Noah Jankowski
MSU NetID: A56478374

NBA Playoff Estimation

1. Introduction

For the last decade, the use of statistical analysis in the professional basketball industry has been significantly growing. Recently, sports betting has taken off as it begins to be legalized in many states in the USA. This means that predictive analytics to evaluate team success before and after the season can lead to potential large earnings in betting, which is what this project aimed to do. I began by analyzing 538's ELO dataset, which was a rolling measure of team rating from the beginning of organized basketball in the 1900s. 538 created their own equation to measure this, giving every team an initial value and taking/awarding points after every game in a zero-sum way (winner would get 20 points, loser loses 20) based on margin of victory, home-court advantage, who was expected to win, etc. Using their dataset from GitHub, I recreated the main feature of their article from 2015, "The Complete History Of The NBA", which was an interactive time-series graph to visualize any team's rolling ELO score. After doing this, I was curious to see how well ELO could predict the playoffs. To do this, I went to another 538 article, "2019-20 NBA Predictions," which let me view the 30 NBA team's ELO scores right before the playoffs started. I compared their predictions to what occurred in real life, then created my own model. My model consists of two parts. The first part uses decision trees (with pruning and boosting) to guess if a team made the playoffs given their statistical resume from the regular season, and the second part uses regression techniques (multiple linear, ridge, and lasso) to predict how many playoff wins each team would have based on that same resume. The data I used for my models was taken from Basketball-Reference, but I had to combine it all into two CSV's that were then edited to make it more accessible. The training set consisted of all 30 NBA teams' regular season per game and advanced statistics from the 2015, 2016, 2017, 2018, and 2019 season. The testing set consisted of those same 30 teams' regular season per game and statistics from the 2020 season.

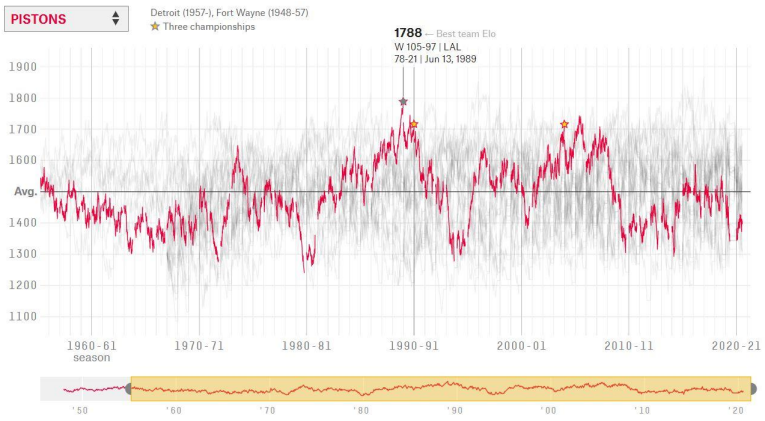
2. Related Work

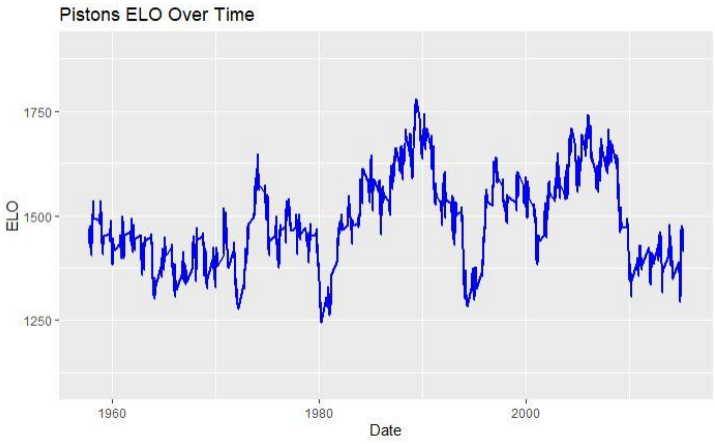
ELO was created in 2015 by FiveThirtyEight, as a way to measure how good a team was across multiple decades, allowing comparisons between teams that never had a chance to play each other, as well as predicting what teams could win the championship or make the playoffs in the future. Ratings are cumulative and update after every game of the NBA season. Winning teams take ELO points from the team they beat, so it is a zero-sum formula, and are awarded extra points if they were underdogs and/or by how much they win by. This allowed 538 to produce a rolling graph beginning at the inception of the team, so viewers could look at their success over time and how it changed. The average ELO is 1500, with really bad teams getting down to 1300 in a season and championship caliber teams getting past 1700. It is independent of the players on the roster, but to combat player movement a team's ELO rating is "reset" after the end of the playoffs every year, getting multiplied by .75, then having 376 added to the score. This determines the team's starting value for the next season (which will then get updated game by game).

