rdd.map(lambda fields:fields[0]).collect()

        \_goalRdd=tDf.rdd.map(lambda fields:fields[1]).collect()

        \_loseRdd=tDf.rdd.map(lambda fields:fields[2]).collect()

        \_nameRdd=tDf.rdd.map(lambda fields:fields[3]).collect()

        \_playRdd=tDf.rdd.map(lambda fields:fields[4]).collect()

        \_numberRdd=tDf.rdd.map(lambda fields:fields[5]).collect()

        \_winRdd=tDf.rdd.map(lambda fields:fields[6]).collect()

        \_winperRdd=tDf.rdd.map(lambda fields:fields[7]).collect()

        value=[None]\*11

        odd=0

        print ("-------------------------------------------------------------------")

        print ("They are currently playing at EPL.")

        print ("PlayerNumber, PlayerName, Goal(CleanSheet), Number of Wins, Number of Losses.")

        for i in range(0,25):

            print np.array(\_numberRdd)[i],np.array(\_nameRdd)[i],np.array(\_goalRdd)[i],np.array(\_winRdd)[i],np.array(\_loseRdd)[i]

        print ("-------------------------------------------------------------------")

        def barchart(data,labels):

            num\_bars = len(data)

            positions = range(1, num\_bars + 1)

            plt.barh(positions, data, align='center')

            plt.yticks(positions, labels)

            plt.xlabel('Win Odd')

            plt.ylabel('Player')

            plt.title('Number of players won per player')

            plt.grid()

            plt.show()

        barchart(\_winperRdd,\_nameRdd)

        ##그래프 표현

        print ("Please enter 11 player numbers.")

        for i in range(0,11):

            playerselect=input("")

            value[i]=playerselect

        print ("-------------------------------------------------------------------")

        print ("The odds of the selected player.")

        for i in range(0,11):

            for j in range(0,25):

                if value[i]==np.array(\_numberRdd)[j]:

                    odd=odd+float(np.array(\_winperRdd)[i])

                    print np.array(\_nameRdd)[i],np.array(\_numberRdd)[i],np.array(\_winperRdd)[i],"%"

        oddsum=odd/11

        vsodd[a]=float(oddsum)

        print ("-------------------------------------------------------------------")

        print ("The odds average of 11 players.")

        print oddsum,"%"

        print ("-------------------------------------------------------------------")

        odd=0

        oddsum=0

    elif team[a]=='manchestercity':

        cfile= os.path.join(os.path.expanduser("~"),\

                   "Desktop/sbigdata/s\_201111235/201111235/ProjectAssignment/data/manchestercity/playerlist.txt")

        rdd = spark.sparkContext.textFile(cfile)

        parts = rdd.map(lambda x:x.split("\t"))

        player = parts.map(lambda p: Row(PlayerNumber=int(p[0].strip()),Name=p[1],Appearance=int(p[2].strip()),Goal=int(p[3].strip()),Win=int(p[4].strip()), Lose=int(p[5].strip()),Play=int(p[6].strip()),Winper=float(p[7].strip())))

        tDf = spark.createDataFrame(player)

        \_AppearanceRdd=tDf.rdd.map(lambda fields:fields[0]).collect()

        \_goalRdd=tDf.rdd.map(lambda fields:fields[1]).collect()

        \_loseRdd=tDf.rdd.map(lambda fields:fields[2]).collect()

        \_nameRdd=tDf.rdd.map(lambda fields:fields[3]).collect()

        \_playRdd=tDf.rdd.map(lambda fields:fields[4]).collect()

        \_numberRdd=tDf.rdd.map(lambda fields:fields[5]).collect()

        \_winRdd=tDf.rdd.map(lambda fields:fields[6]).collect()

        \_winperRdd=tDf.rdd.map(lambda fields:fields[7]).collect()

        value=[None]\*11

        odd=0

        print ("-------------------------------------------------------------------")

        print ("They are currently playing at EPL.")

        print ("PlayerNumber, PlayerName, Goal(CleanSheet), Number of Wins, Number of Losses.")

