Final Project Report

Project Code

Github repository link: https://github.com/russjin/SI507_final_project

Data Sources

First we scrape the web page which contains all the NBA player:
https://www.nba.com/players

```
response = urlopen('https://www.nba.com/players')
html_doc = response.read().decode('utf-8')
player_slug = re.findall("\"PLAYER_SLUG\":\"([^\"]+)\"", html_doc)
player_id = re.findall("\"PERSON_ID\":(\d+)", html_doc)
```

Then we use all the Player ID and the Player Slug scraped from this website to crawl and access the web page for each player iteratively:

```
for i in range(len(player_id)):
   player_url =
f"https://www.nba.com/player/{player_id[i]}/{player_slug[i]}"
   response = urlopen(player_url)
```

When we access those over 500 players' homepage iteratively, we met a lot of 502 Bad Gateway Error, so we use caching to save those data.

```
{"precious-achiuwa": {"slug": "precious-achiuwa", "id": "1630173", "url": "https://www.nba.com/player/1630173/precious-
achiuwa", "fullname": "Precious Achiuwa", "team": "Toronto Raptors", "pos": "Forward", "no": "5", "image":
"https://cdn.nba.com/headshots/nba/latest/1040x760/1630173.png"}, "steven-adams": {"slug": "steven-adams", "id": "203500",
"url": "https://www.nba.com/player/203500/steven-adams", "fullname": "Steven Adams", "team": "Memphis Grizzlies", "pos":
"Center", "no": "4", "image": "https://cdn.nba.com/headshots/nba/latest/1040x760/203500.png"}, "bam-adebayo": {"slug": "bam-
adebayo", "id": "1628389", "url": "https://www.nba.com/player/1628389/bam-adebayo", "fullname": "Bam Adebayo", "team": "Miami
Heat", "pos": "Center-Forward", "no": "13", "image": "https://cdn.nba.com/headshots/nba/latest/1040x760/1628389.png"}, "ochai-
agbaji": {"slug": "ochai-agbaji", "id": "1630534", "url": "https://www.nba.com/player/1630534/ochai-agbaji", "fullname":
```

In all there are 499 records of players which have valid names and numbers in this 2022-23 season. However, some players don't have an appearance yet so they need to be excluded. For all the players scraped, we have their full name, team, position, number, homepage url and one image of their headshot.
Next we access the brr website which has a more rich source of player's stats: https://www.basketball-reference.com/leagues/NBA_2023_per_game.html. We

scrape all players' stats and merge them with player's information above. Since the stats are updated everyday, we do not use the caching here.

