Abstract

The National Basketball Association(NBA) is the premier professional basketball league in the world. In the NBA discussion sphere, the topic of who will win the MVP is a hotly contested debate. Attempts have been made to quantify, both qualitatively and quantitatively, a player's impact on the court as well as predict who will win the MVP. We have found that many discussions surrounding this topic have been heavily biased one way or the other, so we decided to let the stats tell the story. Therefore, for our analysis, we built a regression model in Spark Scala that aggregated per game advanced and basic statistics from NBA.com and BasketbalReference.com to predict the NBA MVP based on numbers alone. We found that our model performed exceptionally in predicting a player's impact on the court (measured by Win Shares), as well as correctly predicting the MVP award for this past NBA season (2023-2024).

Introduction

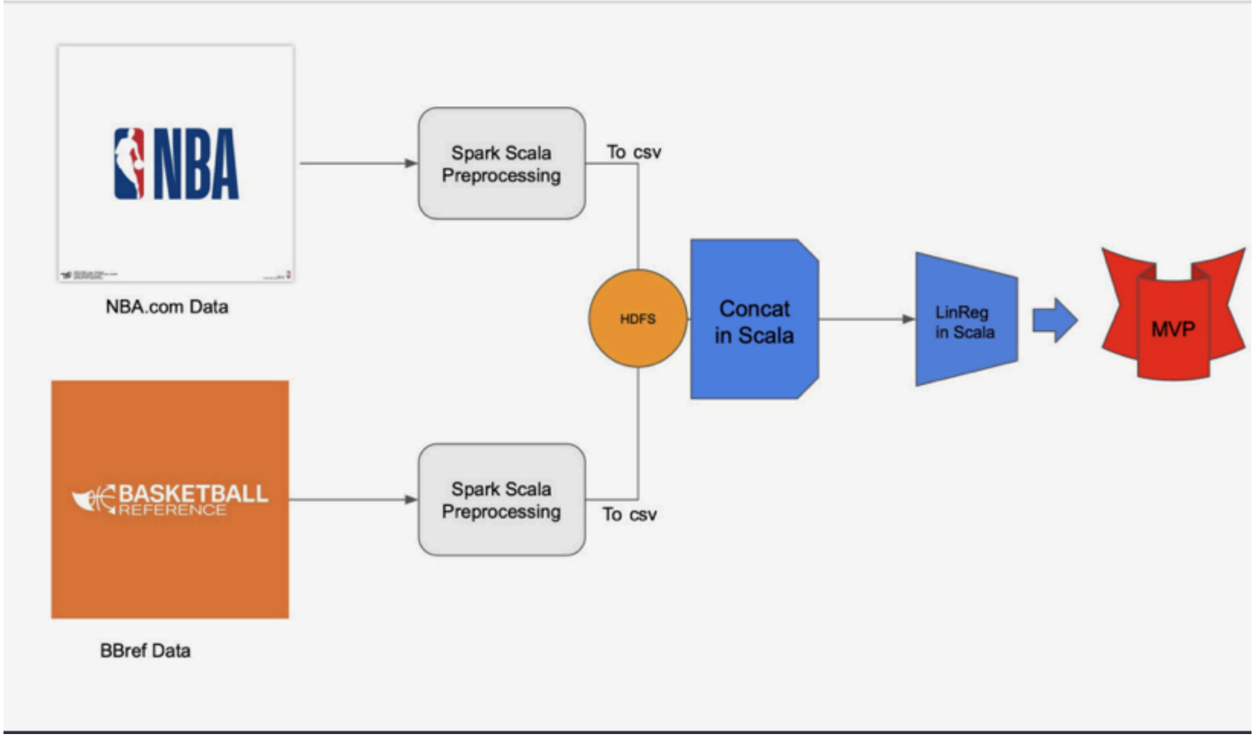
Ever since the first NBA Most Valuable Player award was introduced in 1956, players, analysts, and fans alike have taken a stab at predicting who will be the NBA MVP. Beyond providing entertaining discussions amongst basketball fans, the result of the MVP award bears a significant weight on the profits of players, teams, and fans.

In recent years, the NBA has undergone an unprecedented 'data revolution' that has seen stats recorded for nearly everything you can think of, as well as advanced statistics aiming to capture board trends across your regular counting stats. Through our research however, most of the discussions surrounding the MVP award lack an analytics first approach, and don't leverage the innovations and power of machine learning. Even statistics created that aim to capture the essence of a player's impact like FiveThirtyEight's RAPTOR (Silver, 2019) are not necessarily aiming to produce a prediction on who should win the MVP award. Papers published relating these advanced statistics to the actual outcome of the MVP award could not be found, with the closest work being a study on the Player Efficiency Rating (PER) statistic's ability to predict wins. (Van Curen, 2012)

Given the state of the discussion surrounding the MVP, as well as the wealth of data available to the public, we decided to build a regression model to predict the NBA MVP. Our predictive analytic aims to aid all stakeholders in the MVP race and provide them with a tool that could aid them.

