INTRODUCTION

With ever increasing team payrolls, owners can usually afford to pay an underachieving player. The real issue is in determining the opportunity cost of this: What could you have gotten for that money instead? How else could you have improved your roster? It should come as no surprise then that teams are looking for alternative methods for determining appropriate pay levels for their athletes. This is where predictive analytics can hopefully shed some needed light on this very expensive problem as laid out below:

- What should an NBA player be paid based on his performance statistics?
- Which players are drastically overpaid?
- Which players are drastically underpaid?

We wanted to pull data from three different sources. Two of these sources were from basketball-reference.com[1-2] and the other was from espn.com[3]. Basketball-reference had one dataset that included common stats like points, three-pointers, two-pointers, etc. They also have a separate dataset with "advanced" stats that include fewer common stats like offensive win share, defensive win share, and player efficiency rating. The third dataset from ESPN contains the salaries of players.

BACKGROUND

We scraped data from each of the sites using Beautiful Soup. Each of the websites' URLs included the year the stats are from which made it easy to loop through several different years. The function we wrote takes in a list of years the user would like data from. It creates the URLs with the first year on the list and scrapes the data from each of the three websites and creates three temporary data frames. A column for players names exists on all three data frames so we can loop through each player and see if their name exists on all three data frames. If it doesn't, that name is ignored because we will be missing critical information. If it does exist on all three, we index those names, create an array that includes data on that player from all three data frames and that completed array gets added to the final data frame. This process is completed for every player and then loops through every year that was requested. At the end, we do some cleaning to deal with missing values and players that play on different teams in the same year. We have worked with data as far back as 2010 with this scraper. We haven't used data before this point to reduce the impact of inflation, changes to salary caps, and how the game has evolved over time.

From our original NBA Salary Prediction project, we were able to look at five different models in order to determine the best performing for our particular dataset. These are the models we considered:

1. Ordinary Least Squares (OLS)
2. Ridge Regression (RR)
3. Lasso Regression (LR)
4. ElasticNet Regression (ENR)
5. Extreme Gradient Boosting Regression (XGBR)

Table 1 shows the R^2 and root mean square error (RMSE) values for each model fit. As clearly indicated, the first four models had very similar results. The XGBR model performed significantly better, about a

30% improvement over the over four models we used, and that is the model we chose as the bet fitting model for our original project and it is the model we are using for this effort.

Table 1. Model Results Summary from Original Project

Model	R^2	RMSE
OLS	0.63	4398500
RR	0.63	4398502
LR	0.63	4398500
ENR	0.62	4432638
XGBR	0.92	1818766

GENERALIZATION

We followed a very similar procedure for this project as we did in the original effort. Here is a breakdown of the particular steps we took to prepare the data for the model and the model results for each year considered.

- **Loading the data**—the data were loaded into a Jupyter Notebook directly from a CSV file that was generated from the scraped data script. The resultant dataframe consisted of 3638 rows and 50 columns of data.

- **Formatting the data**—the scraped data had different column headers than what we needed for our model, so the next step was to change all of the column headings.

  Next, since we were interested in salary predictions based on year, we needed to separate the dataframe by year. This gave us 10 separate dataframes for the years 2010 through 2019.

  We then needed to separate the individual dataframes into a features and labels dataframe. The labels dataframes consisted only of the Salary columns, while the features dataframes contained all of the other variables.

  The next step involved reducing the features to only the variable we were interested in. These were determined in our original project after an extensive feature selection process. The final variables that are considered for this model are:
    - Age, Player Efficiency Rating, Blocks, Turnover Percentage, Steals, Assists Percentage, Games Started, Games Played, Total Rebounds, Field Goals, Defensive Box Plus Minus, Defensive Rebound Percentage, Usage Percentage, and Free Throws.

  The last step involved normalizing all of the data except the Salary column. This was done to change the values of the numeric columns to a common scale, without distorting differences in the ranges of values.

- **Running the model**—the model was run 10 separate times. One for each year range. The results of the model runs are shown in Table 2.

Table 2. Summary of Yearly Model Performance

Year	R^2	RMSE
2010	0.946	1099627
2011	0.949	1070874
2012	0.957	989,150
2013	0.942	1132491
2014	0.946	1174769
2015	0.940	1204874
2016	0.955	1147450
2017	0.940	1641827
2018	0.926	2124853
2019	0.923	2246051

- **Plotting the Top 10 overpaid and underpaid players**—once the model predictions were complete, we made a plot of the 10 most overpaid and 10 most underpaid players based on their performance statistics. Figures 1 and 2 show those plots for the 2019 year.

