Project – 1 Documentation

**Scraping and Analysis of Stastistics of Sports Players from Real-Time Webpage**

**OVERVIEW :**A project that gives you a better understanding of scraping data from websites and how to analyse them. Usage of various libraries as NumPy, Mat Plot, Pandas.

In the course of completing the project, you use the web scraping function, converting the extracted data into a pandas data Frame, and Storing the analysed data.

**Problem Statement**

Web scrape basketball statistics from Wikipedia of some of the greatest basketball players and export it as a CSV file format.

**Software Requirements**

1. Programming Language : Python

2. Environemnt: Jupyter Notebooks / Google Collab

3. Database: CSV(export type)

4. Operation System: Windows XP or above

5. Librarires Used: Beautiful Soup, requests, Pandas, NumPy, boto3 ,Matplotlib, display

1. **Open a New Notebook and import the required libraires**

|  |  |
| --- | --- |
|  | import bs4  import requests  import pandas as pd  import numpy as np  ! pip install boto3  import boto3  import matplotlib.pyplot as plt  from IPython.display import display |

Description:

In the above part of the program, we are importing all the necessary modules to get a proper information about the basketball players.

“bs4” is a module named “Beautiful Soup 4”, this module is used to convert HTML code of any website in a proper way.

“Requests” is a module that is used to request the HTML code of any website.

“Pandas” is one of the most important modules that is used in Data Science. Pandas generally deal with the data sets, using pandas one can clean the data, extract the data, explore the data, and manipulate the data.

“NumPy” is a module that is used to perform several mathematical operations.

“matplotlib.pylot” is a module that is used to plot the data in various ways. It is used to visualize the data in much better.

“Ipython.display” is a module that is used to display various objects.

1. **Reading the webpage**

def get\_basketball\_stats(link='https://en.wikipedia.org/wiki/Michael\_Jordan'):

response = requests.get(link)

soup = bs4. BeautifulSoup(response.text, 'html.parser')

Description:

In the above part of the program there is a function called “get\_basketball\_stats” which is taking an argument that contains a link/URL, next line is followed by requests.get(link) in which requests module will extract all the HTML data from specified link and that data is stored under a variable named response.

The following line contains bs4.BeautifulSoup(response.text, 'html.parser') in which BeautifulSoup4 module is used to convert all the HTML code from response in a normal way that is understandable for most of the people.

1. **Main Function Process**

|  |  |
| --- | --- |
| table = soup.find(class\_='wikitable sortable')  headers = table.tr  titles = headers.find\_all('abbr')  data = {title['title']: [] for title in titles}  for row in table.find\_all('tr')[1:]:  for key, a in zip(data.keys(),row.find\_all('td')[2:]):  data[key].append(''.join(c for c in a.text if (c.isdigit() or c == '.' )))  Min = min([len(x) for x in data.values()])  for key in data.keys():  data[key] = list(map(lambda x: float(x), data[key][:Min]))  return data  Description:  In the above part of the code the following action is running,  We know that a website contains various classes, by Inspecting the website we got to know about the class of that particular table from which we need to extract the data, the class of the table that we need is “wikitable sortable”, the data of the table is stored under the variable named table, using the same Inspect method we also found the class name of the headers in the table i.e., “tr” tag and the headers are stored inside the variable “headers”, but in headers, some of the words are having abbreviations that should be expanded so a variable is introduced named titles that finds all the abbreviations of the headers variable.  All the table columns should store the data in the form of a dictionary in which title is the key and the column values will be the values.  Now we need to start filling the values of the dictionary from the second line i.e., index no 1 because index no 0 acts as the key in the dictionary i.e., table header, next line contains “for key, a in zip(data.keys(),row.find\_all('td')[2:]):              data[key].append(''.join(c for c in a.text if (c.isdigit() or c == '.')))”  , here all the table values are stored in tag “td” and if that particular table values has some special characters other than numbers (including point (’.’)) shall be removed and the data shall be passed as the value for corresponding key.  The next line is followed by removing the last two rows that contains information which is not necessary for us to compute which leads to data disturbance and finally the dictionary i.e., data is being returned.     1. **Declaring links and names of the personals to scrap the data** |  |
| links=['https://en.wikipedia.org/wiki/Michael\_Jordan'\  ,'https://en.wikipedia.org/wiki/Kobe\_Bryant'\  ,'https://en.wikipedia.org/wiki/LeBron\_James'\  ,'https://en.wikipedia.org/wiki/Stephen\_Curry']  names=['Michael Jordan','Kobe Bryant','Lebron James','Stephen Curry'] |  |