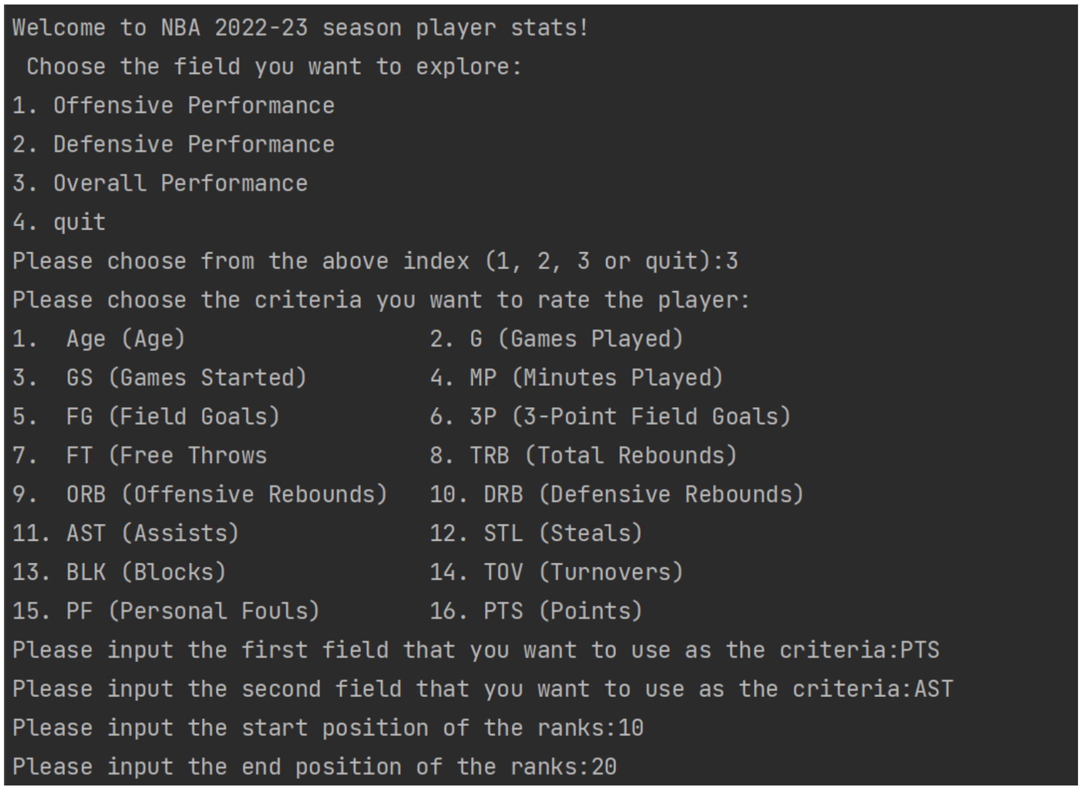
Data Structure

We label all players and store them in a tree structure. To fetch the player's information more quickly, we use a binary search tree structure to store all the information of players. Followings are the code for the binary search tree we implement in this project:

```
class NBAPlayerTree:
   def __init__(self, key=None, val=None):
       self.key = key
       self.val = val  # position in AllPlayers Array
       self.left = None
       self.right = None

   def getKey(self):
       return self.key

   def insert(self, key, val):
       if self.key is None:
           self.key = key
           return

       if key < self.getKey():
           if self.left:
               self.left.insert(key, val)
               return
           self.left = NBAPlayerTree(key, val)
           return
       if self.right:
           self.right.insert(key, val)
           return
       self.right = NBAPlayerTree(key, val)

   def inorder(self, keys):
       if self.left is not None:
           self.left.inorder(keys)
       if self.key is not None:
           keys.append(self.key)
       if self.right is not None:
           self.right.inorder(keys)
       return keys

   def exists(self, key):
       if key == self.key:
           return self.val
       if key < self.key:
           return self.left.exists(key)
       else:
           return self.right.exists(key)
```

