Abstract

For our project, we wanted to try to analyze the NBA MVP voting system using different methods used in class. In order to do this, we compared two methods- a Naive Bayes classifier which would classify players as an MVP or not and Newton's logistic regression to see which was a more accurate algorithm, which statistics end up being most important to voters when deciding the MVP, and finally, whether machine learning can even be used at all in such a subjective vote. We found that while the Naive Bayes classifier is accurate approximately ⅔ of the time, it did not accurately predict this year's MVP. For the logistic regression method, there was approximately ⅔ accuracy in classifying whether or not a player would be an MVP or not, but only 50% accuracy when predicting MVP per year (choosing one MVP out of three finalists). Both methods predicted that the same player would be voted as MVP for the 2020-2021 NBA season, but this predicted MVP was not the one actually chosen two days ago. This could reveal the existence of voter bias that motivates voters to pick an MVP not based on statistics, but other factor categories like popularity level, thus calling to question the subjectivity of MVP voting.

Introduction

Both of us are big NBA fans, and more generally, sports fanatics and were seeking a way to apply the machine learning skills learned in CS229 to analysis of the league. Currently, the league MVP is determined by a vote of journalists, which would suggest that there's a large amount of bias from year to year in how the MVP is decided. However, we wanted to determine if we could create an algorithm that would be trained on past MVP votes and then use that to predict the 2021 MVP. This is important because it will determine how objective voters actually are when deciding how to cast their MVP vote as well as possibly creating an algorithm for how MVP selection should be made. After the algorithms train on many years of MVPs, it can determine what have been the most essential statistics and weigh those higher when picking the MVP, unlike when human voters do the selecting and simply vote based off bias, echo chambers, and personal perception which isn't always fair. We used both a Naive Bayes classifier and logistic regression for our analysis because we wanted to see which algorithm could best align itself to how human voters decided on MVP. This choice of algorithms was also made so that we could investigate whether a generative or a discriminative methods model would be more optimal for predicting the winner.

For the Naive Bayes classifier, we produced two different results- first, the algorithm predicted purely which players it thinks should be MVPs regardless of whether this means picking multiple MVPs in one year or no MVPs in another year. Second, the algorithm picked one MVP per year. We input ten stats of the three players chosen as MVP finalists for each year: games played, average minutes per game, field goal percentage, 3 point percentage, free throw percentage, and rebounds, assists, steals, blocks, and points per game. We then received a few outputs- the predicted winner of the 2021 MVP using both types of predictions mentioned, which five statistics were most implicative of an MVP, and how accurate the algorithm was on the test set. Our training set was the data from the 2000-01 season to the 2015-16 season and our test set was the data from the 2016-17 season to the 2019-20 season.

A similar approach was utilized for the logistic regression procedure. When the logreg algorithm was designed to pick one MVP for a given year, the statistics of the three finalists for that

year were passed in as input and outputted were the probabilities that each finalist would be the MVP. So, the finalist with the highest probability would be the model's choice for MVP. This procedure was conducted across multiple years worth of MVP finalists' statistical data found in the validation data set, and the measured accuracy was found to be approximately 50%. When the algorithm was commanded to purely predict MVPs and was not bounded by year, players were accurately classified as MVP or not at a rate of 67%.

Related work

Utilizing machine learning for sports is not a new concept. A popular data science website Towards Data Science actually did a similar analysis earlier this year where they attempted to use several models to predict the MVP for this upcoming season.[1] They used all the stats we used in addition to a few more team based stats like team wins and overall playoff seed as well as more advanced stats like win shares. Our access to this data wasn't quite as easy and would've added several hours to our overall process so we left these stats out. They then compared a deep neural network, k nearest neighbor regression, and a random forest regression to see which method worked the best. This seemed to be a strong approach because of the utilization of multiple types of models as well as the wide variety of stats used.