        for i in range(0,22):

            print np.array(\_numberRdd)[i],np.array(\_nameRdd)[i],np.array(\_goalRdd)[i],np.array(\_winRdd)[i],np.array(\_loseRdd)[i]

        print ("-------------------------------------------------------------------")

        def barchart(data,labels):

            num\_bars = len(data)

            positions = range(1, num\_bars + 1)

            plt.barh(positions, data, align='center')

            plt.yticks(positions, labels)

            plt.xlabel('Win Odd')

            plt.ylabel('Player')

            plt.title('Number of players won per player')

            plt.grid()

            plt.show()

        barchart(\_winperRdd,\_nameRdd)

        ##그래프 표현

        print ("Please enter 11 player numbers.")

        for i in range(0,11):

            playerselect=input("")

            value[i]=playerselect

        print ("-------------------------------------------------------------------")

        print ("The odds of the selected player.")

        for i in range(0,11):

            for j in range(0,22):

                if value[i]==np.array(\_numberRdd)[j]:

                    odd=odd+float(np.array(\_winperRdd)[i])

                    print np.array(\_nameRdd)[i],np.array(\_numberRdd)[i],np.array(\_winperRdd)[i],"%"

        oddsum=odd/11

        vsodd[a]=float(oddsum)

        print ("-------------------------------------------------------------------")

        print ("The odds average of 11 players.")

        print oddsum,"%"

        print ("-------------------------------------------------------------------")

        odd=0

        oddsum=0

    elif team[a]=='chelsea':

        cfile= os.path.join(os.path.expanduser("~"),\

                   "Desktop/sbigdata/s\_201111235/201111235/ProjectAssignment/data/chelsea/playerlist.txt")

        rdd = spark.sparkContext.textFile(cfile)

        parts = rdd.map(lambda x:x.split("\t"))

        player = parts.map(lambda p: Row(PlayerNumber=int(p[0].strip()),Name=p[1],Appearance=int(p[2].strip()),Goal=int(p[3].strip()),Win=int(p[4].strip()), Lose=int(p[5].strip()),Play=int(p[6].strip()),Winper=float(p[7].strip())))

        tDf = spark.createDataFrame(player)

        \_AppearanceRdd=tDf.rdd.map(lambda fields:fields[0]).collect()

        \_goalRdd=tDf.rdd.map(lambda fields:fields[1]).collect()

        \_loseRdd=tDf.rdd.map(lambda fields:fields[2]).collect()

        \_nameRdd=tDf.rdd.map(lambda fields:fields[3]).collect()

        \_playRdd=tDf.rdd.map(lambda fields:fields[4]).collect()

        \_numberRdd=tDf.rdd.map(lambda fields:fields[5]).collect()

        \_winRdd=tDf.rdd.map(lambda fields:fields[6]).collect()

        \_winperRdd=tDf.rdd.map(lambda fields:fields[7]).collect()

        value=[None]\*11

        odd=0

        print ("-------------------------------------------------------------------")

        print ("They are currently playing at EPL.")

        print ("PlayerNumber, PlayerName, Goal(CleanSheet), Number of Wins, Number of Losses.")

        for i in range(0,20):

            print np.array(\_numberRdd)[i],np.array(\_nameRdd)[i],np.array(\_goalRdd)[i],np.array(\_winRdd)[i],np.array(\_loseRdd)[i]

        print ("-------------------------------------------------------------------")

        def barchart(data,labels):

            num\_bars = len(data)

            positions = range(1, num\_bars + 1)

            plt.barh(positions, data, align='center')

            plt.yticks(positions, labels)

            plt.xlabel('Win Odd')

            plt.ylabel('Player')

            plt.title('Number of players won per player')

            plt.grid()

            plt.show()

        barchart(\_winperRdd,\_nameRdd)

        ##그래프 표현

        print ("Please enter 11 player numbers.")

        for i in range(0,11):

            playerselect=input("")

            value[i]=playerselect

        print ("-------------------------------------------------------------------")

        print ("The odds of the selected player.")

