PROJECT REPORT

BOSTON CELTICS (2018 – 2019)

ATLANTIC DIVISION WEBSCRAPING

AND DATA ANALYSIS.

DS 5010 -FALL 19

BY- OMKAR WAGHMARE

- ## PROBLEM STATEMENT:

Aim to extract 2018-2019 Boston Celtics (Atlantic Division) team and player statistics from the official NBA site using beautiful soup (python library), perform exploratory data analysis on it and find attributes which make a player valuable.

- ## REASON FOR SELECTING THIS TOPIC:

Since the age of 12 I fell in love with basketball. I played at numerous levels, be it for college or state. I always knew that the key to winning a game was team work, but I never knew what makes a player 'valuable' to a team. Hence, I used this opportunity to explore the attributes which make a player important to the team, and what better way to do this than performing analysis on NBA players. I chose Boston Celtics as this is the official Boston's NBA team.

- ## OBJECTIVES OF THIS PROJECT:

Obtain few key characteristics of a player that make him valuable. In our case the value of the player will be determined by the Salary he gains per season. This data has been collected from the ESPN website.

- PROCEDURE:

1. Successfully scrape the data from official NBA website.
2. Clean the data
3. Data Processing
4. Perform Data Visualization
5. Analyse Data and gather Inferences
6. Generate a Hypothesis with respect to the objective
7. Test the hypothesis using a Machine Learning Model

1) Scraping the data from official NBA website:

a) LIBRARY USED:

i) Pandas : for storing the data in the dataframe

ii) Beautiful Soup 4 : for extracting the data from the HTML text

iii) Selenium : for automating the extraction process.

b) DATA EXTRACTED:

i) TEAM BOX SCORES (2018 – 2019):

(1) Data layout: contains 82 rows and 21 columns

(2) First column has the match date and the teams. (‘v/s’ implies: location = Home, ‘@’ implies: location = ‘Away’)

(3) Each entry in the first column is a hyperlink, which in-turn clicked forwards us to the box score of the respective match which has player statistics in it.

(4) 82 rows correspond to 82 Matches in the regular season

(4) Rest of the columns contain overall team stats for each game such as minutes played, points scored, won or lost, rebounds, turnovers, etc

ii) PLAYER BOX SCORES (2018 – 2019):

(1) Data layout: contains 23 columns and 883 rows

(2) Every hyperlink from the Team stats table is accessed using selenium, and data is extracted for that match

(3) Data contains player names and statistics for that particular match

(4) Dynamically the match name is appended using pandas and bs4.

(5) Data is then stored in a cvs file for further analysis.

c) CHALLENGES FACED:

-The first two days for me went in learning beautiful soup's full functionalities.

- when I started data extraction process, I found out that the data on the NBA website was being dynamically generates.

- that is the data used JavaScript to be rendered, and hence using just beautiful soup would not be sufficient.

- I then added selenium to my code, the logic behind this was I loaded the website using selenium, an automated testing tool, then used bs4 to scrape the loaded website. Adding a sleep time of 5 seconds was important as the browser rendered the data locally.

- fortunately for me the NBA website had uniform formatting, hence using the same logic for all the 82 links (matches) was possible.

-The final challenge I faced during web scraping was going to the next page of the table. If you have a close look at the NBA website (https://stats.nba.com/team/1610612738/boxscores/?Season=2018-19&SeasonType=Regular%20Season) , you'll see that all 82 links are not displayed in one table.

- I overcame this problem by using selenium's xpath function to find the next tab and then clicked it using selenium's click method.

d) OTHER DATA:

- Unfortunately, NBA did not provide two main attributes for player stats. These were annual player salary and player efficiency rate. Anyone who knows a little bit about basketball will know that these players are paid a lot of amount and just like every other sport they are ranked according to their salary and player efficiency rate.

- Hence I used Instant Data Scraper on the ESPN website to find out these stats, and manually stored them in a new csv file called player_salary.

- This file has 17 rows and 3 columns (player name, salary cap, player efficiency rate)

2) CLEANING THE DATA: (JUPYTER NOTEBOOK)

1. TEAM STATS:

   - The team statistics data file was already clean. Fortunately for me the NBA website has complete data and no missing values were found.
   - I added one more column called as W/L using pandas, which tells the status of the match, weather the Celtics won or lost.

2. PLAYER STATS:

   - The player stat data file was clean and had no missing values.

   - some processing was needed as the players had positions in which they played leading after their names.

   - eg: Kyrie Irvin C (C – Centre position)

3. PLAYER SALARY:

   - The player salary dataset was missing column names. I added these names explicitly after importing the dataset.

   - Along with that a few missing values were found, which I dropped completely.

- Everything from data cleaning onwards was carried out on jupyter notebook, as it is extremely handy in data visualization. Data extraction process was written and executed on a python IDE.

3) DATA PREPROCESSING:

1. TEAM STATS:

- The MIN column was dropped as each match was played for exactly 240 min and was not an important factor.

- A ‘LOCATION’ column was added in the team stats table using a simple logic. If the row in MATCHUP column contained ‘@’ location was set to A, or else if it contained ‘V/S’ location was set to ‘H’.

- After this Label Encoding was performed to convert object types to int. Specifically on the LOCATION and W/L columns.

2. TEAM SALARY:

- No Pre processing was needed for this table.

3.PLAYER STATS:

- The position names leading the player names were removed.

- along with the W/L column was added to the table.

- The player annual salary from player stats was divided by the number of games the player played to create a new column called SALARY_PER_GAME.

- The players efficiency rating from player_salary was also added to this table.

4)DATA VISUALIZATION:

1. LIBRARIES USED:

   - matplotlib.pyplt

   - seaborn

2. TYPES OF GRAPHS GENERATED:

   - Heat Maps

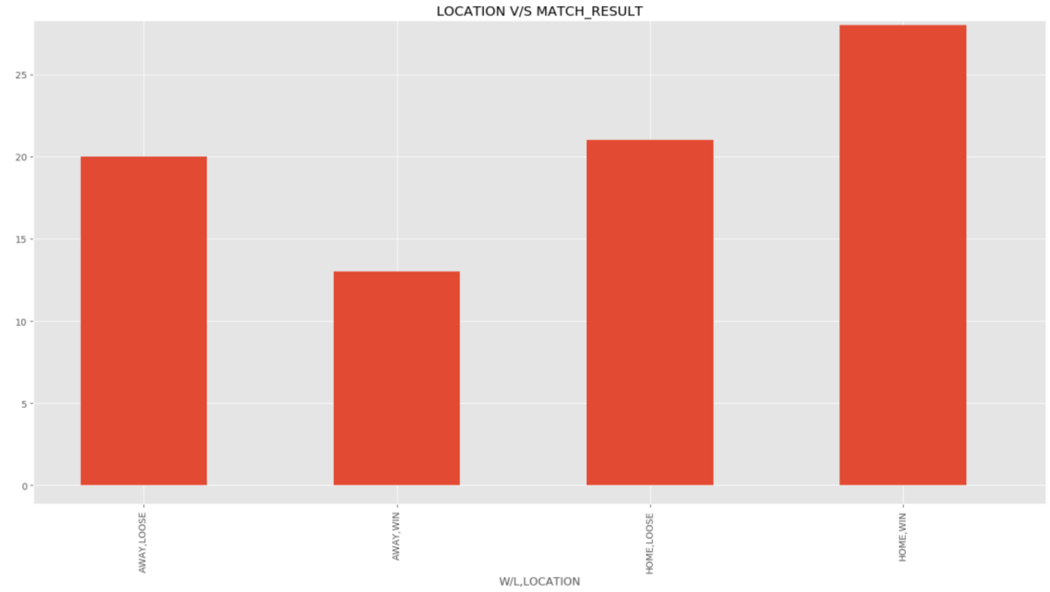
   - Scatter plots

   - Bar charts

- used in co-relation with the data analysis step, explained in more detail in the next section.

5)DATA ANALYSIS:

- The main objective of this project is to find what makes a player valuable, but I thought that it would be interesting in knowing if the location of the match affected the result in any ways.



Looking at the graph above, we could say that playing at home is more advantageous to the team.

- Next, I looked at player stats table. I divided the analysis process in two main categories, Offence and Defence

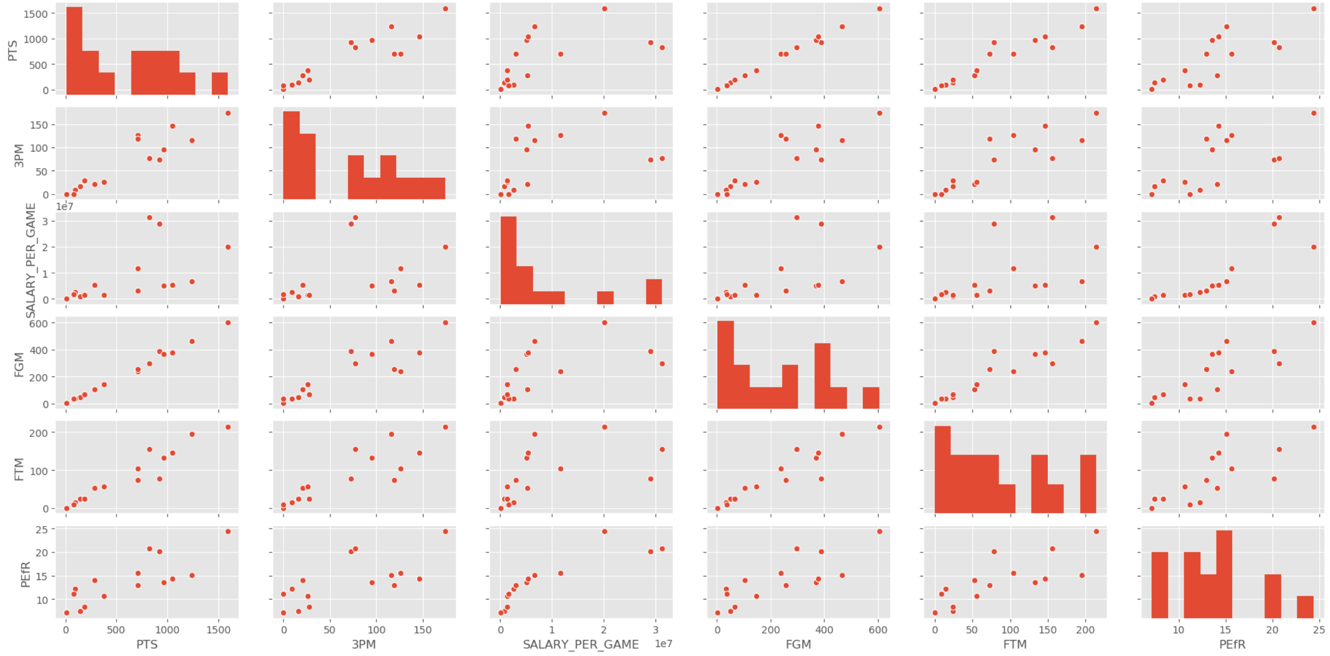
GLOSSARY:

https://stats.nba.com/help/glossary/

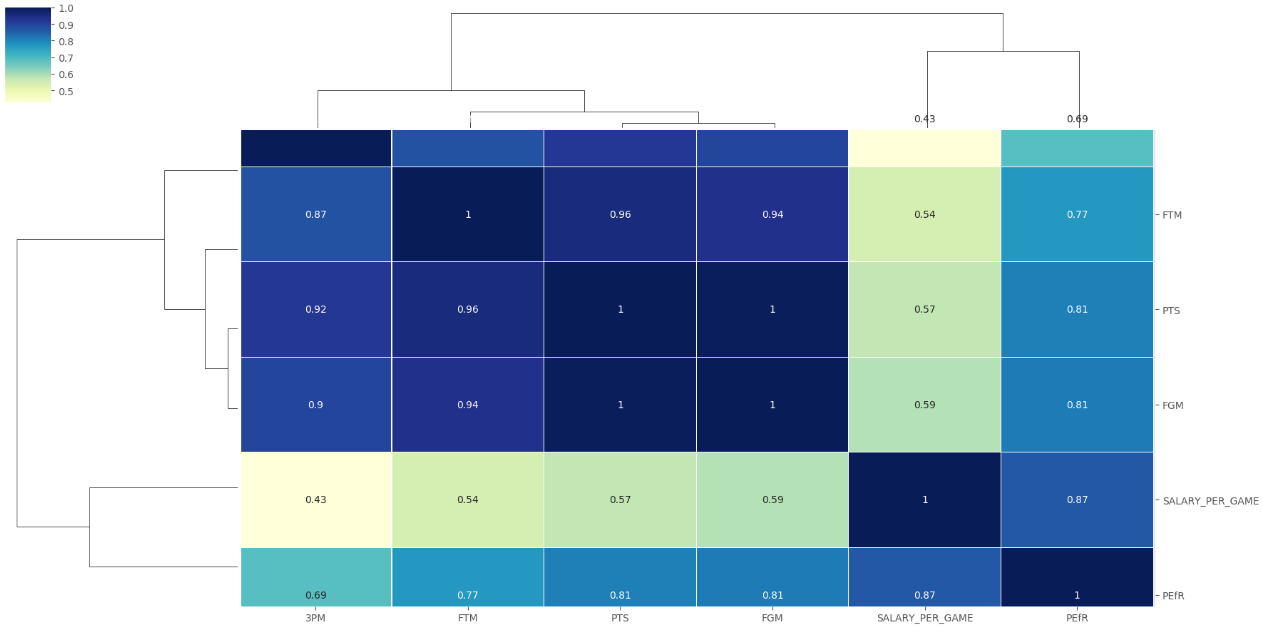
- All the columns are in an abbreviated form, the above link explains all the factors. Please refer to it in order to understand the data better.
- Note: all stats listed on the website are not used in this project.