The general design of our predictive analytic is shown in Figure 1. Advanced statistics from NBA.com are inputted as a CSV into the Hadoop Distributed File System (HDFS) hosted on NYU's dataproc instance. This data is cleaned and processed before being saved once again as a CSV in HDFS. Similarly, the Advanced statistics, as well as traditional per game statistics from Basketball Reference underwent similar procedures and were saved as CSVs in HDFS. The two files were then joined on player name and year in spark scala before being transformed and inputted into our regression model. The final output is a table where the player whose score is the highest is our MVP prediction.

Figure 1



Motivation

The motivation of why we are trying to predict the MVP is because of the massive effect it has on all of the stakeholders involved in the NBA. If a player wins the MVP award, per the NBA collective bargaining agreement, they are entitled to a significantly higher proportion of their team's salary cap, a pay raise to the scale of tens of millions of dollars. Aside from its effect on a team owner's payroll, having a player who wins the MVP award would drive fans to fill a team's stadiums and purchase merchandise and jerseys belonging to that player and his team. More recently, with the legalization of sports betting in the United States of America, companies like Fanduel and DraftKings have set up odds for NBA MVP awards that fans can wager on anytime during the season. These bets make up millions of dollars of profit for both the fans as well as the oddsmakers. Personally, we are both massive fans of the NBA, and find the project challenging yet rewarding.

Related Work

Silver. [Silver (2019)] This page is the description and methodology behind the Robust Algorithm (using) Player Tracking (and) On/Off Ratings (RAPTOR) statistic of the website FiveThirtyEight. The statistic aims to capture how the NBA values a particular player, however there is no predictive component of the statistic in relation to the MVP race, which we aim to produce

Van Curen. [Van Curen (2012)] This paper is an analysis on how the statistic Player Efficiency Rating affects a team's wins. Although this paper does perform some analysis on linking an advanced statistic to both team and player success, it lacks both a comprehensive dataset of teams and other possible statistics to use as well as a link to the MVP race, which we sought to improve

Description of Datasets

I. Basketball Reference Dataset

Basketball Reference is one of the premier sites online for tracking player data. This includes each player's season advanced statistics as well as each season's basic per game statistics for each player. We took the Advanced Statistics table, and the Per Game Statistics table from the last 5 seasons, including the current season (2019-2023). Each table has at least one row for each player who played at least one game in a given season. Figure 2 is a table that includes more detailed information on the specific schema of the tables in each dataset.

Figure 2

Advanced

NAME	DATA TYPE	DESC
Rk	Int	Index of player
Player	String	Name of Player
Pos	String	Position of Player
Age	Int	Age of Player
Tm	String	Player's Team
G	Int	Games Played In Season
MP	Int	Minutes Played in Season
PER	Float	Player Efficiency Rating (Catch-all metric)
TS%	Float	True Shooting Percentage (Adjusted Shooting Percentage)
3PAr	Float	Three Point Attempt Rate %FG from 3pt
FTr	Float	FT attempts per shot
ORB%	Float	% of available offensive rebounds a player grabbed
DRB%	Float	% of available defensive rebounds a player grabbed
TRB%	Float	% of available total rebounds a player grabbed
AST%	Float	An estimate of the percentage of teammate field goals a player assisted while they were on the floor.

STL%	Float	An estimate of the percentage of opponent possessions that end with a steal by the player while they were on the floor.
BLK%	Float	An estimate of the percentage of opponent two-point field goal attempts blocked by the player while they were on the floor.
TOV%	Float	An estimate of turnovers committed per 100 plays.
USG%	Float	An estimate of the percentage of team plays used by a player while they were on the floor.
OWS	Float	An estimate of the number of wins contributed by a player due to offense.
DWS	Float	An estimate of the number of wins contributed by a player due to defense.
WS	Float	An estimate of the number of wins contributed by a player.
WS/48	Float	An estimate of the number of wins contributed by a player per 48 minutes (league average is approximately .100)
OBPM	Float	A box score estimate of the offensive points per 100 possessions a player contributed above a league-average player, translated to an average team.
DBPM	Float	A box score estimate of the defensive points per 100 possessions a player contributed above a league-average player, translated to an average team.
BPM	Float	A box score estimate of the points per 100 possessions a player contributed above a league-average player, translated to an average team.
VORP	Float	A box score estimate of the points per 100 TEAM possessions that a player contributed above a replacement-level (-2.0) player, translated to an average team and prorated to an 82-game season.