        for i in range(0,11):

            for j in range(0,20):

                if value[i]==np.array(\_numberRdd)[j]:

                    odd=odd+float(np.array(\_winperRdd)[i])

                    print np.array(\_nameRdd)[i],np.array(\_numberRdd)[i],np.array(\_winperRdd)[i],"%"

        oddsum=odd/11

        vsodd[a]=float(oddsum)

        print ("-------------------------------------------------------------------")

        print ("The odds average of 11 players.")

        print oddsum,"%"

        print ("-------------------------------------------------------------------")

        odd=0

        oddsum=0

print team[0],("vs"),team[1],(":"),vsodd[0],"%",("vs"),vsodd[1],("%"),("\n")

def barchart(data,labels):

    num\_bars = len(data)

    positions = range(1, num\_bars + 1)

    plt.barh(positions, data, align='center')

    plt.yticks(positions, labels)

    plt.xlabel('Win Odd')

    plt.ylabel('Team')

    plt.title('Team odds average')

    plt.grid()

    plt.show()

barchart(vsodd,team)

시작으로 스파크를 기때문에 스파크 경로 설정을 하고 파이스파크를 사용한다. 그 후 사용자가 표준입출력으로 두개의 팀을 기입 할 수 있다.

앞서 말한 대로 처음 입력한 팀의 선수 별 능력치를 표준 출력으로 보여준다. 이때 데이터프레임을 만들어서 선수의 정보를 8가지로 나눈다.

PlayerNumber , Name , appearance , Goal , Win , Lose , Play , Winner

각 선수 번호,이름, 능력,득점,승리,패배, 이다.(출장횟수와 승률을 출력안함 승률은 그래프로출력)

for i in range(0,20):

       print np.array(\_numberRdd)[i],np.array(\_nameRdd)[i],np.array(\_goalRdd)[i],np.array(\_winRdd)[i],np.array(\_loseRdd)[i]

이처럼 선수의 명수 로 반복문을 설정하고 출력한다.

선수 별 승률은 그래프로 출력한다

.         def barchart(data,labels):

            num\_bars = len(data)

            positions = range(1, num\_bars + 1)

            plt.barh(positions, data, align='center')

            plt.yticks(positions, labels)

            plt.xlabel('Win Odd')

            plt.ylabel('Player')

            plt.title('Number of players won per player')

            plt.grid()

            plt.show()

        barchart(\_winperRdd,\_nameRdd)

이 코드로 각 선수의 이름과 승률을 가져와 그래프로 표현한다.

그래프를 끄면 다음 차례가 이어진다. 자신이 표준입력으로 11명 선수의 번호를 입력하고 그 입력된 선수들의 승률과 승률 평균을 계산하여 보여준다.

print ("Please enter 11 player numbers.")

        for i in range(0,11):

            playerselect=input("")

            value[i]=playerselect

        print ("-------------------------------------------------------------------")

        print ("The odds of the selected player.")

        for i in range(0,11):

            for j in range(0,20):

                if value[i]==np.array(\_numberRdd)[j]:

                    odd=odd+float(np.array(\_winperRdd)[i])

                    print np.array(\_nameRdd)[i],np.array(\_numberRdd)[i],np.array(\_winperRdd)[i],"%"

        oddsum=odd/11

        vsodd[a]=float(oddsum)

두번째 팀 도 마찬가지로 진행을 할 수 있고 마지막 두 팀의 승률 평균을 비교하는 그래프를 보여준다.

print team[0],("vs"),team[1],(":"),vsodd[0],"%",("vs"),vsodd[1],("%"),("\n")

def barchart(data,labels):

    num\_bars = len(data)

    positions = range(1, num\_bars + 1)

    plt.barh(positions, data, align='center')

    plt.yticks(positions, labels)

    plt.xlabel('Win Odd')

    plt.ylabel('Team')

    plt.title('Team odds average')

    plt.grid()