- OFFENCE: Looking at how the columns co-relate with each other was important at this stage.

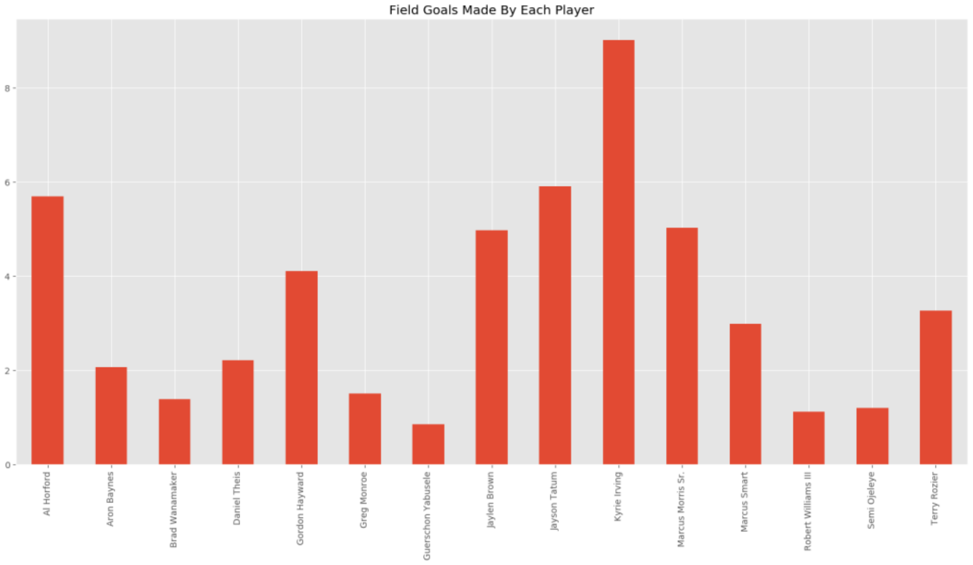
- The attributes which were selected were : PTS, 3PM, SALARY,FGM, FTM, PEfR.



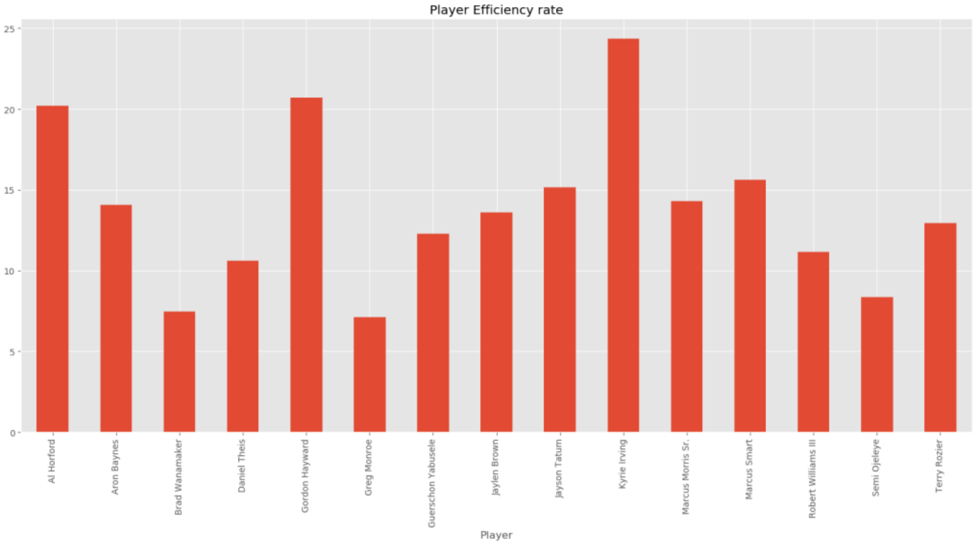
- We can see a lot of linear co-relations from the above scatter plot. A heatmap would be more informative.



- Exploring FGM and player efficiency rate.



-

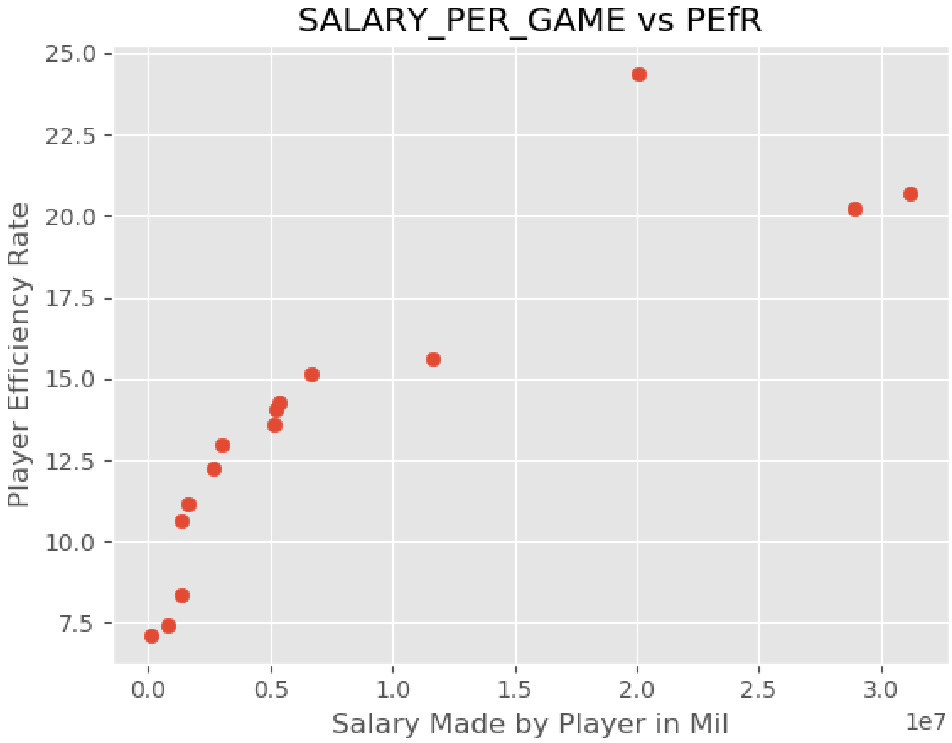


- we can see a direct co-relation between the two. The first graph explains the number of Field Goals made by each player. (Field Goals are 2 pointers or 3 pointers made by the player)

