COS 424, Final Project

Jake Reichel
AB COS ’19
jreichel@princeton.edu

1 Abstract

In this paper I examine the factors that determine who gets vote shares in the NBA MVP race. By training both a Random Forest and an AdaBoost model, I show that: (1) winning percentage is very important to the voters, (2) narrative may not be as important as some may think to winning MVP and (3) James Harden should have won the 2017 MVP award. I then also use these models to predict that while Giannis Antetokounmpo will mostly likely win the 2019 NBA MVP award, the vote will be much closer than current betting odds predict.

2 Introduction

Every year, the National Basketball Association (NBA) gives out its Most Valuable Player (MVP) to the player who receives the highest number of “vote shares” from a group of voters who collectively select the winner. Each member of the panel votes for who they believe is deserving of the award from first place through fifth place. A first place vote is worth 10 points, second place is 7 points, third place is 5 points, fourth place is 3 points and fifth place is 1 point. The player with the highest point total after the voting is granted the MVP award.

In this paper, I discuss the building of a regression-based predictive model that calculates the odds of who will be declared the NBA MVP each season. While a model could be created to simply predict if a player would or would not win MVP, a more robust prediction requires an estimate of how many vote shares each player will receive. Building on earlier assignments, I intend to use multiple models, such as Linear Regression (LR), Random Forest Regressor (RFR), and AdaBoost (ADA), to predict the percentage of available vote share points that every player will receive, that way if more panelists are added, the underlying model can remain the same.

Additionally, I plan on answering the following questions: Which statistics are the most important for predicting MVP? How important is narrative to an MVPs case? Can statistics alone predict MVP? Does the players teams record matter? (i.e. can a great player on a bad team win MVP?) Who will win the 2019[1] NBA MVP?

If my model is successful in answering the above questions, it has potential future use for gambling purposes to either find good opportunities in Vegas futures market for NBA MVP, or to drive the pricing of such futures.

3 Related Work

Mine will not be the first attempt at trying to use machine learning within this problem space. There have been attempts to do a similar thing within Major League Baseball [9], and for many years, there have been Vegas betting lines for NBA MVP futures that have been determined based on statistical models. Additionally, there have been multiple more targeted attempts, ranging from student projects done at University of Virginia [7] to individual Github repositories [10], to a data analysis blog dedicated to basketball [3] to build machine learning models to predict NBA MVP.

As far as I can tell, my project will be the first of them to specifically predict NBA MVP Vote Shares, and the models above all returned mixed results, with the best being about 65% accurate. I hope to build upon the work of these earlier projects to create a more accurate and robust model.

[1] Typically, seasons span two years. Thus, the ongoing season is the 2018-2019 NBA Season. However, for brevity, I refer to each season in this paper by the latter year only.

4 Data

4.1 Dataset Creation

In order to predict MVP, I needed to combine data from a variety of sources. Firstly, I downloaded a dataset from Kaggle [4] that included all NBA player statistics, both traditional (points, rebounds, steals, etc.) and advanced (Value Over Replacement Player (VORP), Effective Field Goal Percentage (eFG%), etc.) statistics, from each of the 1950 2017 seasons. Next, I manually created a CSV file for both traditional and advanced statistics taken from `basketball-reference.com` for the 2018 and 2019 seasons, and appended it to the earlier dataset.

I then created a CSV of every NBA team's record (wins, losses) from each of the 1981-2019 seasons. Lastly, using `basketball-reference.com` I created a dataset of the how many MVP vote shares every single NBA player has received for each of those seasons, and appended that as a final column to the player dataset.

4.2 Data Cleaning - Player Statistics

The CSV files that I downloaded and created were far from ideal and required significant preprocessing. Before the 1980 season, the NBA had yet to introduce the three-point shot to the game. Thus, any seasons before then would not include any data on how well players shot a three pointer, which is now a major facet of the game[2]. Additionally, many traditional and advanced statistics are not available for seasons before 1980. Thus, I trimmed the player dataset to 1980-2019.

All missing values were imputed with 0s, as that would generally indicate lack of performance in that field. For example, a player who is missing information about their three-point shooting did not attempt a three-point shot that year, so 0's are appropriate.