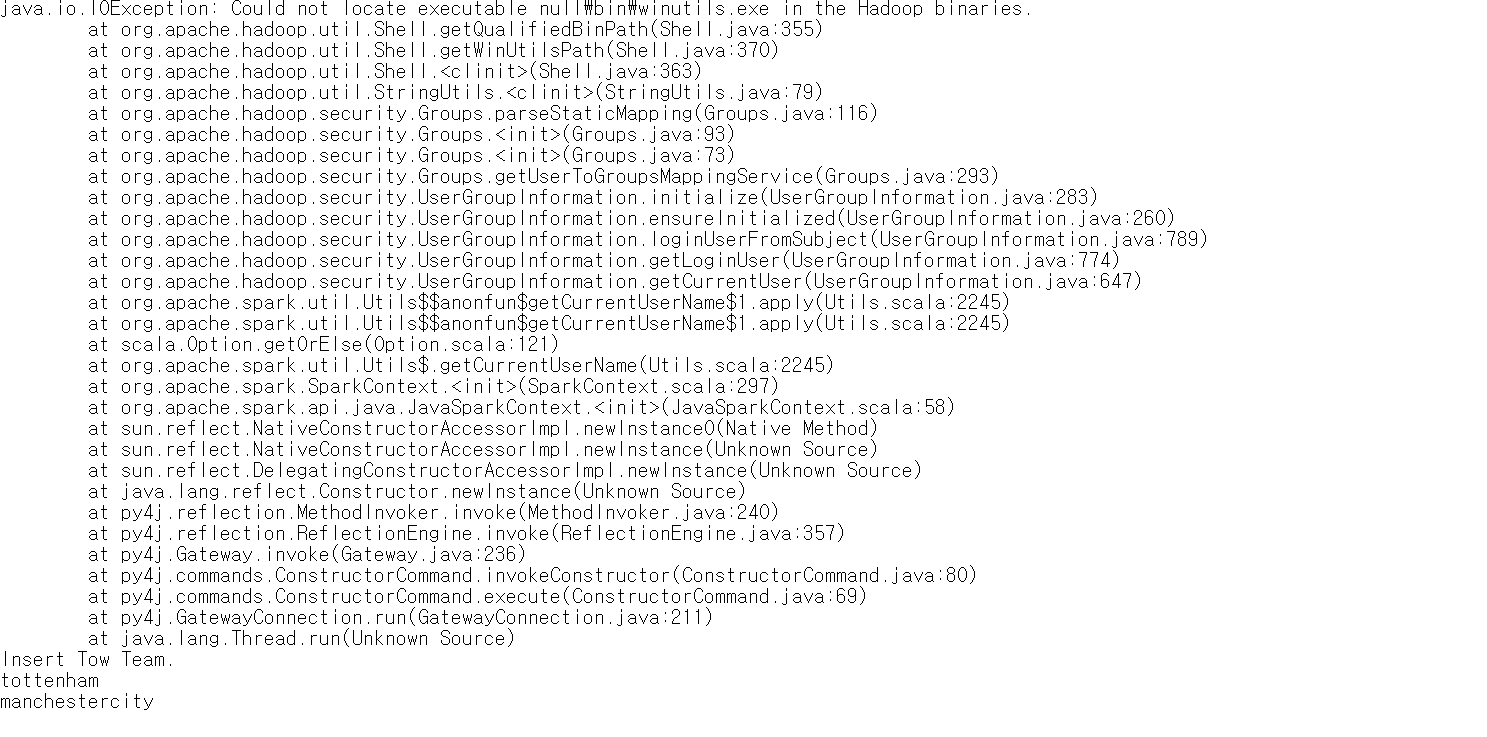
    plt.show()

barchart(vsodd,team)