michael\_jordan\_dict = get\_basketball\_stats(links[0])

kobe\_bryant\_dict = get\_basketball\_stats(links[1])

lebron\_james\_dict = get\_basketball\_stats(links[2])

stephen\_curry\_dict = get\_basketball\_stats(links[3])

mj\_table = pd. DataFrame(michael\_jordan\_dict)

kb\_table = pd. DataFrame(kobe\_bryant\_dict)

lj\_table = pd. DataFrame(lebron\_james\_dict)

sc\_table = pd. DataFrame(stephen\_curry\_dict)

list\_table =[mj\_table, kb\_table, lj\_table, sc\_table]

i = 0

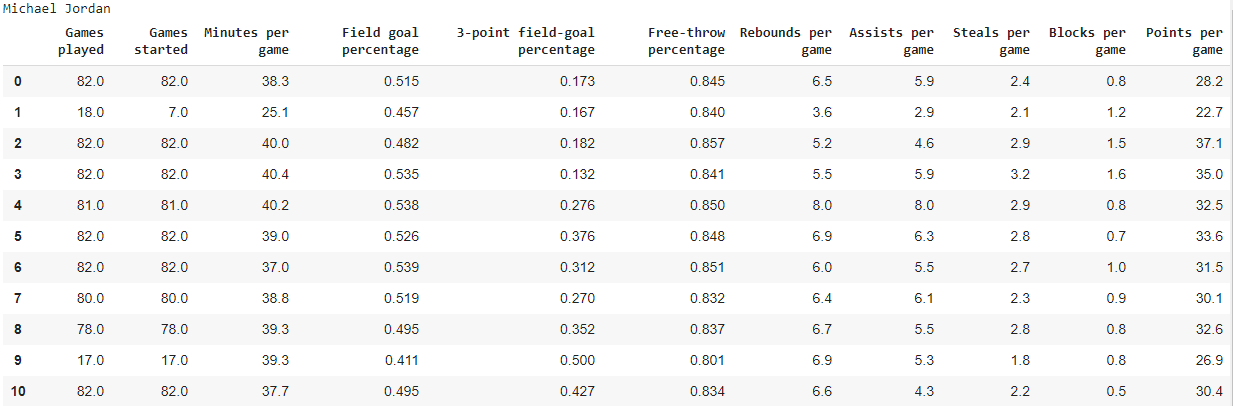
for name in names:

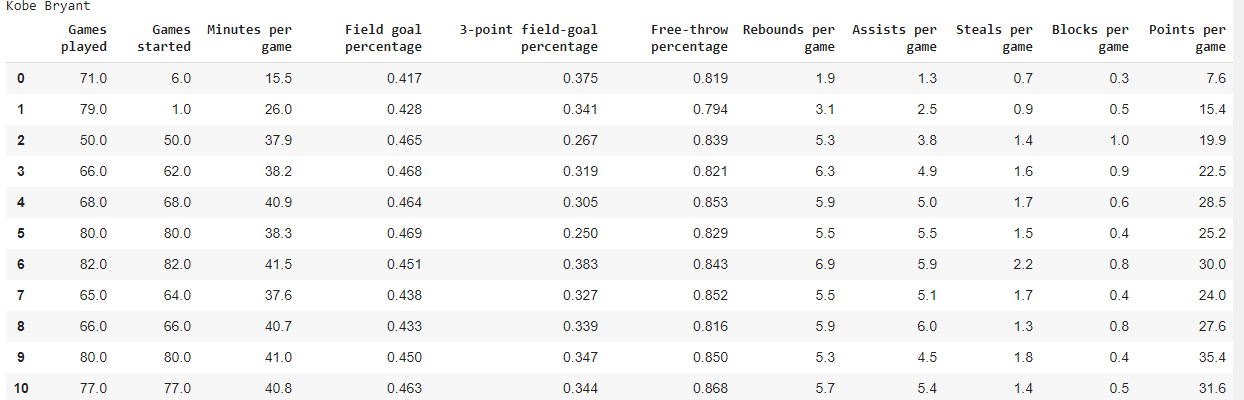
    print(name)