538 used ELO in two different ways. When they first released this project, it was used as a visualization tool to look at team history. There is an interactive time-series plot at the top of the 538 article "The Complete History of the NBA", where you can choose which team (active or no longer playing) you want to look at, and analyze their trends as they were more or less successful. This is what I recreated, which is shown in section 3. After creating ELO, 538 began to use it as a predictive tool to predict a team's chances to make the playoffs and how far they'd get in the playoffs. I looked at their 2019-20 ELO playoff predictions after the regular season finished and analyzed how accurate it was. Their two title favorites were the Los Angeles Clippers and Lakers, given a 29% and 22% chance of winning the championship respectively. The Lakers ended up winning it all. In terms of the conference finals, two of their four picks were correct (Los Angeles Lakers and Boston Celtics). However, the model did not accurately predict the Milwaukee Bucks and the L.A. Clippers' playoff success: 538 gave them the 3rd best and best odds to win the title respectively, but both teams lost in the conference semifinals. The ELO model shined in the first round of the playoffs, where it correctly picked the winner of all 8 matchups. In the second round, it was only 1-3, and in the third round, 0-2. This was due to their poor ranking of the Miami Heat, who made it to the finals, but were given just a 2% chance of doing so according to the model.

While ELO is just a cumulative score from wins and losses throughout the season, I used machine learning to train regression models from the past 5 seasons to attempt to more accurately predict who would win in the playoffs, using team statistics. This will be seen in section 4.

3. Recreation





Seen above on the left is 538's graph of the Detroit Pistons' ELO from 1957 to present. My recreated model, using ggplot, is on the right, also showing the Pistons' ELO from 1957 to 2015 (the downloadable data on GitHub only went up to 2015). I also created a shiny site to put all 30 active NBA teams' ELO graphs with a drop down selection menu, similar to 538, but couldn't include it on this report because it is non-static and wouldn't work with a PDF.

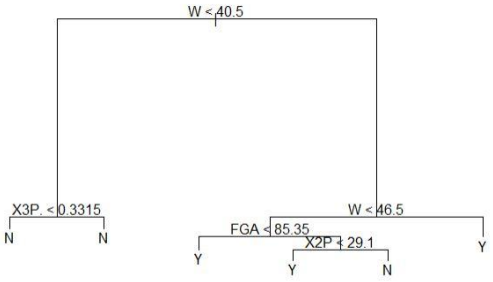
All of the data to create the above graph was taken from 538's "nba-elo" dataset on GitHub. After loading the data frame into R, I had to use the as.Date() function to change 538's dates to one that is more readable by ggplot. I then created a subset of the data, using the team_id notation, to look at the Detroit Pistons' ELO data from 1957 to 2015. After that I could use the geom_line() function with date

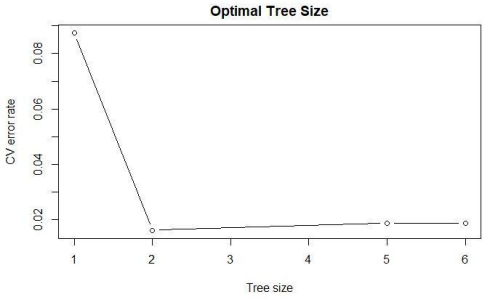
on the x and ELO on the y to make a visualization similar to 538s. To make it interactive, I used the plotly library to allow the viewer to zoom in on different time periods and view the ELO for any given date.