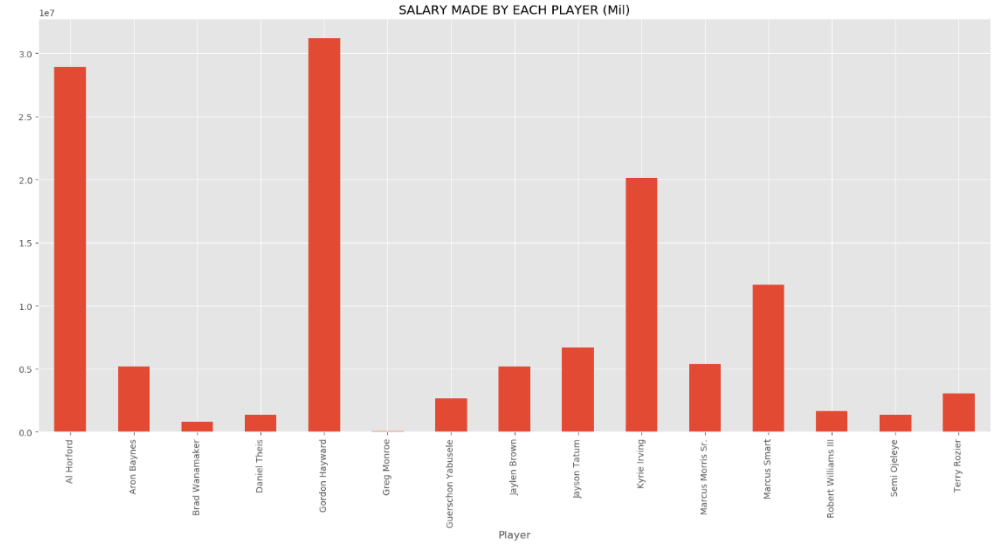
- The second graph explains the player efficiency rating.

- INFERENCE: We can infer that a player with high efficiency rating is more probable to get higher Field Goals.

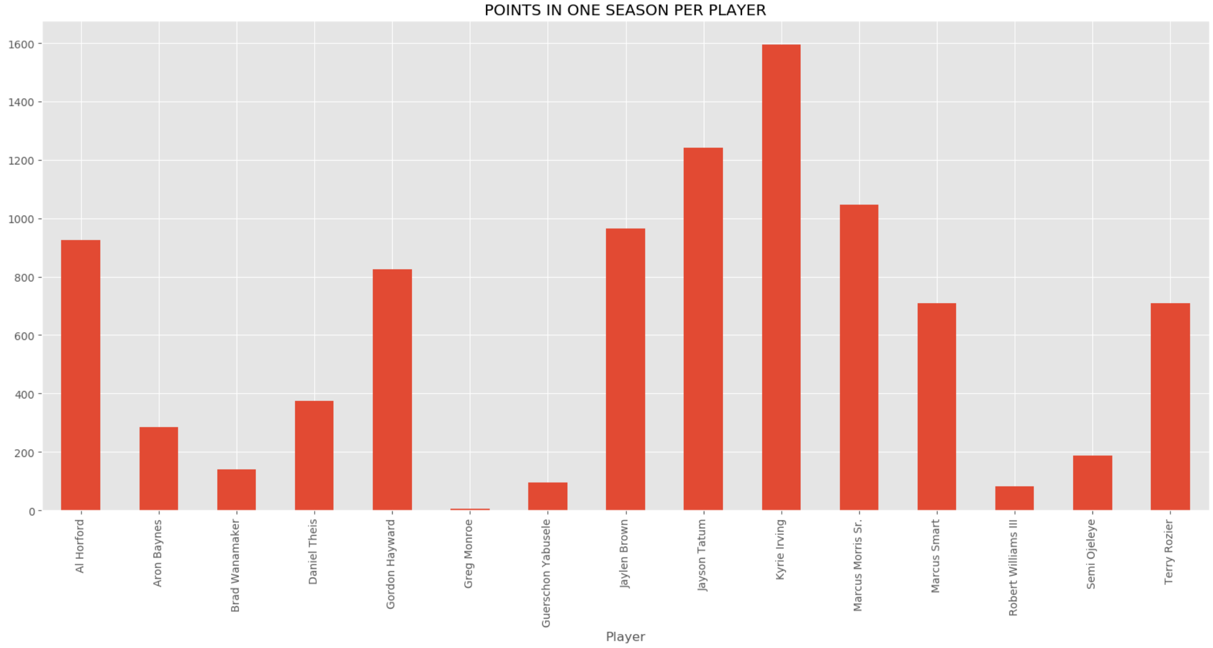
- Clearly player efficiency rate is an important factor. Lets see how player efficiency rate compares to salary made by the players.



- We can see that they are overall linearly dependent, but small discrepancies are found. Let’s explore these variations in more detail.
- INFERENCE : Player efficiency rate is important in determining the player salary.
- We can see that the player who’s making 2M is Kyrie Irving. He clearly is not making as much as other two players are, but has a higher player efficiency rate.

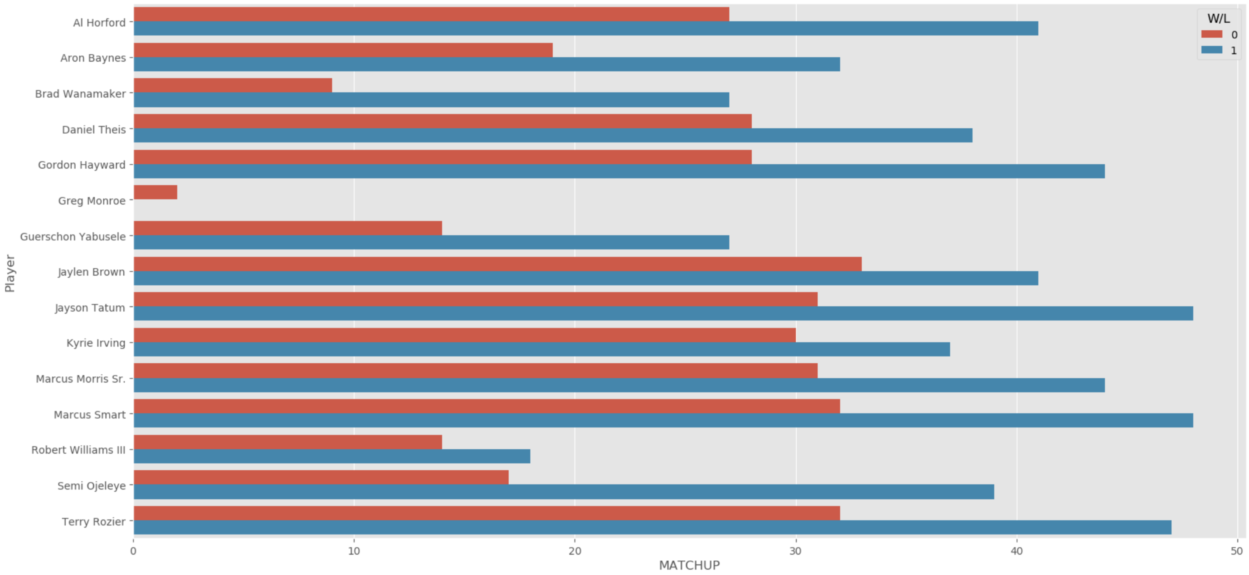


- Let’s find out the reason for his high efficiency rate.