Another article which attempts to predict the MVP was one also written by fans who tried to just use a decision tree model.[2] They asked a series of questions such as "Did the player's team win more than 75% of their games" as opposed to pure stats to try to come to their conclusion. This actually ended up correctly predicting the MVP for this year, but the fact that they just used one model and it didn't seem to use a strict machine learning algorithm made it less scientifically sound to us. One aspect they did include that could've been useful was instead of classifying players as MVP winner or not, they ranked them based on how they finished in the MVP results, ie first, second, third, etc. Using this method wouldn't have been feasible in our Naive Bayes or logistic regression classifiers, which just take 1s and 0s in the labels vector, but it may have increased the accuracy. This overall prior research guided our choice of algorithms, as we decided on ones that had been neglected in the available literature in an attempt to find even better predictive models for NBA MVP winners.

Dataset and Features

Our training dataset consisted of 16 years worth of MVP finalists statistics in what ended up being a 48x10 matrix, with each player on each of the rows and each statistic on each of the columns. In order to avoid processing, we deleted all labels from the matrix because the players represented in the training matrix don't matter and the order of our statistics was stored elsewhere: games, minutes, fg%, 3P%, ft%, rebounds per game, assists per game, steals per game, blocks per game, and points per game. Our validation dataset was used to determine how accurate our algorithm was at predicting the MVP for a given season as well as when compared to other MVP candidates across multiple

---

[1] https://towardsdatascience.com/predicting-2020-21-nbas-most-valuable-player-using-machine-learning-24aaa869a740

[2] https://www.perthirtysix.com/essay/2021-nba-mvp-race

seasons. This ended up being a 12x10 matrix. We then had separate vectors which included the players' labels: a 1 if the player won an MVP in a specific year and a 0 if they didn't. This vector represented players in the same order they were represented in the original matrices, we just wanted to simplify the parsing process within Python and separate this vector from the rest of the matrix ahead of time. Finally, we had a 3x10 matrix for our prediction matrix for this past season. Altogether, a snippet of some of our input data looked like this:

71	42	0.42	0.32	0.814	3.8	4.6	2.5	0.3	31.1
74	39.5	0.572	0	0.513	12.7	3.7	0.6	2.8	28.7
82	38.7	0.499	0.259	0.618	12.2	3	0.9	2.3	22.2
67	36.1	0.579	0	0.555	10.7	3	0.6	2	27.2
82	40.6	0.508	0.1	0.799	12.7	3.7	0.7	2.5	25.5
82	37.3	0.391	0.321	0.814	7.3	9.9	2.1	0.2	14.7
82	41.5	0.451	0.383	0.843	6.9	5.9	2.2	0.8	30
81	39.3	0.513	0.273	0.71	12.9	3.9	0.7	2.9	23.3
82	40.5	0.502	0.282	0.751	13.4	6	1.4	1.6	23
82	39.4	0.499	0.256	0.791	13.9	5	1.5	2.2	24.2
69	36.6	0.501	0.167	0.599	12.4	3.1	0.9	2.7	22.3
78	35.7	0.434	0.111	0.757	10	2.1	0.8	2.6	20.1
78	38.7	0.459	0.399	0.869	9.7	3.1	1.2	1.5	26.1

Thus, our training data ended up making up ¾ of our data while our validation data made up ¼ of the data. Initially, our datasets were much bigger- instead of just using stats from the MVP candidates, we utilized stats from all players in the league every single year which ended up giving us a matrix of size 3184x10, with the vast majority of those labels being equal to 0. This meant that when the algorithms ran, they were getting super high accuracy values but predicting no one to win MVP since the MVP winners were such a low percentage of the overall dataset. Thus, we had to switch to just the best three players in the league each season, since this would both allow the algorithm to pick a winner amongst only the best players in the league as well as force it to give approximately ⅓ of each dataset the MVP label (since there are 3 finalists each year). In doing this data preprocessing, we avoided binary classification with too strongly imbalanced classes. We obtained all our data from the very convenient site Basketball Reference, which allowed us to handpick the stats we wanted in our dataset.[3]

Note: An important assumption that our algorithms make is the strong independence condition among features: a player's number of blocks per game is unrelated to a player's number of rebounds per game.