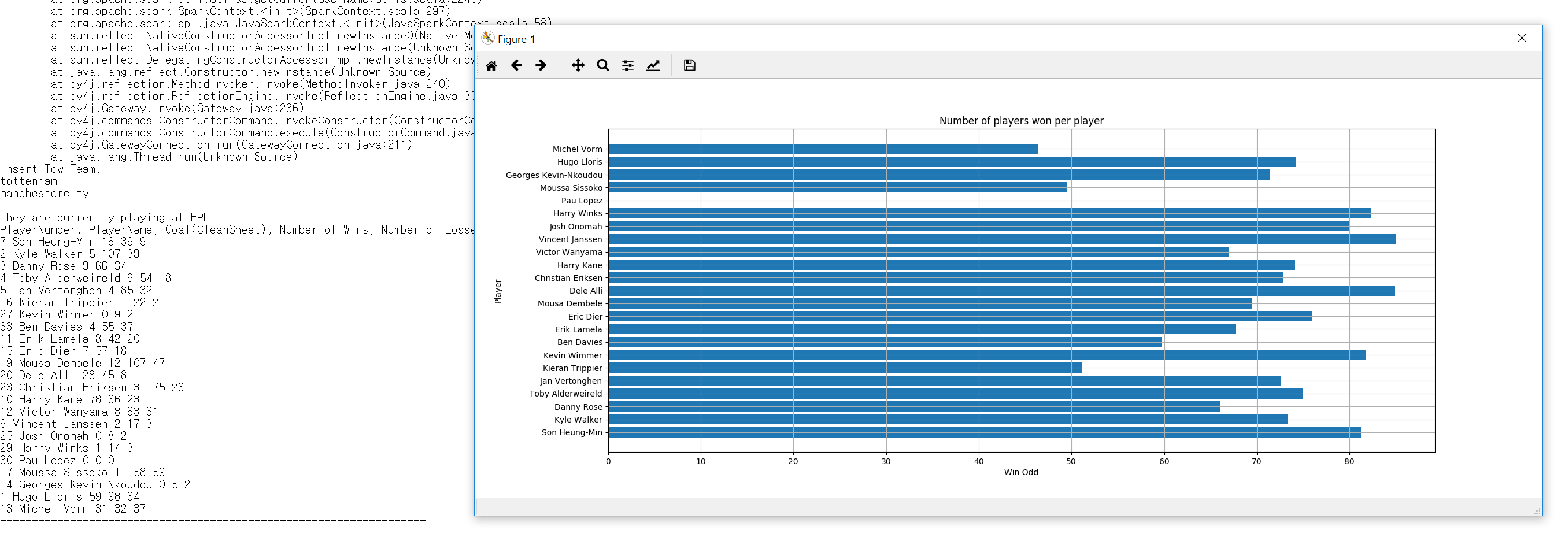
team 이라는 배열 변수에 들어있는 첫번째와 두번째 팀의 이름과

vsodd라는 배열 변수에 들어있는 첫번째와 두번째 팀의 평균 승률을 그래프로 나타낸다.

3. 프로그램 시행 과정



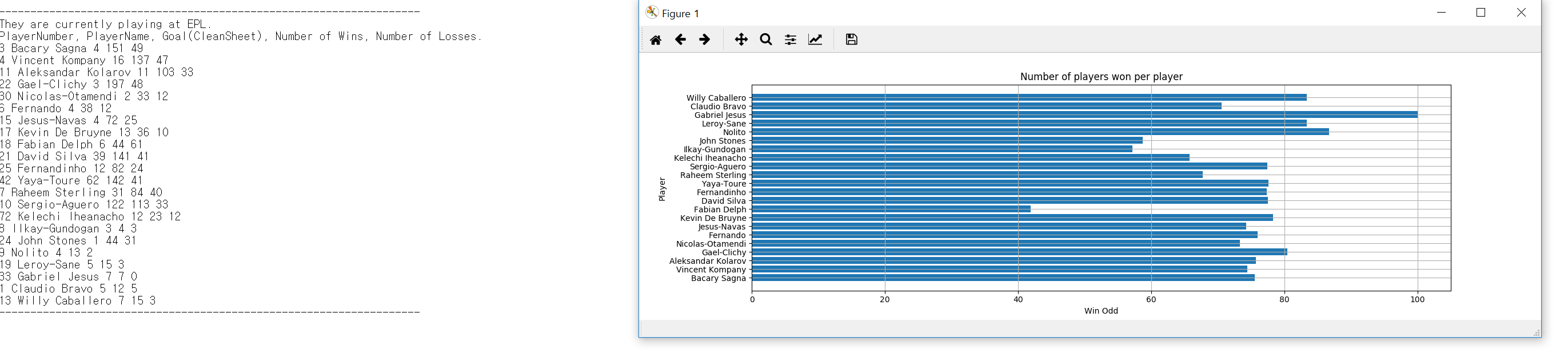
(1) Insert Tow Team 이라는 표준 출력과 두개의 팀을 기입 할 수 있다.