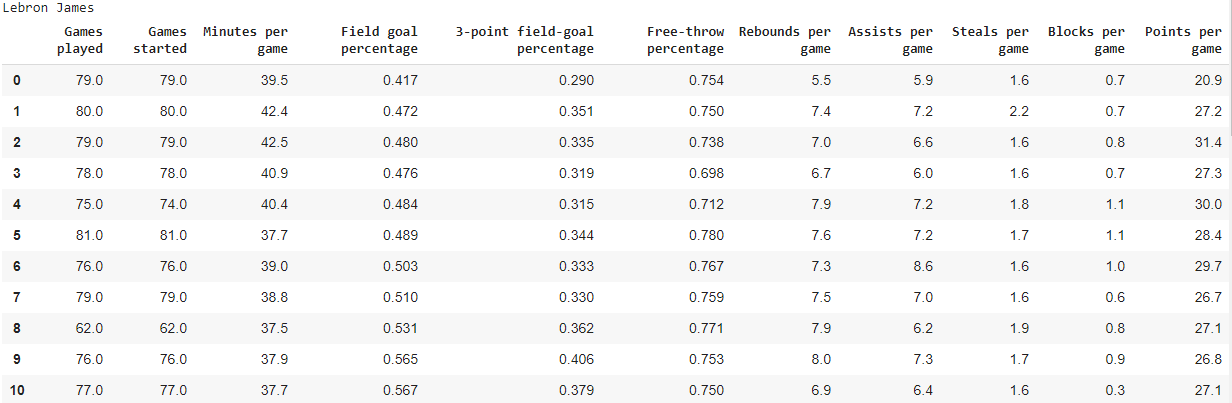
    display(list\_table[i])

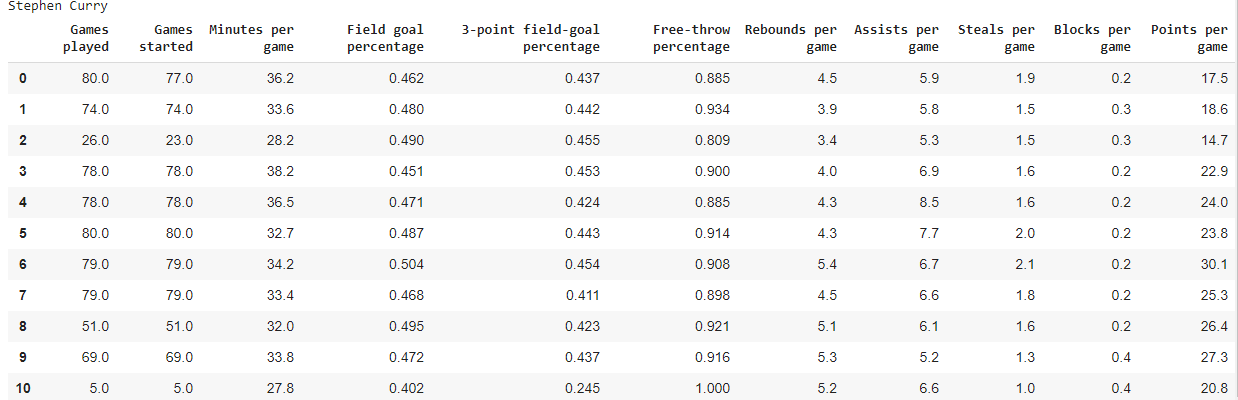
    i += 1

**Output:**









Description:

In the above part of the code, we are storing Wikipedia links of player’s Michael Jordan, Kobe Bryant, Lebron James and Stephen Curry in a list called links and storing their names in the list names., the code is now followed by creation of dictionaries for each player and each variable is calling the function “get\_basketball\_stats” and passing corresponding link as arguments from variable links, now each variable stores the table of each player.

Using pandas, every player’s data frame is created and are stored in list named list\_table.

Using for loop every players table is now displayed.