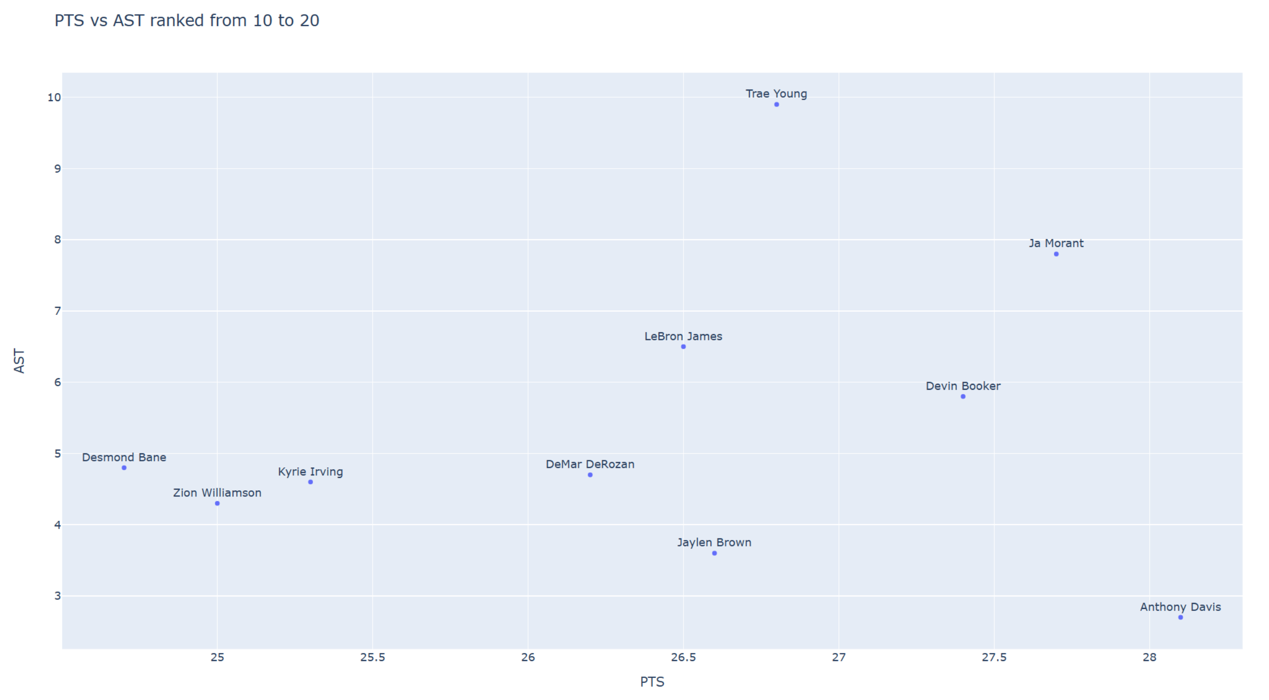
Interaction and Presentation Plans

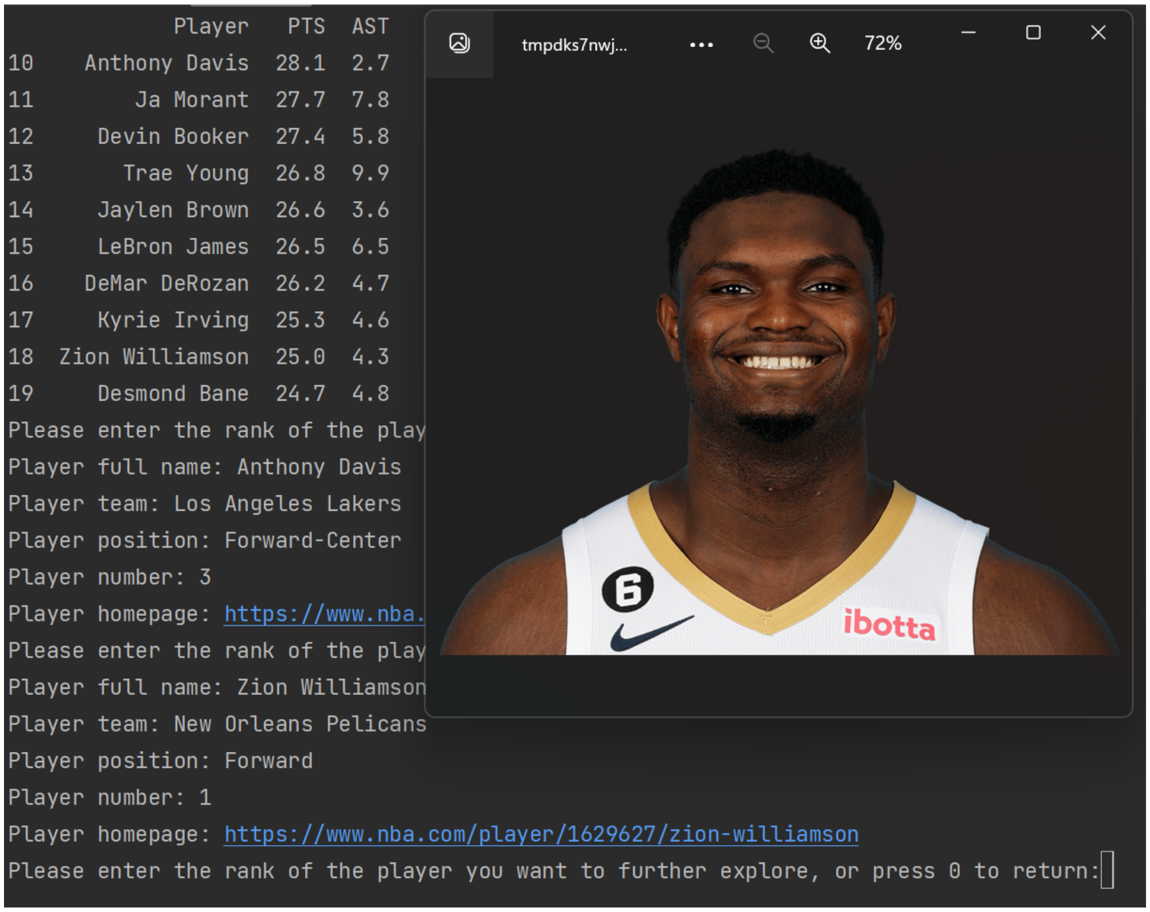
Interaction tasks should be performed when users input the methods to access the data. To be specific, when users want to see who were among the top 10 offensive players last season, they may select point and field goals made as the judging criteria, and the system will prompt out the predicted players using the preset model. According to the project overview, we use ‘plotly’ to visualize the data in a scatter plot. We use command prompt to perform all the interaction tasks in this project.
We show an example of the workflow of our project, details will be shown in the video.



```
Welcome to NBA 2022-23 season player stats!
 Choose the field you want to explore:
1. Offensive Performance
2. Defensive Performance
3. Overall Performance
4. quit
Please choose from the above index (1, 2, 3 or quit):3
Please choose the criteria you want to rate the player:
1.  Age (Age)                  2. G (Games Played)
3.  GS (Games Started)         4. MP (Minutes Played)
5.  FG (Field Goals)           6. 3P (3-Point Field Goals)
7.  FT (Free Throws            8. TRB (Total Rebounds)
9.  ORB (Offensive Rebounds)   10. DRB (Defensive Rebounds)
11. AST (Assists)              12. STL (Steals)
13. BLK (Blocks)               14. TOV (Turnovers)
15. PF (Personal Fouls)        16. PTS (Points)
Please input the first field that you want to use as the criteria:PTS
Please input the second field that you want to use as the criteria:AST
Please input the start position of the ranks:10
Please input the end position of the ranks:20
```