Per Game

NAME	**DATA TYPE**	**DESC**
Rk	Int	Rank
Player	String	Player Name
Pos	String	Position

Age	Int	Age
Tm	String	Team
G	Int	Games Played
GS	Int	Games Started
MP	Double	Minutes Played/Game
FG	Double	Field Goals/Game
FGA	Double	Field Goal Attempts per Game
FG%	Double	Field Goal Percentage
3P	Double	Three-Point Field Goals per Game
3PA	Double	Three-Point Field Goal Attempts per Game
3P%	Double	Three-Point Field Goal Percentage
2P	Double	Two-Point Field Goals per Game
2PA	Double	Two-Point Field Goal Attempts per Game
eFG%	Double	Effective Field Goal Percentage
FT	Double	Free Throws per Game
FTA	Double	Free Throw Attempts per Game
FT%	Double	Free Throw Percentage
ORB	Double	Offensive Rebounds per Game
DRB	Double	Defensive Rebounds per Game
TRB	Double	Total Rebounds per Game
AST	Double	Assists per Game
STL	Double	Steals per Game
BLK	Double	Blocks per Game
TOV	Double	Turnovers per Game

PF	Double	Personal Fouls per Game
PTS	Double	Points per Game

II. NBA.com Dataset

NBA.com is the official website of the NBA and keeps track of its own advanced statistics that differ from those available on Basketball Reference. There is one CSV file, or datatable, for each season and we took a table from the last 5 seasons, including the current season (2019-2023). Each table has at least one row for each player who played at least one game in a given season. Figure 3 is a table that includes more detailed information on the specific schema of the tables in each dataset.

Figure 3

NAME	**DATA TYPE**	**DESC**
INDEX	Int	Index of player
PLAYER	String	Name of player
TEAM	String	Name of player’s team
AGE	Int	Age of player
GP	Int	Games played by player
W	Int	Number of games won
L	Int	Number of games lost
MIN	Float	Minutes played by player
OFFRTG	Float	Offense rating - an estimate of points produced (or scored) by a player per 100 possessions
DEFRTG	Float	Defensive rating - an estimate of points allowed by a player per 100 defensive possessions
NETRTG	Float	Net rating - the difference between a player's or team's offensive rating and defensive rating, indicating overall impact
AST%	Float	Assist percentage - the percentage of teammate field goals a player assisted while he was on the floor
AST/TO	Float	Assist to turnover ratio - the number of assists a player has per turnover
AST RATIO	Float	Assist ratio - the number of assists a

		player averages per 100 possessions used
OREB%	Float	Offensive rebound percentage - the percentage of available offensive rebounds a player grabbed while he was on the floor.
DREB%	Float	Defensive rebound percentage - the percentage of available defensive rebounds a player grabbed while he was on the floor
REB%	Float	Rebound percentage - the percentage of total available rebounds a player grabbed while he was on the floor
TO RATIO	Float	Turnover Ratio - the number of turnovers a player commits per 100 possessions used
EFG%	Float	Efficient field goal percentage - this stat adjusts for the fact that a 3-point field goal is worth more than a 2-point field goal
TS%	Float	True shooting percentage (Adjusted Shooting Percentage) - takes into account field goals, 3-point field goals, and free throws
USG%	Float	Usage percentage - an estimate of the percentage of team plays used by a player while he was on the floor
PACE	Float	Pace of play - the number of possessions a player is involved in
PIE	Float	Player impact estimate - a measure of a player's statistical contribution against the total statistics in games they play in
POSS	Int	Number of possessions played

Analytic Stages, process

I. Ingestion

A. Basketball Reference Data (Advanced)

For each season, Basketball Reference has a separate page that contains the table for all players' advanced statistics data titled "2023-24 NBA Player Stats: Advanced", where 2023-24 represents the 2023 season. For each season we selected (2019-2023), we selected "Get table as CSV (for Excel)" and copy-pasted the raw text data from the webpage into a separate CSV file and we uploaded each to HDFS and loaded it as a dataframe in spark scala.

Basketball Reference Data (Per Game)

Follow the exact same steps as the above section, except with the “2023-24 NBA Player Stats: Per Game”, where 2023-24 represents the 2023 season, for each season we selected (2019-2023).