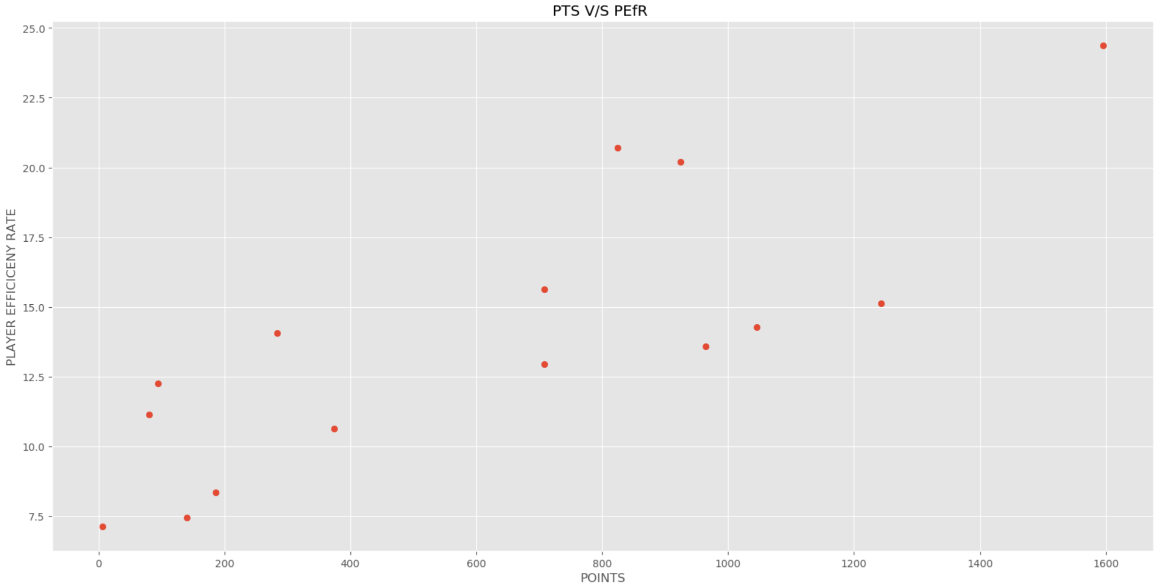


- We can see that Kyrie Irving has the highest points scored with respect to the whole team. One reason can be that he has played the most number or matches. It would not be reasonable to say that just because of his points he has a high player efficiency rate.

- Let’s look at how many matches each player has played, and won/ lost.



- We can see that Kyrie Irvin has not played the most number of matches, but he still has the highest points. So it would be safe to assume that points and player efficiency rate are related.



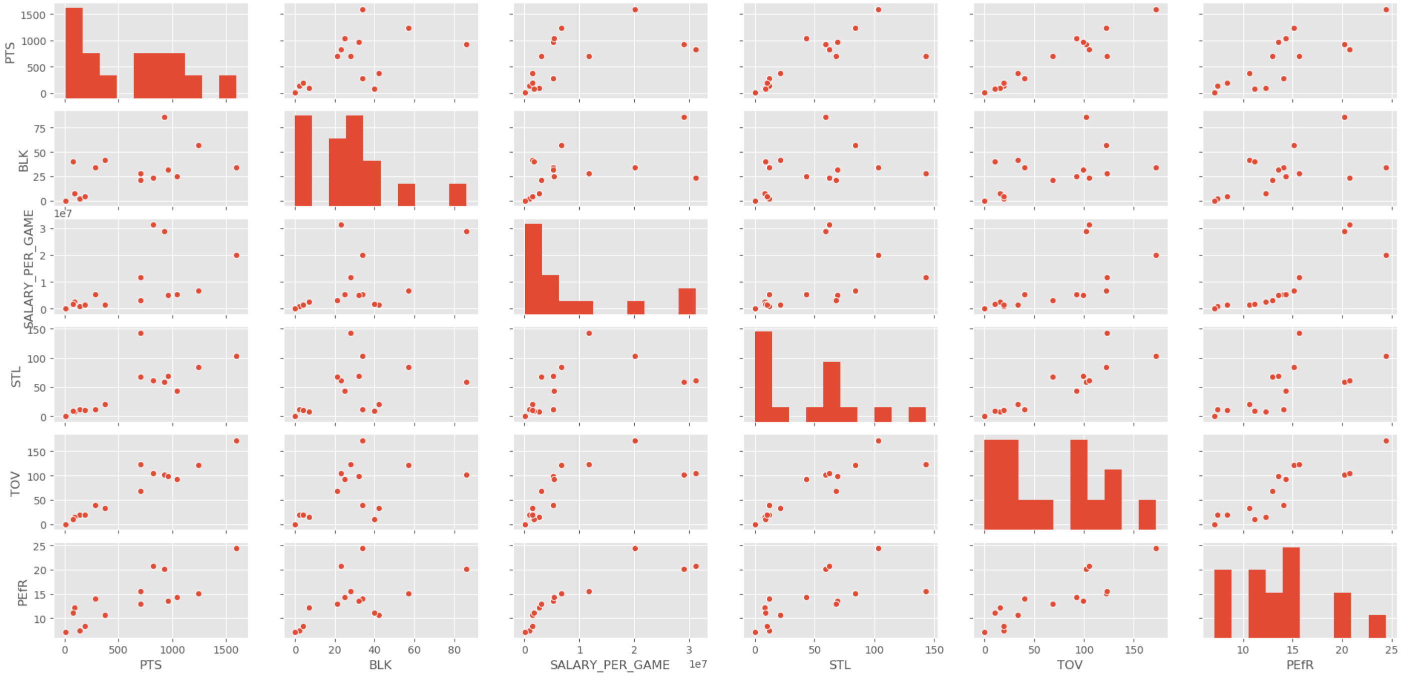
- We can see that they are positively co-related, but there are some discrepancies. A rational reasoning to this would be the position in which the player plays. A forward positioned player would score more points than the one in defence.

- INFERENCE : PTS and PEfR are both related to each other and are important in determining the value of a player.