4.3 Data Cleaning - MVP Vote Shares

I then further trimmed the player dataset to begin in the 1981 season, as that was the first year in which MVP was determined by a media panel, instead of by players, meaning that the consistency of voting changed. Every year, additional members of the media are added to the MVP panel, thus the total number of "points" that each player receives is not important from year to year. Rather, it is about the "share" of votes received. That is, the percentage of possible points that each person got (each person eligible for full amount). Thus, in my dataset, I changed MVP votes to the vote share. Additionally, there were some players in each year who received a minimal number of votes. However, to prevent the model from being influenced by a long tail or a "hometown" voter, for all players who received less that 1% of the vote share, their "share" column was imputed with 0's.

4.4 Data Cleaning - Team Records

In determining the MVP, I figured that the player's team record could be a significant factor. Thus, I needed to match up each team's record from 1981-2019 with the players from the player database. This proved to be a difficult endeavor, as team names and abbreviations (i.e. New York Knicks are NYK) have changed over the course of time, and to do a "join" we need to ensure team names are unique. For example, the Charlotte Hornets (CHA) became the New Orleans Hornets (NOH) and then became the New Orleans Pelicans (NOP). Meanwhile, after the Hornets moved to New Orleans, the Charlotte Bobcats (CHA) were founded and later became the Charlotte Hornets (CHO). Thus, joining this dataset with the earlier one required manual care. Lastly, just like with MVP Vote shares, the total number of wins is not necessarily important, as there were a few seasons with a lesser number of games. Therefore, the team's winning *percentages* for each season were used.

5 Methods

In this paper, I evaluate the performance of two different scikit-learn [8] models, ADA [1] and RFR [2], on different variations of the dataset (as described below). I will continually compare them to one another, as well as to a general baseline performance of a LR model. To ensure that I could both properly interpret the results and achieve maximal precision of the model, I used five forms of the player dataset, described in order of execution.

[2] For example, recent winners of the MVP award, Stephen Curry (2015, 2016) and James Harden (2018), are known for their exemplary three point shooting.

Firstly, there is the “full” dataset. This is the entire dataset described in Section 4. It comes in two forms – one with winning percentage included and one without. In this manner, I can test to see if winning percentage plays a factor in model performance. Secondly, there is the “reduced” dataset. This is a dataset comprised of *only* the players who have received more than 1% of the vote for that season. By limiting the data, I wanted to see if the models would perform better with less noise and, perhaps, be able to tell the differences between the top level of players, as generally, voters only truly decide between a small fraction of the players who have played that season.

Third, I noticed that many of the columns had strong correlation with one another (see Figure 1, stronger correlation is in red – column labels not important to reader). For example, there is a column for total rebounds, another column for defensive rebounds, and another column for offensive rebounds. Obviously, total rebounds is the sum total of the other two. Therefore, if we were to delete any one of these 3 columns, we would still have the same information and would give the models fewer decisions to make.

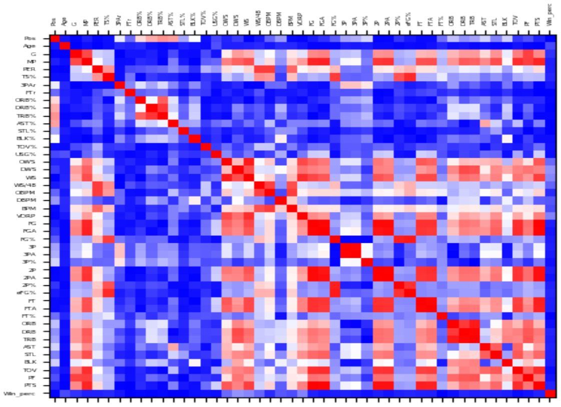


Figure 1: Correlation Matrix, Full Dataset

I deleted 8 columns in this way, as well as 2 others that did not seem to contribute much information – the player’s age and the position he played. This dataset is referred to as the Redundancy-Removed (RR) dataset.

Fourth, I used *Standard Scaler* to mean-center the data and divided each column by its standard deviation, to create the “scaled” dataset, that way the relative performances of each season could be seen. That is, each season’s data was taken its own dataframe, scaled, then reinserted to create a new dataset. In the scaled dataset, each player’s statistics would then represent the number of standard deviations above the mean that the player had in that statistic, *for that year*.