When we run the program, the command line will prompt us with the field we can explore with, that is players' offensive performance, defensive performance and overall performance. Here we select overall performance as an example. And for the criteria we would like to rank the players, we choose the points they made and the assists they made per game and we would like to see the players who rank from no.10-no.19 here. Then the program generates a pandas dataframe showing the players who rank in this interval and their stats we selected. Also a plotly-generated picture prompt out in the web browser showing the scatter plot of the above two stats. Afterwards, we can choose the rank number and look into more detail of that player. Here we input 18 and the program prompts Zion Williamson with his team, position, number, image and a url directing to his homepage. We can choose to look into more players in this list or start a new ranking by pressing 0.




```
                 Player   PTS  AST
10    Anthony Davis  28.1  2.7
11        Ja Morant  27.7  7.8
12     Devin Booker  27.4  5.8
13       Trae Young  26.8  9.9
14     Jaylen Brown  26.6  3.6
15     LeBron James  26.5  6.5
16    DeMar DeRozan  26.2  4.7
17     Kyrie Irving  25.3  4.6
18  Zion Williamson  25.0  4.3
19     Desmond Bane  24.7  4.8
Please enter the rank of the play
Player full name: Anthony Davis
Player team: Los Angeles Lakers
Player position: Forward-Center
Player number: 3
Player homepage: https://www.nba.
Please enter the rank of the play
Player full name: Zion Williamson
Player team: New Orleans Pelicans
Player position: Forward
Player number: 1
Player homepage: https://www.nba.com/player/1629627/zion-williamson
Please enter the rank of the player you want to further explore, or press 0 to return:
```

Demo Link

https://github.com/russjin/SI507_final_project/blob/main/russjin_si507_demo.mp4