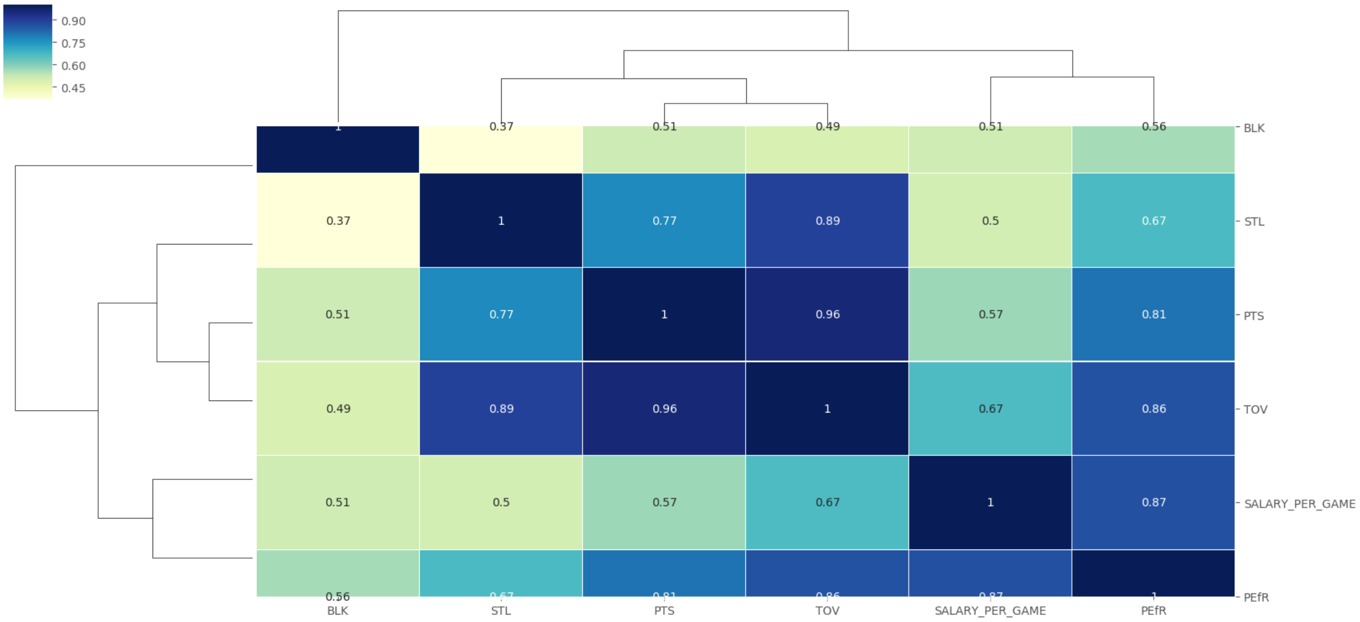
- INFERENCE: From the (6 x 6) scatter plot on page 11 we can see that 3PM are PTS has high co-relation, this implies more the three points scored more the points increased.

    - We can also infer that FTM (Free throw made) has high co-relation with FGM. This also makes sense as a player who has high free throw success with be good at shooting and hence will have more FGM.
    - We can also see the FTM and player efficiency rate have high co-relation. We can then conclude that player efficiency rate depends on FTM
    - We can also see the FTM and PTS are related to each other. This makes sense as both these events are dependent. Making a free throw will always increase the points scored.

- DEFENCE: lets look at defence now, how do these stats affect our two main attributes, points and player efficiency rate.

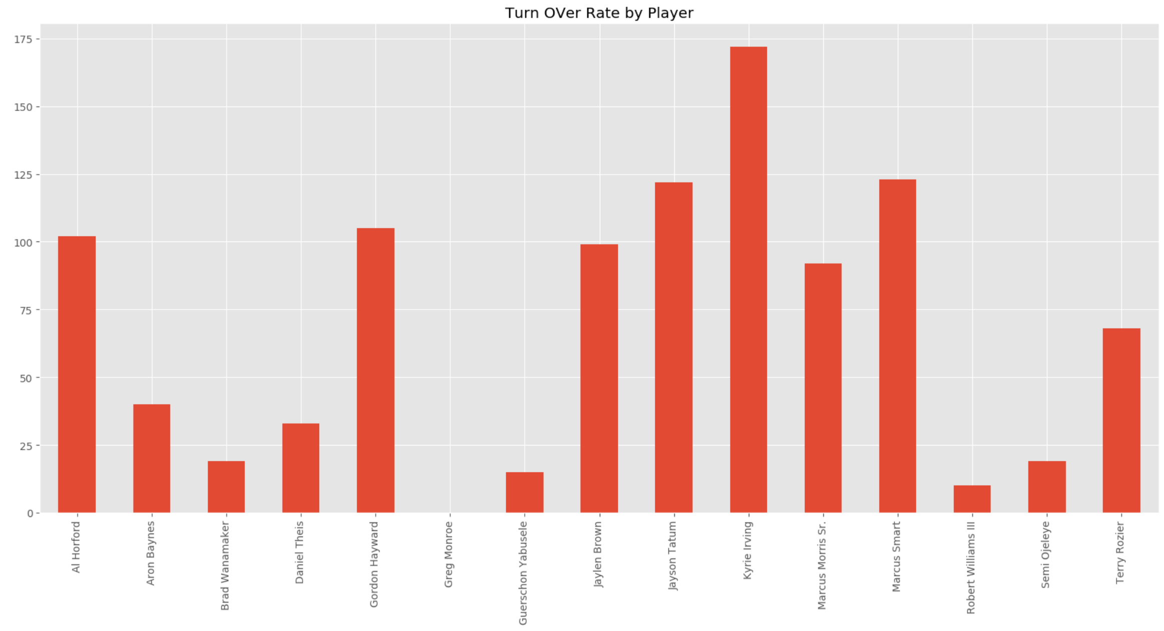


- A heat mapped co-relation matrix would be more informative.



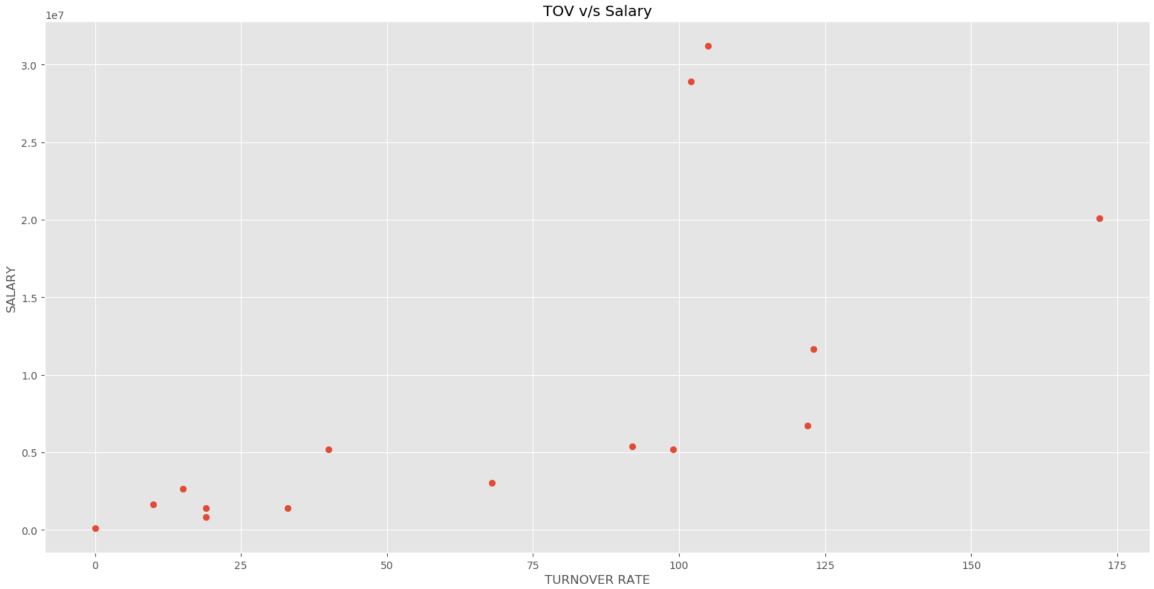
	BLK	STL	PTS	TOV	SALARY_PER_GAME	PEfR
BLK	1	0.37	0.51	0.49	0.51	0.56
STL	0.37	1	0.77	0.89	0.5	0.67
PTS	0.51	0.77	1	0.96	0.57	0.81
TOV	0.49	0.89	0.96	1	0.67	0.86
SALARY_PER_GAME	0.51	0.5	0.57	0.67	1	0.87
PEfR	0.56	0.67	0.81	0.86	0.87	1