For the first four models, the data was then split into a training and a test set, where the 1981-2011 seasons were considered the training data, and the 2012-2018 models were treated as the test data. The fifth dataset, however, is the “Yearly” dataset, in which the models were created in a loop, being trained on all of the individually scaled seasons combined, with one scaled season held out, and then evaluated. For this approach, the evaluations are either season-by-season or a total average (see Results).

For all instances of ADA, I used a *DecisionTreeRegressor* as the base estimator, with $n\_estimators$ set to be 5000, chosen as performance did not seem to increase by much with more than 5000, but timing degraded majorly. For *RFR* this number was 750. The *DecisionTreeRegressor* was further specified to have a maximal depth of 30 for the same reason given above as well as to reduce memory consumption of an unnecessarily large tree. All other hyper-parameters used were the defaults.

6 Results

For clarifications on what each of these subsections represents, see section 5. After discussing the results of each dataset, I will explain the takeaway from it, and how it impacted the development of future iterations of the models.

6.1 Full Dataset

When trained on the full dataset, the models had varying degrees of success. The baseline LR model had an unimpressive mean squared error (MSE) of $2.21 * 10^{-3}$ and an R-squared (R2) value of 0.29. This model did not perform any differently when the team’s winning percentages were added. However, the RF and ADA models performed much better than the baseline LR, even on this basic dataset. RF had a MSE of $9.6 * 10^{-4}$ and R2 of 0.69.

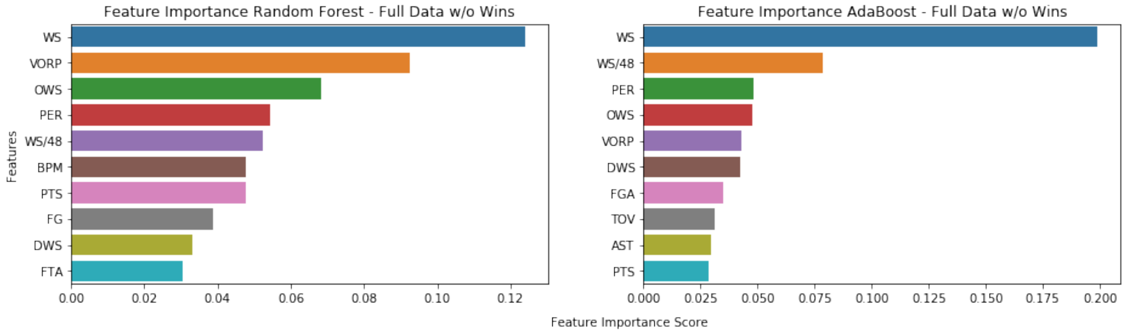


Figure 2: Top 10 Feature Weights on Full Dataset

Taking a look at some of the feature importances of the models (Figure 2), it appears they focused on statistics that would make a lot of sense towards predicting MVP, such as Win Shares[3] (WS) and VORP. Once I added the win percentage column to the full dataset, the MSE of the RF model fell to $8.7 * 10^{-4}$ and the R2 jumped to 0.71. The ADA model had MSE of $9.1 * 10^{-4}$ and R2 of 0.71. These metrics improved to $7.8 * 10^{-4}$ and 0.75 when wins were added.

Thus, from the full dataset, I primarily learned two facts: RF and ADA are good models for this problem space, and the winning percentage column (Win_perc) is very important to the models.

6.2 Reduced Dataset

Models trained on the reduced dataset saw a drastic shift in performance, in an unfortunate way. The logic behind the reduced dataset, as explained in Section 5, was to limit the amount of data that the models had to process to try to limit some of the noise. In practical terms, these models would hopefully do a better job at emulating what the voters realistically do – choosing between a number of top candidates each season, not necessarily the entire player pool.

On the baseline LR, this logic worked very well, as linear regression generally sees a performance increase with fewer decisions to make. While the MSE tremendously rose to $4.28 * 10^{-2}$, the R2 also doubled to 0.58, showing that much more of the variance was accounted for.

However, the RF and ADA models fared much worse on the reduced dataset with respective MSEs of $4.91 * 10^{-2}$ and $4.08 * 10^{-2}$, a 100x increase over the full dataset, and respective R2s of 0.52 and 0.60, approximately 25% decreases. This is because more data points enhance decision trees' ability to split the data based on important value thresholds.