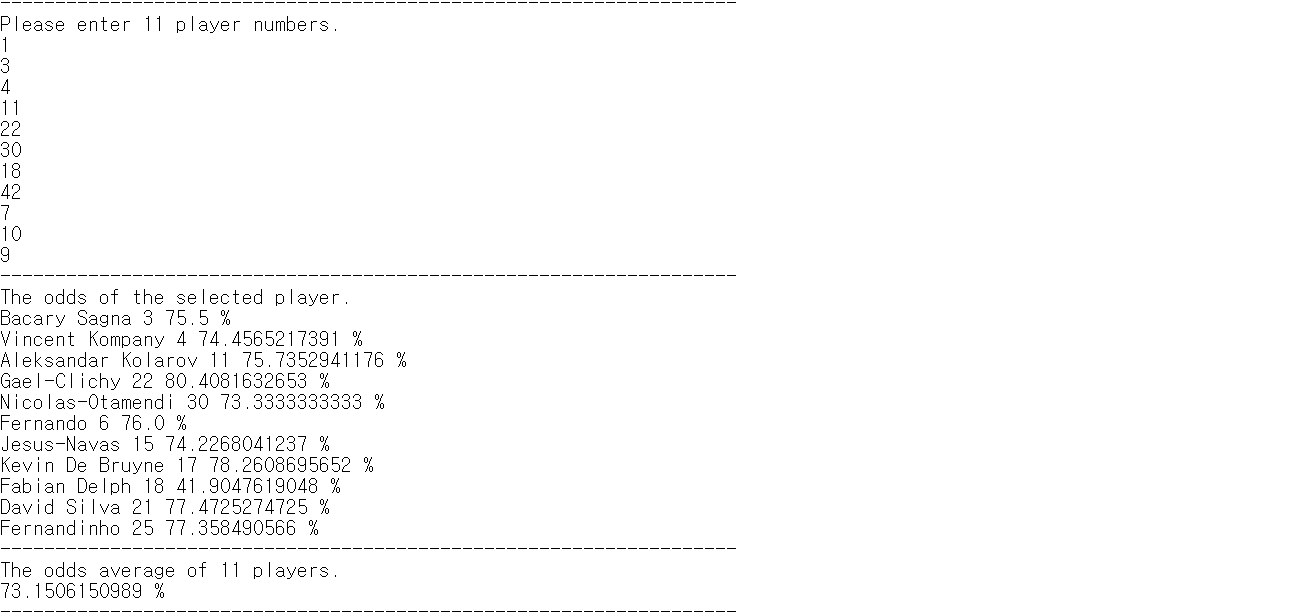
(2) 선택한 팀 중 첫번째 팀의 능력치들 과 선수 별 승률을 보여준다.