- We can see that turnovers and player efficiency have a high co-relation. (TOV is when a player steals the ball from the attacking team to score points)
- Let's see who has the most turn overs.



- Its Kyrie Irving again. No surprises, he has the highest player efficiency.
- INFERENCE: player efficiency rate is dependent on turn over rate.

- Let's see how turnover rate depends on salary

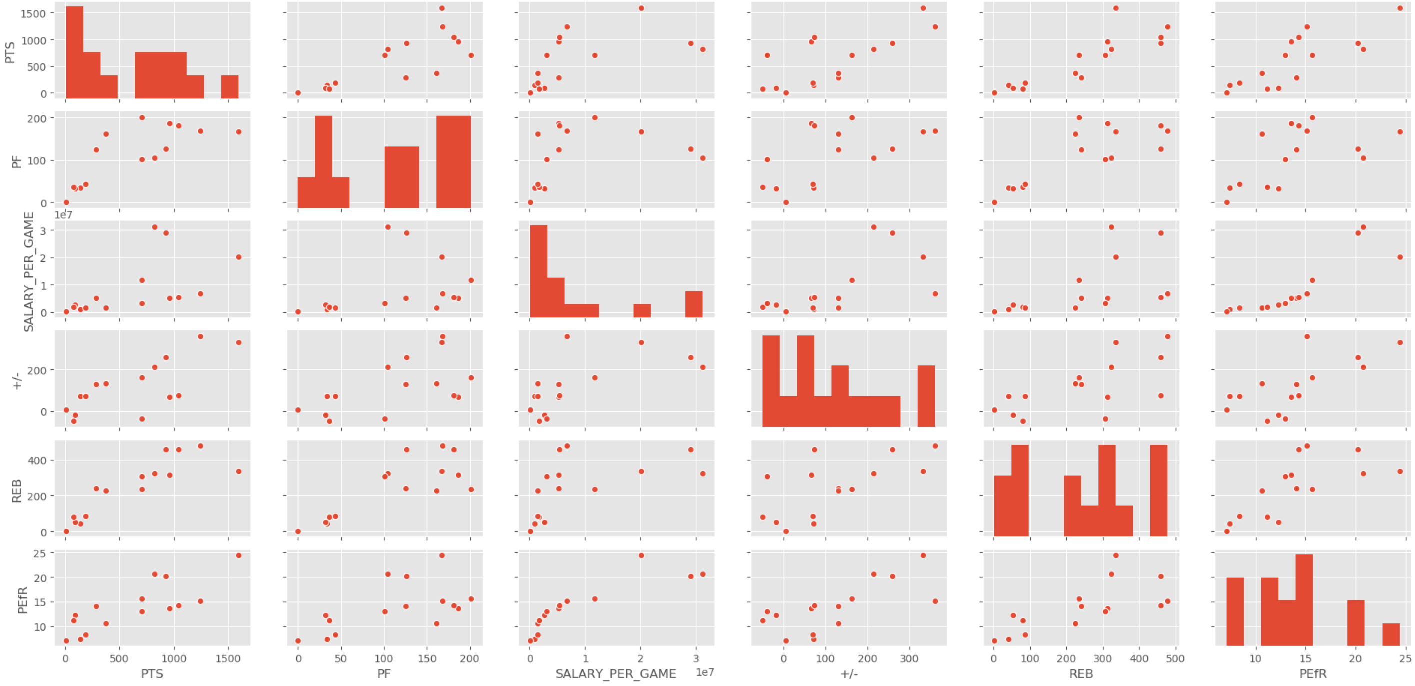


- We can see that they are linearly dependent, except one outlier. Which is Kyrie Irving.

- <u>INFERENCE: </u>From the (6 x 6) scatter plot on page 17 we can infer that:

    - Turn over and Steals have a high positive co-relation. This makes sense as both these events are related. A steal which then leads to scoring a point is called as turnover.
    - PTS and Turnover have a high positive co-relation with each other. Even this makes sense as a turnover leads to gain in Points.
    - Further we can see that blocks has no effect on any of the attributes.

- <u>REMAINING ATTRIBUTES: </u> It would also be useful to have a look at the remaining attributes.



<u>INFERENCE</u>: we can also see that rebounds and points have a good positive co-relation this makes sense as the more rebounds the player takes, more the points he scores.

1. we can also see that rebounds and PEfR have a positive co-relation this also makes sense as rebounds and points are co-related, so more the rebounds, more the points, and higher the player efficiency rate.

5) CONCLUSION:

From the above graphs we can see that a player’s salary is closely dependent on the Player efficiency Rate and the Points he scores in one season