From the reduced dataset, we thus learned that the best performing models will learn from data from all players, not only the vote-getters. Therefore, all efforts afterwards were on some altered version of the full dataset.

6.3 RR Dataset

As expected, taking the full dataset removing many of the redundant columns did improve the model performances. However, the improvements were rather minor – about 1-2% improvement across the board for each of the models in both MSE and R2. It did heavily improve model interpretability, as there were fewer columns that were not as collinear, so I used the RR dataset as my new source of data for all later models.

6.4 Scaled Dataset

All datasets before the scaled RR dataset assumed that across seasons, there was a limited amount of variability. They were used to train the models in a way that did not account for differences on a per-season basis in how the voters voted, the players performed, or for meta-information such as the number of games played in that season. However, accounting for these differences by scaling the data relative to each season **massively** improved the RF and ADA models.

The baseline LR model performed very similarly on this scaled dataset as it did on the initial data, with a MSE of $2.28 * 10^{-3}$ and R2 of 0.263. The RF and ADA models had MSEs of $7.90 * 10^{-4}$ and $5.1 * 10^{-4}$, respectively, and R2 of 0.745 and 0.834, respectively. Obviously, as I explained earlier

[3] A measure of how much of the team's success can be attributed to an individual player [5].

about "narrative," I could not expect any statistical model to be able to account for all of the variance in the data. However, I was still shocked that any model could capture even 83% of the variance, *purely* from statistics alone.

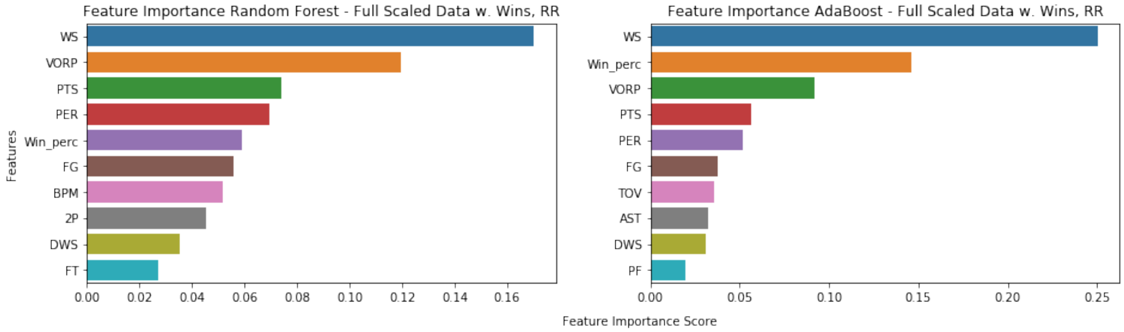


Figure 3: Top 10 Feature Weights on Scaled Dataset

Removing the redundant columns and scaling the data tremendously improved the performance of RF and ADA, by refocusing the feature importances of the models (Figure 3) to features that were more important to each season, and that were much less correlated with one another (for example, we see that Win Shares per 48 Minutes (WS/48) is no longer featured, as it was highly correlated with WS).

As can be seen in the comparison between Figures 2 and 3, adding the winning percentage was very influential to both models, along with Points (PTS), VORP, and WS, and scaling the data refocused RF on many other features. Additionally, something I noticed is that nearly every single one of the top 10 most important features, for both models, was related to the *offensive* performance of the players. The only *defensive* statistic that ranked within the top 10 most important was Defensive Win Shares (DWS), and that might be a bit of an anomaly, given the likelihood of its correlation with WS as a whole. Thus, a potential source of error (see section 7) is that the statistical models do not properly account for a player's defensive performance as a factor into MVP consideration.

6.5 Yearly Dataset - Final Attempt

After evaluating the models' respective performances on the scaled datasets, the only thing that was left to be done to improve the models was increasing the amount of data that it was trained on. Therefore, by holding out one season in a continual retraining/retesting loop, the models would receive an additional 6 seasons to be trained on that were previously used as a testing set holdout.

Because each of the years technically had its own models, I took the mean of each of the years' performance metrics to evaluate them. Firstly, the MSEs of both of the models (see Figure 4) were, overall, a bit worse than their respective counterparts from the scaled dataset (Section 6.4), but they each saw major improvement in recent years. The same held true for the R2 scores of the models.