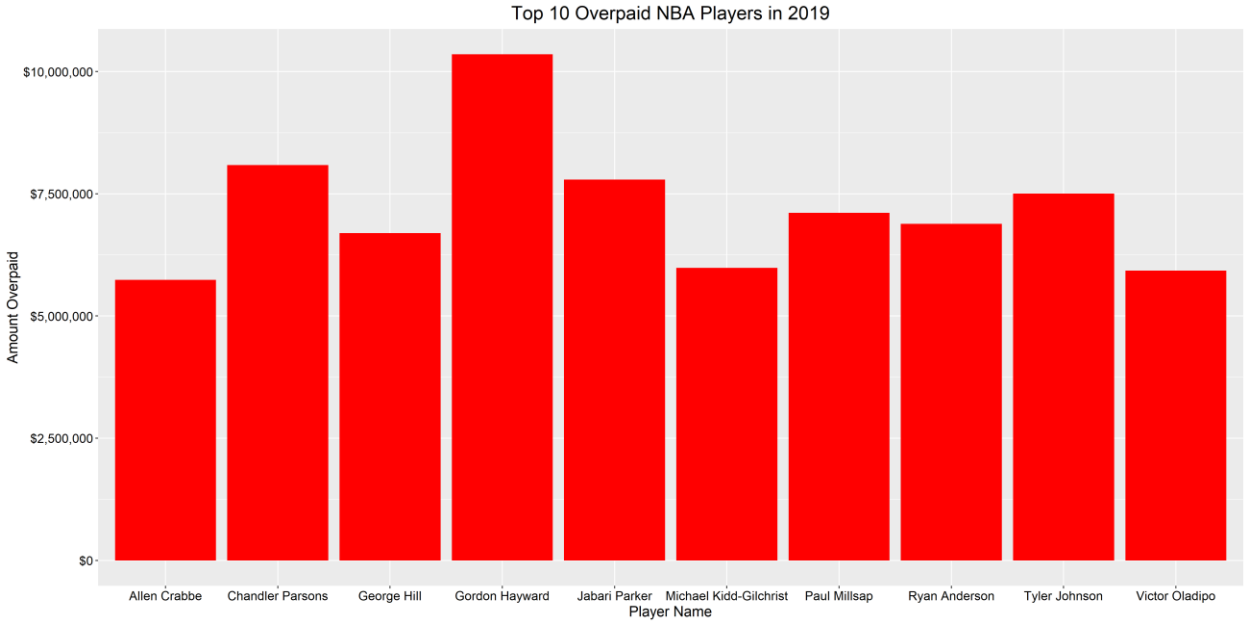

Figure 1. Ten Most Overpaid Players in 2019

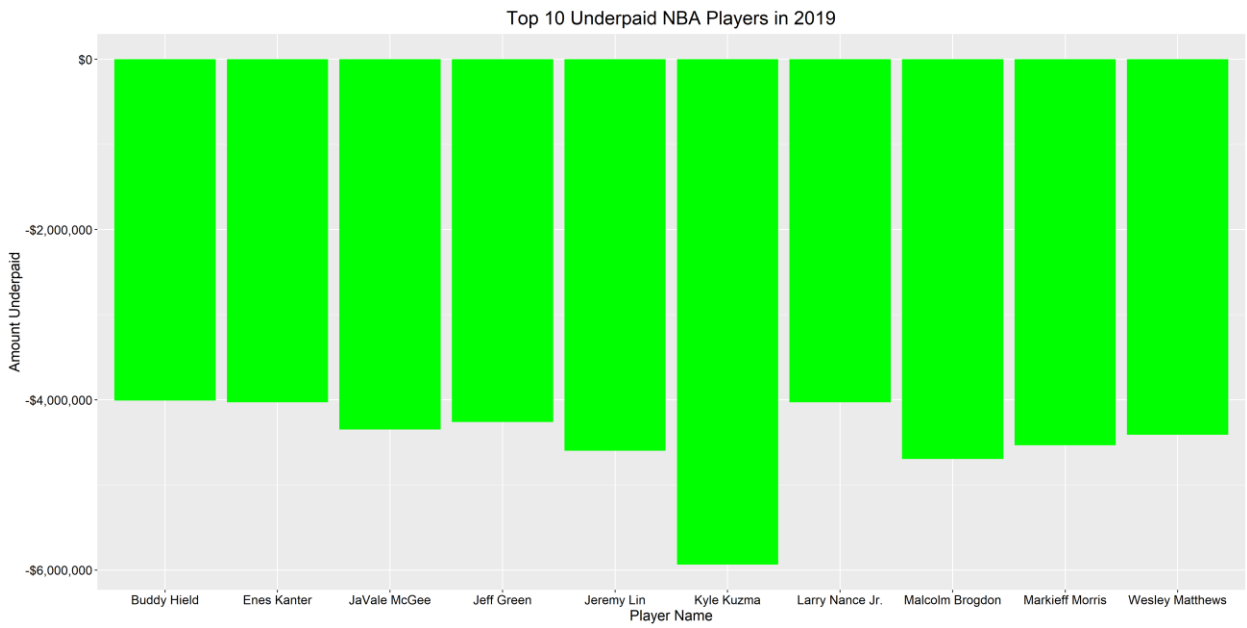

Figure 2. 10 Most Underpaid Players in 2019

CONCLUSIONS

The XGBR model provided excellent generalization to our new datasets. The R^2 values were high and the RMSE values were considerably lower than what we saw with the original four models shown in Table 1. One trend we did start to notice was that the model seems to be decreasing in performance in the last four years. The R^2 values are decreasing and the RMSE values are increasing. We are not sure at this point what is causing this, but it is an interesting trend and it grabbed our attention.

ACKNOWLEDGEMENTS

We are indebted to the communities behind the multiple open-source software packages and statistical reporting on which we depend.

REFERENCES

[1] Player stat totals for the 2018-19 NBA season. (n.d.). Retrieved from https://www.basketball-reference.com/leagues/NBA_2019_totals.html

[2] Advanced players statistics for the 2018-19 NBA season. (n.d.). Retrieved from https://www.basketball-reference.com/leagues/NBA_2019_advanced.html

[3] High and low player salaries in the NBA for 2019-2020. (n.d.). Retrieved from http://www.espn.com/nba/salaries