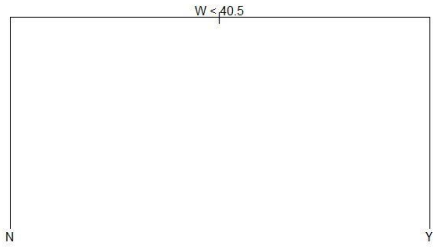
4. Designing a Model & Results

Although ELO did pretty well in terms of forecasting the future, I was curious to explore other methods we'd learned about. ELO is a fairly simple model as it only uses wins, losses, and margin of victory to calculate how good each team is. For my model, I wanted to predict two different things using a team's regular season statistical resume: whether they made the playoffs, and how many wins in the playoff each team would get. To get my datasets, I went on BasketballReference, seasons, summary, then for the 2014-15 through 2019-20 seasons, downloaded both the team per game stats and miscellaneous stats into an excel sheet. I then added two columns, one "Playoffs" and one "Playoff_Wins". Playoffs was a binomial variable, with Y being that they made it, and N being that they didn't. I changed it to 1s and 0s later in R. For Playoff_Wins, I went through the playoffs for 2015-2020 and gave each team their number of wins in the playoffs. Champions won 16 games (up to 4 series, winning 4 games to advance). I then converted the two excel sheets (training data was 2015-2019, testing data was 2020) to CSVs so I could read them into R.

To predict if a team would make the playoffs, I used a decision tree to analyze the binomial variable defined above (Playoffs). I defined the training and testing sets, then had to remove the team, year, and playoff_wins variables. I removed playoff_wins as it would make the decision tree too easy, because only teams that make the playoffs can get a playoff win. After that, I used the tree() function in the ISLR library to run a basic decision tree. The results are below. The summary() function showed that four variables were used in the tree construction: Wins, 3 pointers made per game, number of shots attempted per game, and 2 pointers made per game. The wins variable makes sense as the best predictor of who makes the playoffs, since the 8 teams with the most wins in each conference make the playoffs. Then the rest is how many shots the team made, and if you are attempting more, you'll make more. After creating the tree, I used the predict() function on the testing data to see how accurately the model could predict if a team made the playoffs or not. It did alright, accurately guessing 23/30 (77%) teams outcomes. The 7 incorrect classifications the model got wrong were predictions that the team did not make the playoffs when in reality they did. I then performed a cross-validation tree to prune it and see if I could increase the accuracy, but pruning made the tree smaller such that the only variable used was wins. That tree is also shown below. It did result in a more accurate prediction though, guessing 27/30 (90%) of the correct outcomes.