However, as this dataset represented the "final" attempt to train the models, I wanted to see just how well it was doing at predicting the MVP, beyond just the MSE and R2 values. Therefore, I created additional metrics for evaluation:

1. MVP Accuracy – How well does the model predict the top finisher?

2. Top 2 Accuracy – Of the top 2 finishers, what percentage did the model predict correctly?

3. Top 5 Accuracy – Of the top 5 finishers, what percentage did the model predict correctly?

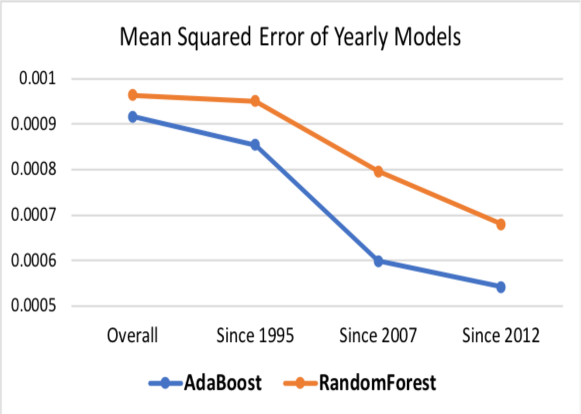


Figure 4: Mean Squared Error - Yearly Models

Further, I wanted to see whether the models' performance changed over time, as the "modern" NBA game may look differently than those of the 1980s when the three-point shot was newly introduced. To do that, I further broke down the average performances of each of the models into a time-based summary (see Figure 5).

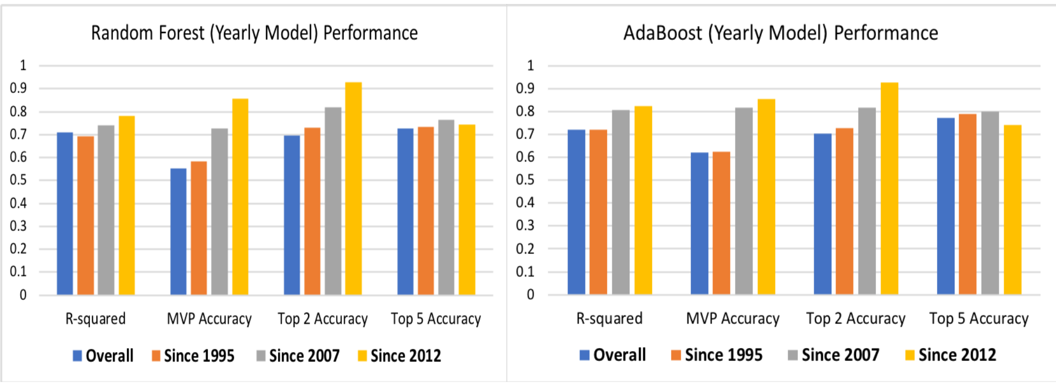


Figure 5: Performance of each of the yearly models

While the performance of RF and ADA in predicting MVP was fairly accurate overall (58% and 62%, respectively), their performance since 2007 is absolutely remarkable.

What we can see is that both of the models are increasingly well tuned for the modern era of the NBA for voting and prediction, particularly in predicting who will be the MVP and the runner-up. Both models have remarkable accuracy in predicting the MVP as of late, with RF predicting 73% of MVP winners and 82% of top 2 finishers correctly since 2007 and 86% of MVP winners and 93% of top 2 finishers correctly since 2012. The ADA model is slightly more accurate in predicting the first place finisher (82% since 2007 and 86% since 2012), as well as nearly identical ability in predicting the top 2 finishers (82% since 2007 and 93% since 2012). However, if one wants to figure out who would finish in the top 5, the ADA model consistently outperformed the RF model, no matter which year bucket we are examining.

In the trade-off for the increased level of accuracy in predicting the top 2 finishers, the models have faltered a bit in their prediction of who will round out the top 5 of MVP voting. However, this could also be explained by the propensity of very popular players getting a large number of votes simply due to their stature, and not necessarily due to their stats. On the flip side, there are some more unheralded players, such as Utah Jazz center Rudy Gobert, who, according to the model, ought to be given a higher consideration in the MVP for their performances. The models thus show us that in determining the MVP and the runner-up, there appears to be somewhat less credence given to the player's popularity over their statistical performance, whereas voters would be more likely to cast a vote for third through fifth place for a player who is more famous, despite a less-popular player having performed better over the course of the season.