Methods

We wanted to use multiple algorithms for our project because we wanted to determine whether one could better predict an MVP.

For Naive Bayes, we trained the data in a very similar way to how it was trained in the second problem set where we were trying to classify whether emails were spam or not. First, for every player in the matrix, if the player was an MVP, one would be added to the MVP counter. Then, for each stat, there was a dictionary entry made with the average value for that stat. For example, if the average of three MVPs was 35.6 points per game and the average of nine non MVPs was 32.2,

[3] https://www.basketball-reference.com/leagues/NBA_2021_per_game.html

these were added as the keys to the dictionary. Then, the total likelihood that a player in a given set was or wasn't an MVP was calculated. The total likelihood list was returned and used on the validation data. For each player in the validation dataset, the log of the training model at a stat multiplied by the player's stat for each stat was taken. These were then added to the probability that a player either was or wasn't the MVP. For example, if the model said that field goal percentage was .487% correlated with an MVP and .423% correlated with a non MVP, each of these were multiplied by the player's field goal percentage and then the log of this was taken. All of the stats were taken like this and then added together and then this was added to the probability of a player being an MVP or not being an MVP. Finally the log of this was taken, so that each player ended up with a probability of either being an MVP or not being an MVP. However, we quickly realized this wasn't enough because sometimes, the algorithm was predicting multiple MVPs for one season or no MVPs for other seasons. Thus, we calculated the "pure" MVPs as well as the "seasonal" MVPs by parsing our dataset into threes and forcing it to assign 1 as the label to the highest difference between MVP likelihood and non-MVP likelihood of each of three players coupled together, or 0 otherwise.

Under non-normality of features (namely, basketball statistics in this case), we chose the logistic regression model over discriminant analysis. Our decision to use logreg also relied on the fact that feature space is split linearly with this model, allowing for our assumption of independence to be cushioned if some features turn out to be correlated in reality. For our logistic regression methodology, the training input file had n=48 examples, one player's statistics x^(i) per row. In particular, the i-th row contained $x_d \in R$ with d = 10. Since this is a binary classification, the training label file contained a matrix of 48x1, where the outcome in each row could take on two values {0, 1} to indicate if the player was chosen as MVP. After training this logreg model, we were then able to predict probability scores for our validation input. To predict the "pure MVPs" (as described in the Naive Bayes methodology above), the algorithm was designed to select the players whose probability scores were highest: (1) divide total number of players in the validation set by the number of finalists per year to calculate how many MVPs the model should pick (let this value equal w), then (2) select the top w amount of players with the highest probability scores. To predict the "seasonal" MVPS, the validation input was split by year, so that the algorithm had to choose one MVP out of three finalists for each year by figuring out which of the three players had the highest probability score.

Results and Discussion

Our Naive Bayes model was accurate 66% of the time both when using a pure model as well as a seasonal model, meaning when it was forced to pick an MVP each season, it correctly selected the MVP twice and incorrectly selected it twice. When it didn't have to pick one each season, there was one season where it didn't pick any MVP and one season, 2015-16 where it picked two MVPs. Contextually, this was fitting. The 2015-16 MVP race ended up being one of the closest races of all time, so it is encouraging that although the model picked two winners from the same year, there nearly were two winners that year anyways. Unfortunately, the model incorrectly selected Joel Embiid as the MVP winner for this year using both the pure MVP model and the seasonal model, when Nikola Jockic actually won. Embiid placed second in the voting. When looking at this, Jokic played 20 more games than Embiid, played slightly more minutes per game, had slightly higher shooting percentages across the board, and had a lot more assists. Embiid averaged more points and blocks per game. So why did our model predict him to win it over Jockic?

```
Naive Bayes had an accuracy of 0.6666666666666666 on the testing set
The top 5 indicative words for Naive Bayes are:  ['3P%', 'assists', 'ft%', 'fg%', 'steals']
```

```
The MVP for this year will be  Joel Embiid
```