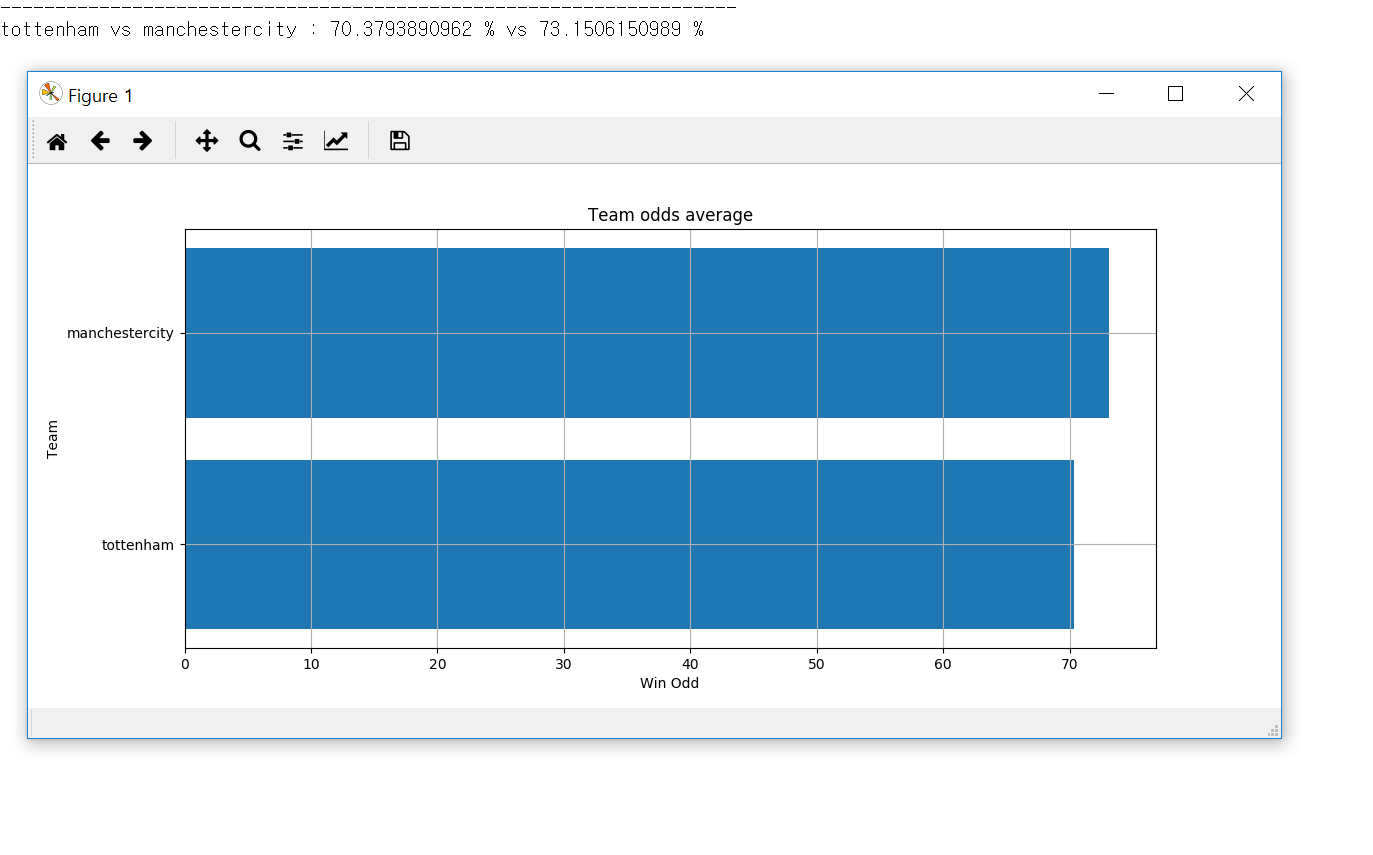
(3) 사용자가 첫번째 팀에 대한11명의 번호를 기입 할 수 있다. 기입한 선수들의 승률을 표준 출력으로 보여준다. 또한 승률의 평균도 보여준다.



(4) 두번째 팀의 선수 능력치와 선수별 승률 그래프를 보여준다.



(5) 사용자가 두번째 팀에 대한 11명의 번호를 기입 할 수 있다. 기입한 선수들의 승률을 표준 출력으로 보여준다. 또한 승률의 평균도 보여준다.



(6) 두 팀의 승률을 표준 출력으로 비교하고 그래프로 보여준다.