6.6 Notable Outliers

One outlier in terms of R2 value is the 1999 season, in which the RF model had an R2 of 0.46 and the ADA model had an R2 of 0.43. In this year, the NBA had a lockout-shortened season, meaning that the teams only played 50 games each instead of the usual 82. Thus, the models were much less confident about the different thresholds by which to split the data and make a decision.

Additionally, in a very widely criticized decision, Steve Nash was awarded the MVP for the 2005 NBA season. As the leader of the revolutionary "7 seconds or less" Phoenix Suns offensive system, Nash was given this award largely based on narrative, despite leading the league in only a single major category, and placing out of the top 10 in all others. Additionally, in Win Shares, the most heavily weighted statistic in the random forest model, Nash finished in 15th, an incredibly low placement for an MVP. Thus, both of the models failed to predict the MVP for this season, and had R2 values of 0.06 and -0.01, respectively, for the RF and ADA models.

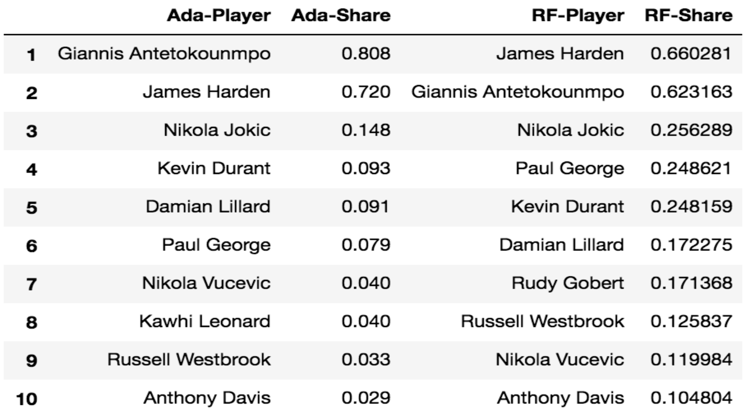
Lastly, both of the models predicted that James Harden should have won the hotly contested 2017 NBA MVP, with RF predicting that the eventual award winner Russell Westbrook would finish second, and ADA predicting that Kawhi Leonard would be the runner-up. Statistically, Russell

Westbrook finished 5th in WS, and his team, the Oklahoma City Thunder, finished 6th in the Western Conference in the regular season, compared to James Harden finishing 1st in WS and his Houston Rockets finishing in 3rd place, and Kawhi Leonard finishing 4th in WS, and his San Antonio Spurs finishing 2nd. However, these models do not capture "narrative" (see Section 7); in 2017, Westbrook finished the season averaging a "Triple-Double," meaning that he had double-digit average in PTS, Rebounds, and Assists – the first player to do so since Oscar Robertson had accomplished that feat in the 1962 Season.

Explaining the difference in predicting the 2nd place finisher in 2017 then lines up very will with the feature weights from Figure 3. RF's second-most important feature in evaluation is VORP, by a wide margin over the 3rd most important. In the 2017 season, Westbrook finished with a VORP that was 38% higher than the next highest player's, scored 8.5% more PTS than the next highest and a PER 10% higher. Thus, the RF model believed his performance in those categories was enough to lift Westbrook into second, as winning percentage is only the fifth most important factor. However, the ADA model's second-most important feature in evaluation is Win_perc, by a wide margin over the 3rd most important. Because Russell Westbrook's team did not have as good a winning percentage as Leonard, ADA placed Westbrook in 3rd in the MVP race.

6.7 2019 Prediction

After training each of the models on a scaled dataset with highly collinear columns removed from all of the 1980-2018 seasons, I sought to predict who would win the 2019 NBA MVP. The models actually ***disagree*** with one another on who should win (Figure 6).