**6.Making a plot using matplotlib**

j - 0

for name in names:

    plt.plot(list\_table[j][['Points per game']],label=name)

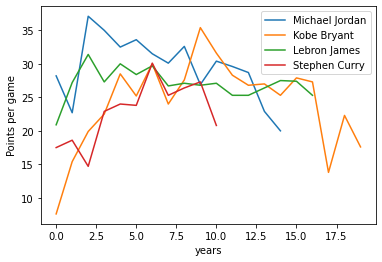
    plt.legend()

    plt.xlabel('years')

    plt.ylabel('Points per game')

d - 1

**Output:**

****

Description:

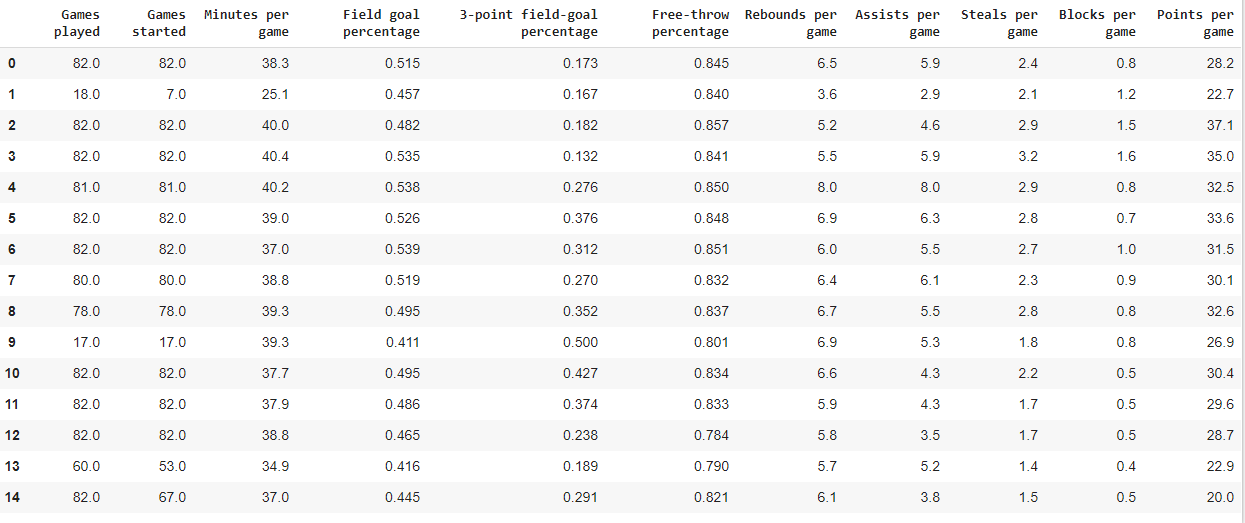
In the above piece of the code, using matplotlib.pyplot, graph is drawn for the column “Points per game” of every player, plt.legend() helps in identifying the plots corresponding to which player, “plt.xlabel('years')” is used to label the x-axis as years and “plt.ylabel('Points per game')” is used to label the y-axis as Points per game.

**7.Storing the Player Statistics in Object Storage**

csv\_name = 'MJ1.csv'

mj\_table.to\_csv(csv\_name)

mj\_table



Description:

The above part of the code is used to convert the data into csv format, as csv format is much prioritized to use in Data Science. “ csv\_name = 'MJ1.csv' ” is used to store the csv file of ‘MJ1.csv’ to the variable ‘csv\_name’, “ mj\_table.to\_csv(csv\_name) ” is used to convert ‘mj\_table’ to csv file and finally displayed csv file.

|  |
| --- |
|  |

**Conclusion**

**From this project I understood the steps involved in Data Science accordingly from**

**Data Collection, in which we collected the data from Wikipedia profiles of Basketball Players,**

**Data Cleaning & Data Wrangling, we cleaned the data i.e., we removed the data that is not necessary by keeping only integer and decimal values,**

**Data Analysis, one can analyze any data he/she has.**

**Data Visualization, we Visualized the data of all the Basketball players according to their points,**

**Data Prediction, this is the final step of Data Science in which one can predict the data that he/she has by training the model.**