Rk	Player	Pos	Age	Tm	G	MP	FG%	3P%	FT%	TRB	AST	STL	BLK	PTS ▼
1	Stephen Curry	PG	32	GSW	63	34.2	.482	.421	.916	5.5	5.8	1.2	0.1	32.0
4	Joel Embiid	C	26	PHI	51	31.1	.513	.377	.859	10.6	2.8	1.0	1.4	28.5
10	Nikola Jokić	C	25	DEN	72	34.6	.566	.388	.868	10.8	8.3	1.3	0.7	26.4

Our model stated that 3P%, assists per game, ft%, fg%, and steals were the top five indicators of an MVP, making this even more of a mystery. Part of this could be due to factors we didn't include in our model such as playoff seeding and added wins. When doing a deeper analysis, it appeared that since Jockic had across the board stats, this contributed to a higher likelihood of being both a MVP candidate and a non MVP candidate. Additionally, our model may have been hurt by the binary classification of just 1 for being an MVP or 0 for not.

Our logistic regression model, on the other hand, resulted in a 67% accuracy for predicting "pure" MVPs versus a 50% accuracy for predicting seasonal MVPs. When our logreg model was applied to predict this year's MVP from finalists Stephen Curry, Joel Embiid, and Nikola Jokic, the resulting MVP was the same as the one predicted by Naive Bayes: Joel Embiid. Both models predicted the same MVP for 2021, but this result was not the player who was officially chosen as 2021 MVP as of two days ago: Nikola Jokic.

```
Probability scores for Steph Curry, Joel Embiid, Nikola Jokic: [0.9828921  0.99542214 0.99219258]
```

However, Embiid was only chosen by our logreg model over Embiid because of a difference of 0.003 in their probability scores, which indicates a small margin of error.

Conclusion and Future Work

Our results indicate that Naive Bayes was more accurate in predicting "seasonal" MVPs over logistic regression, and this disparity may be attributed to the difference in learning mechanisms when accounting for the limit of the training data size. With limited data, the generative model of Naive Bayes has more "information" about the data than the discriminative model of logistic regression, so Naive Bayes is better suited for predicting the MVP overall.

Our logistic regression model has room for improvement. A larger sample of MVP finalists may benefit its accuracy, but this data is not readily accessible for older NBA seasons. Also, the high dimensionality of our input file could benefit from dimensionality reduction in the form of regularization such as lasso or ridge reduction.

Another mitigating factor in the accuracy of our models is the assumption of feature independence. Further research can be conducted to better understand the correlation between steals, points, rebounds per game, etc.

To continue this project in the future, we might opt away from using one-hot vectors as our labels and instead use the number of votes that each finalist receives. This would change our output to be a continuous variable and allow us to experiment with other machine learning models in our prediction of MVP. However, this dataset is not readily available for use and would require further research and communications with official basketball statistic keepers.

Additionally, if votes are casted with descriptive voter feedback, it might be useful to implement sentiment binary classification on the voter's language to find patterns in voter thought process. This additional input can be factored in to account for voter subjectivity when deciding who should be classified as winners and losers.

References

Page, BBR et al. "2020-21 NBA Player Stats: Per Game | Basketball-Reference.Com". Basketball-Reference.Com, 2021, https://www.basketball-reference.com/leagues/NBA_2021_per_game.html. Accessed 11 June 2021.

"Predicting 2020–21 NBA'S Most Valuable Player Using Machine Learning". Medium, 2021, https://towardsdatascience.com/predicting-2020-21-nbas-most-valuable-player-using-machine-learning-24aaa869a740. Accessed 11 June 2021.

"The 2021 NBA MVP Race". Perthirtysix.Com, 2021, https://www.perthirtysix.com/essay/2021-nba-mvp-race. Accessed 11 June 2021.

Contributions

Sally was the lead on the Naive Bayes algorithm while Huong was the lead on the logistic regression. Both contributed equally to this report.