	Ada-Player	Ada-Share	RF-Player	RF-Share
1	Giannis Antetokounmpo	0.808	James Harden	0.660281
2	James Harden	0.720	Giannis Antetokounmpo	0.623163
3	Nikola Jokic	0.148	Nikola Jokic	0.256289
4	Kevin Durant	0.093	Paul George	0.248621
5	Damian Lillard	0.091	Kevin Durant	0.248159
6	Paul George	0.079	Damian Lillard	0.172275
7	Nikola Vucevic	0.040	Rudy Gobert	0.171368
8	Kawhi Leonard	0.040	Russell Westbrook	0.125837
9	Russell Westbrook	0.033	Nikola Vucevic	0.119984
10	Anthony Davis	0.029	Anthony Davis	0.104804

Figure 6: Predictions for 2019 NBA MVP

These predictions of the ADA model line up pretty well with the current Vegas betting odds [6], as well as `basketball-reference.com`'s MVP tracker[4]. However, the RF model predicts that Harden's statistical superiority in WS, VORP, and PTS will catapult his MVP case over Antetokounmpo's.

7 Sources of Error

There can be multiple reasons why the model may have struggled with some of the predictions. As mentioned before, narrative/popularity of the player might be highly influential to the voters. To account for narrative in a model, we could potentially create an additional variable that measures either the amount of positive news coverage using sentiment analysis or the number of search engine results published during this season with "MVP" and a player's name. Lack of narrative in the model would also explain why there is a relatively low R-squared score, as much of the variance can be explained through the narrative/popularity angle.

Additionally, while there are some advanced statistics, such as DWS, that try to measure a player's defensive performance, it is extremely difficult to do so. No available defensive statistics can properly account for their impact on a game such as by preventing the other teams' star players from scoring when they are guarding those players. Thus, "two-way" players such as Kawhi Leonard and

[4] https://www.basketball-reference.com/friv/mvp.html

Paul George, who excel on defense as well as on offense, will typically have their MVP vote shares under-predicted by the models[5].

8 Discussion and Conclusion

In this paper, I set out to build a model that would be able to predict the vote shares for the NBA MVP award for each NBA player. I trained a RandomForest-based model as well as an AdaBoost model in my attempt to find the best predictor. Five different iterations of the underlying dataset were used to train the models, and I found that the best performing dataset was a version in which every season of the data was used, the teams winning percentage was factored in, certain columns with high collinearity were removed, and the data was scaled on a season-by-season basis. Both models achieved a surprising degree of accuracy, particularly, in their performance in the past decade. I found that while "narrative" may be influential to the voters at times, using statistics alone, not only can these models predict the vote shares, and thus winners, very well, but they can also capture a very large percentage of the variance. Additionally, these models give a statistical backing to the claims that Steve Nash should not have won MVP in 2005, and that James Harden should have been the winner of the 2017 MVP award. Lastly, based on the trained models, it is an extremely close race. However, because the ADA model has performed slightly better in the past, I predict that Giannis Antetokounmpo will be the 2019 NBA MVP.

References

[1] Scikit-learn documentation: Adaboost regressor. *SciKit-Learn*.

[2] Scikit-learn documentation: Section 3.2.4.3.2. – random forest regressor. *SciKit-Learn*.

[3] T. Boger. Using machine learning to predict the 2019 mvp: mid-season predictions. `https://dribbleanalytics.blog/2019/01/ml-mvp-predict-midseason/`, Jan 2019.

[4] O. Goldstein. Nba players stats since 1950. `https://www.kaggle.com/drgilermo/nba-players-stats`, Apr 2018.

[5] J. Kubatko. Nba win shares. `https://www.basketball-reference.com/about/ws.html`.

[6] J. Logan. Antetokounmpo and harden a two-man race in vegas' nba mvp betting odds, Apr 2019.

[7] S. O. Mitchell, A. Kromkowski, K. Saha, and Y. Song. Predicting nba's most valuable player. `https://github.com/csbond007/Linear_Models/blob/master/Project_Report.pdf`, Dec 2015.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, Nov. 2011.

[9] R. Pollack. Predicting the 2018 mvp winners with machine learning. `https://tht.fangraphs.com/mvp-2018-mookie-betts-christian-yelich-machine-learning/`, Oct 2018.

[10] S. Rajaram. Yet another basketball related project. here's how my work fared in the 2018 season. `http://sidrajaram.me/mvppredict.html`.

[5] Some might be quick to point out that Rody Gobert, typically thought of as a "defense-first" player, finished very high in RF's 2019 predictions. However, in the 2019 season, Rudy Gobert actually finished tied in second place with Giannis Antetokounmpo for total win shares, after registering the 2nd highest DWS and 4th highest OWS overall.