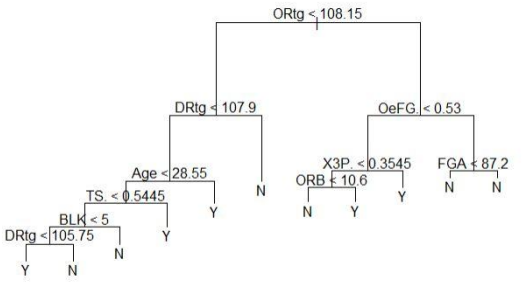


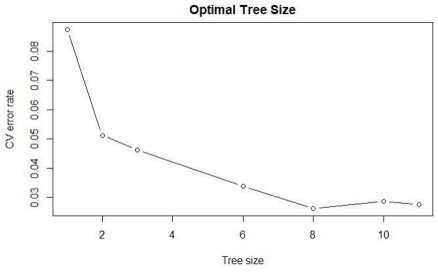


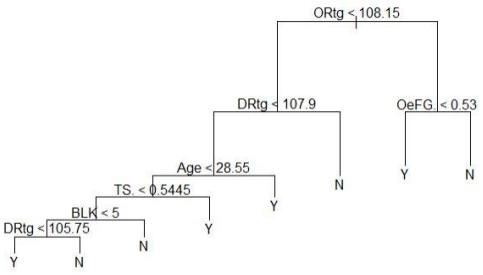


Initial Decision Tree Cross-Validation to find optimal size Optimal sized tree

Although these trees make sense, I was more interested in finding if there were specific stats that could be used to predict if a team made the playoffs. So I did the same methods as above, but I masked out Wins, Losses and NetRtg to see if I could get a decision tree that was more interesting. In the pre-pruned tree, there were 9 variables used: ORtg, DRtg, Age, TS%, Blocks per game, OeFG, 3PM per game, Offensive rebounds per game, and FGA per game. ORtg and DRtg are measures of how good a team's offense or defense is per 100 possessions (ORtg is points scored per 100 possessions, DRtg is points allowed per 100). OeFG% is how well the opposing team shot. The resulting tree is shown below. The prediction on the testing data worked out pretty well, returning the right prediction on 26 out of 30 teams (87%). After pruning, the optimal tree size was found to be 8, and when plugged into the tree() and predict() function, it also returned 26 out of 30 correct classifications. However, this time it incorrectly predicted a team to make the playoffs when it really didn't. That was the first time that type of error was made in the predictions.







Initial Decision Tree Cross-Validation to find optimal size Optimal sized tree

The last decision tree method used was boosting with the GBM library. Similar to the last tree, I removed playoff wins, wins, losses, and net rating from the dataframe to get more insight into other statistics. I did boosting with a shrinkage rate of .01 and 1000 trees, then predicted the results with the testing data. An arbitrary value of 0.45 was set, implying that if the boosted tree gave a team a 45% chance or better of making the playoffs, then they would make it. The resulting prediction returned a 24/30 accuracy rate, or 80% success. Compared to a simple linear model using the glm() function, with the same parameters, the boosted model did better. The four most influential variables, according to the boosted prediction model, were ORtg, DRtg, opposing eFG%, and average age. The results are shown below.

pred_boost		
	0	1
0	10	4
1	2	14

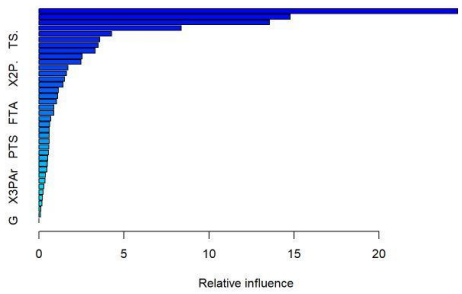
Boosted Prediction Table

lm_pred		
	0	1
0	12	2
1	5	11

Linear Model Prediction Table

var <fctr>	rel.inf <dbl>
ORtg	24.62048794
DRtg	14.77008531
OeFG.	13.57193875
Age	8.36044242
FT.	4.27065222
TS.	3.57999110
X3P.	3.48164813
STL	3.30523845
eFG.	2.53406410
BLK	2.47944077

Relative Influence Table



Relative Influence Graph

After working with decision trees, I created a few different regression models estimating how many playoff games a team would win, given their regular season statistics. I started with a simple multiple linear regression model, then used both ridge and lasso regression to try to get my predictions more accurate.

The multiple linear regression model did not perform well. The training data was modeled with the glm() function, then the resulting model was used to predict the results on the testing set. The resulting data frame is shown below. Although no team was projected to get close to 16 wins, achieving those results with a linear model and small sample would be very difficult. The success of the model was measured such that if the model had a team ranked higher in expected playoff wins, then that team would be favored against all others below it. Using this method, in the first round, the linear regression model would have gone 4-4 in predicting winners. The championship team, the Los Angeles Lakers, were predicted to win only 2.2 playoff games in this model. The r-squared for the model was .03 which is really low and suggests most of the variation wasn't explained.