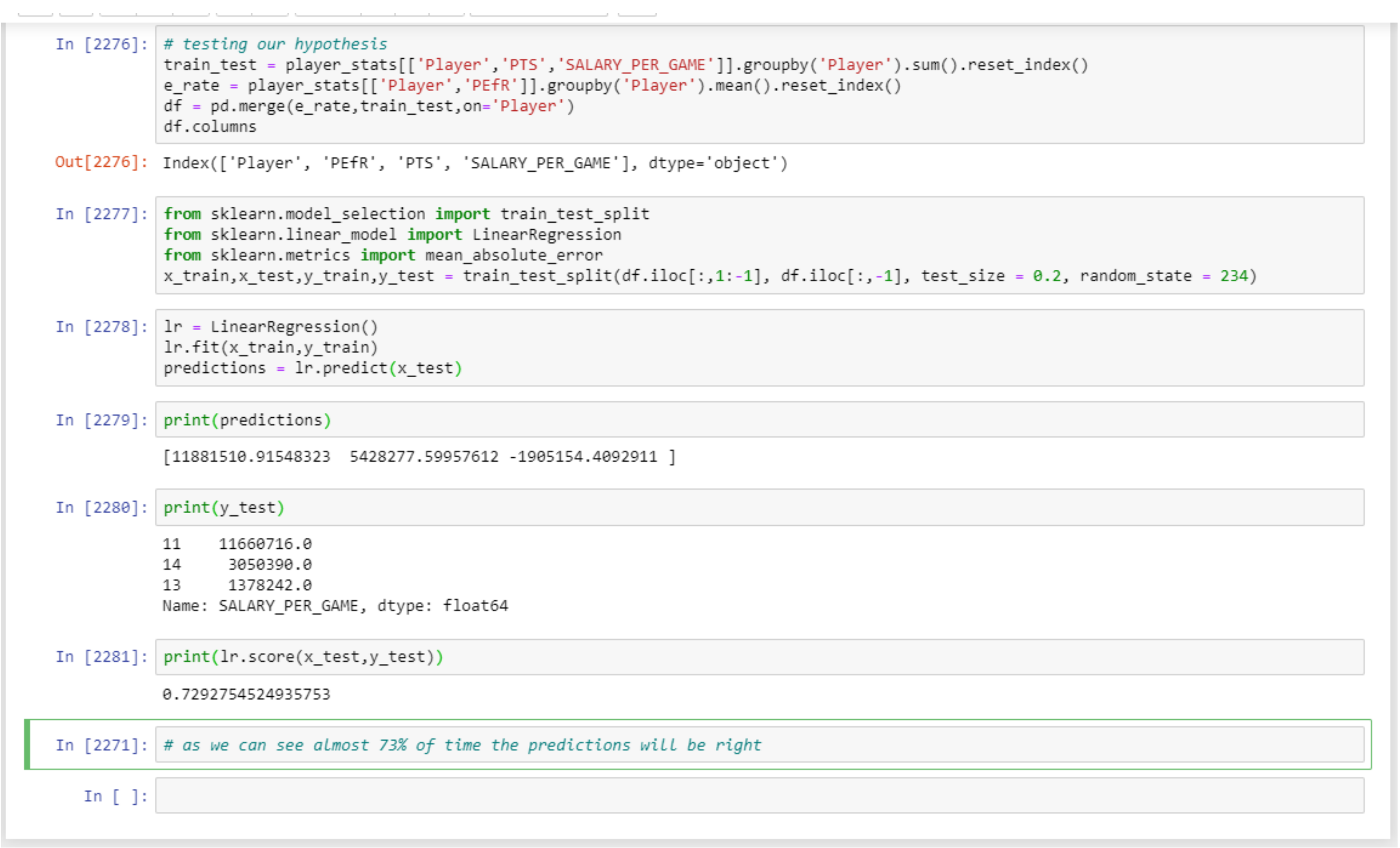
PEfR and PTS are further dependent on the FGM, FTM , 3PM (offence), REB ,TOV, STL (defence)

therefor a player’s salary is dependent on FGM,FTM,PTS,PEfR,3PM,REB,TOV,STL

6) HYPOTHESIS:

Player salary / value in our case is dependent on two main factors, PTS and Player Efficiency rate.

7) HYPOTHESIS TESTING:



```
In [2276]: # testing our hypothesis
           train_test = player_stats[['Player','PTS','SALARY_PER_GAME']].groupby('Player').sum().reset_index()
           e_rate = player_stats[['Player','PEfR']].groupby('Player').mean().reset_index()
           df = pd.merge(e_rate,train_test,on='Player')
           df.columns

Out[2276]: Index(['Player', 'PEfR', 'PTS', 'SALARY_PER_GAME'], dtype='object')

In [2277]: from sklearn.model_selection import train_test_split
           from sklearn.linear_model import LinearRegression
           from sklearn.metrics import mean_absolute_error
           x_train,x_test,y_train,y_test = train_test_split(df.iloc[:,1:-1], df.iloc[:,-1], test_size = 0.2, random_state = 234)

In [2278]: lr = LinearRegression()
           lr.fit(x_train,y_train)
           predictions = lr.predict(x_test)

In [2279]: print(predictions)

           [11881510.91548323  5428277.59957612 -1905154.4092911 ]

In [2280]: print(y_test)

           11    11660716.0
           14     3050390.0
           13     1378242.0
           Name: SALARY_PER_GAME, dtype: float64

In [2281]: print(lr.score(x_test,y_test))

           0.7292754524935753

In [2271]: # as we can see almost 73% of time the predictions will be right

In [ ]:
```

FUTURE WORK

We did not use the team_stats table for any conclusions, the team_stats table along with player_stats will give some interesting results.

For this project I only scraped data related to Boston Celtics, and completely ignored the other team. Compiling that data and performing analysis on it would also help us in determining the teams winning factors.

Down the line we could even use this data to predict a winning team in the NBA fantasy league. We can come up with winning strategies by analysing the opponent team.

Using the web scraper which I wrote, with some modifications, we could scrape the whole NBA website. Currently the web scraper takes a lot of time to run, but we could implement multithreading on it to increase its efficiency.

We can also use this data to predict a players efficiency rate for the next year.

REFERENCES

1) https://stats.nba.com/team/1610612738/boxscores/?Season=2018-19&SeasonType=Regular%20Season

2) http://insider.espn.com/nba/hollinger/statistics/_/year/2019

3) http://www.basketballinsiders.com/boston-celtics-team-salary/boston-celtics-salary-archive-2018-19/

4) https://www.geeksforgeeks.org/python-pandas-dataframe-where/

5) https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.groupby.GroupBy.apply.html

6) https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.bar.html

7) https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9

8) https://towardsdatascience.com/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-83a8f7ae2b4f

9) https://stats.nba.com/help/glossary/#fgm

10) https://stats.nba.com/team/1610612738/traditional/

11) https://www.kaggle.com/getting-started/27261

12) https://www.youtube.com/watch?v=2Pmf6Kqak3w&t=634s

13) https://seaborn.pydata.org/generated/seaborn.barplot.html

14) https://www.geeksforgeeks.org/python-pandas-dataframe-insert/

15) https://stackoverflow.com/questions/8575062/how-to-show-matplotlib-plots-in-python

16) https://stackoverflow.com/questions/34794067/how-to-set-a-cell-to-nan-in-a-pandas-dataframe

17) https://stackoverflow.com/questions/10373660/converting-a-pandas-groupby-output-from-series-to-dataframe

18) https://www.basketball-reference.com/contracts/BOS.html

19) http://insider.espn.com/nba/hollinger/statistics/_/year/2019

20) https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html

21) https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.MultiIndex.rename.html

22) https://stackoverflow.com/questions/30228069/how-to-display-the-value-of-the-bar-on-each-bar-with-pyplot-barh

23) https://stackoverflow.com/questions/24458645/label-encoding-across-multiple-columns-in-scikit-learn