	Team <fctr>	Predicted Playoff Wins <dbl>	Actual Playoff Wins <int>
29	UTA	10.4324335	3
28	TOR	8.4284053	7
7	DAL	7.7154643	2
16	MIA	7.5293847	14
2	BOS	6.5309453	10
23	PHI	6.0583806	0
13	LAC	5.9212404	7
17	MIL	5.1402010	5
25	POR	4.3939413	1
8	DEN	3.8772899	9

1-10 of 30 rows

Predicted Playoff Wins vs. Actual for Linear Regression Model

To get better results, I decided to try both ridge and lasso regression models. The first thing I did was use 5-fold cross-validation on the training data to find the best lambda values, then take the best one

using the 1 standard error method. After finding the best lambda value, I used the glmnet() and predict() functions with the testing data to predict each team's number of playoff wins for the 2019-20 season. It ended up being more accurate than the linear model above, returning an r-squared of 0.347. The ridge model was actually 7-1 in predicting the correct first round winners, and did have the Los Angeles Lakers as the third most likely team in the league to win it all. The resulting data frame for the ridge predictions is seen below.

	Team <fctr>	Predicted Playoff Wins <dbl>	Actual Playoff Wins <int>
17	MIL	7.53355389	5
28	TOR	6.10623382	7
14	LAL	5.84784158	16
13	LAC	5.61839114	7
29	UTA	5.53669391	3
16	MIA	5.50019802	14
7	DAL	5.43711406	2
11	HOU	5.40220403	5
2	BOS	5.14191841	10
23	PHI	4.44574510	0

1-10 of 30 rows

Predicted Playoff Wins vs. Actual for Ridge Regression Model

I did the same exact technique for lasso regression, and it returned with the best r-squared value of the 3 techniques at 0.41. This makes sense because there were a lot of predictors in this dataset and lasso is about shrinkage. Lasso regression was also 7-1 in predicting the first round winners, and also had the Lakers as the 3rd most likely team to win the championship.

	Team <fctr>	Predicted Playoff Wins <dbl>	Actual Playoff Wins <int>
17	MIL	6.6549048	5
28	TOR	5.8392157	7
14	LAL	5.6034277	16
13	LAC	5.2075568	7
16	MIA	4.8474437	14
29	UTA	4.7880230	3
2	BOS	4.7142420	10
11	HOU	4.3597455	5
8	DEN	4.2542615	9
7	DAL	4.1963393	2

1-10 of 30 rows

Predicted Playoff Wins vs. Actual for Ridge Regression Model

5. Conclusion

Based on my analysis for both 538's ELO method and using different regression models, it seems like it is difficult to accurately predict who will win an NBA championship solely based on a regular season statistical resume. However, it is easy to predict whether they simply made the playoffs or not using a decision tree or similar method. This makes sense because basketball is a sport with a lot of variability. There are injuries that can't be accounted for in these numbers, different matchup issues between some teams (the Lakers might find it very easy to beat the Clippers, but struggle against the Nuggets for instance) that you can't adjust for in a model like this, etc. I think in order to more accurately predict who could win the championship the prediction model would have to go game by game, and use player statistics as well as team statistics for peak performance. It might seem like 5 years of training data is too small for a testing set that's 1 year large, but I actually don't think there is a huge issue with this because the NBA and teams' playstyles are rapidly changing. Creating a model that incorporates data

from 10 years ago would not accurately represent how basketball is played today. If I were to continue this project I would add a bootstrap model and see how that performed in terms of guessing playoff wins. This was a very interesting process and I'm satisfied with the results, even though they didn't end up being as accurate as I had hoped.

Citations

FiveThirtyEight. "The Complete History Of The NBA." *FiveThirtyEight*, 7 Dec. 2015, projects.fivethirtyeight.com/complete-history-of-the-nba/#pistons.

Silver, Nate. "How We Calculate NBA Elo Ratings." *FiveThirtyEight*, FiveThirtyEight, 21 May 2015, fivethirtyeight.com/features/how-we-calculate-nba-elo-ratings/.

FiveThirtyEight. "2019-20 NBA Predictions." *FiveThirtyEight*, 12 Oct. 2020, projects.fivethirtyeight.com/2020-nba-predictions/.

"Fivethirtyeight/Nba-Elo/Data." *GitHub*, github.com/fivethirtyeight/data/blob/master/nba-elo/README.md.

Basketball statistics and history. (n.d.). Retrieved April 19, 2021, from https://www.basketball-reference.com/