B. NBA.com Data

For each season, NBA.com has a separate page for advanced statistics. For each season, we copy and pasted the table from the website onto an Excel spreadsheet, and then saved that season’s data as its own CSV. Each CSV was uploaded to HDFS and loaded as a dataframe in spark scala.

II. Cleaning and profiling

A. Basketball Reference Data Cleaning

We wrote a function to process each year’s data from basketball reference so that we could clean the data for each season while reusing code, as well as a User Defined Function (UDF) to clean the names of each player for proper joining. After loading the advanced data for the selected season, we began by stripping the accents off of each player’s name with our UDF as the NBA.com data does not include accents. We then cut trailing and leading spaces for all columns as well as dropping duplicates for every player as we only wanted one record per player. Players with multiple records were players traded mid season. Since we wanted to get the picture of a player’s entire season we dropped all rows that weren’t the first occurrence of their name, which left us with just the row that incorporated their entire season. We then converted all columns that were incorrectly classified as string types to doubles.

We then loaded the per game data and selected the columns that we wanted to include, PTS, BLK, STL, AST, and TRB, before joining this table with our advanced statistics table by player after setting all data in the remaining string columns as lower case. We also added the last two digits of the season to the player name in order to accurately match each player and season with the proper player and season in the NBA.com dataset.

We ran this function over the first 4 seasons as a training set, and the last season separately as a testing set before saving the training set and testing set as their own CSV.

B. Basketball Reference Data Profiling

We wrote a function to profile each season’s data so we could reuse the code for each season. We began by repeating the data cleaning steps above without saving the data frame. We then calculated the mean, median, and standard deviation for all numerical columns in the dataset and displayed the output in the spark scala environment. We repeated this process for all 5 seasons.

C. NBA.com Data

We began by loading each CSV into a dataframe and for each player name added a suffix of the last two digits of the year that the CSV represents for proper matching. We then combined the first 4 seasons into a single dataframe that we would train our model on (2019-2022). The current season (2023) would be our testing data set. For both the training and testing data, we began by converting all numerical columns to floats for processing, and then converted the data in the “PLAYER” and “TEAM” columns to all lowercase letters for proper matching. We then created a “WIN%” column for each player based on the games they won and lost. We finally saved the combined training data as its own CSV and the testing data as its own CSV.

D. NBA.com Data Profiling

For each numerical column in the training data dataframe, we calculated the median, mean, mode, max count, and standard deviation and outputted that data into the spark scala display. We calculated and displayed the same stats for the testing set.

E. Merging/Cleaning Data for Regression

To merge the two datasets, we loaded the training and testing data for Basketball Reference and NBA.com into dataframes from the saved CSVs above, resulting in 4 dataframes. For the Basketball Reference data, we renamed the “Player” column “PLAYER” for easier merging. We also renamed the columns “Age”, “TS%”,”USG%”, and “AST%” all to “DROP” since they were duplicates in the NBA.com data. We then joined the NBA.com training data on the “PLAYER” column. We then dropped the columns "W","L","WINNER","Tm","G","MP","DROP", and "is_frontcourt" since we determined that they were not necessary for our analysis in the case of “is_frontcourt” or were duplicates. We then filtered the training data to only include players that played more than 10 games in order to only include analysis on players that had impact, and then converted all numerical columns to doubles.

We then selected “WS” as our target variable and renamed it “label” for our regression. We chose this metric since we concluded that it captured what it meant to be the MVP, as Win Shares are an estimate of the overall contribution a player has had in the wins of their team. We also dropped the columns “OWS”, “DWS”, and “WS/48” as they are components of Win Shares, or are calculated from Win Shares. This prevents both overfitting to our dataset as well as data leakage during the training of our model. We then used the necessary functions to create a feature vector for each player, and saved that into a table that included the player’s name, position, team, and label so that we can keep track of our model’s predictions.

We performed the exact same merging and cleaning techniques on the testing sets of the NBA.com data and Basketball Reference data except for filtering off 10 games since we wanted to include all data.

At the end of this step, we had a cleaned and prepped training data table and a separate cleaned and prepped testing table.

III. Regression and Final output

We began by initializing our basic linear regression and trained it on our training dataset from before. We then inputted our testing data for predictions and measured both the RMSE as well as r^2 of our model in order to assess its performance. The basic linear regression resulted in an RMSE of 0.561 and an r^2 of 0.957, indicating excellent performance in predicting WS. We also displayed the coefficient table to help aid in the interpretability of our model. For our final output table, we ordered the predictions by predicted WS as well as limiting the table to players who have only played 65 games due to the MVP being limited to players who have only played 65 games.

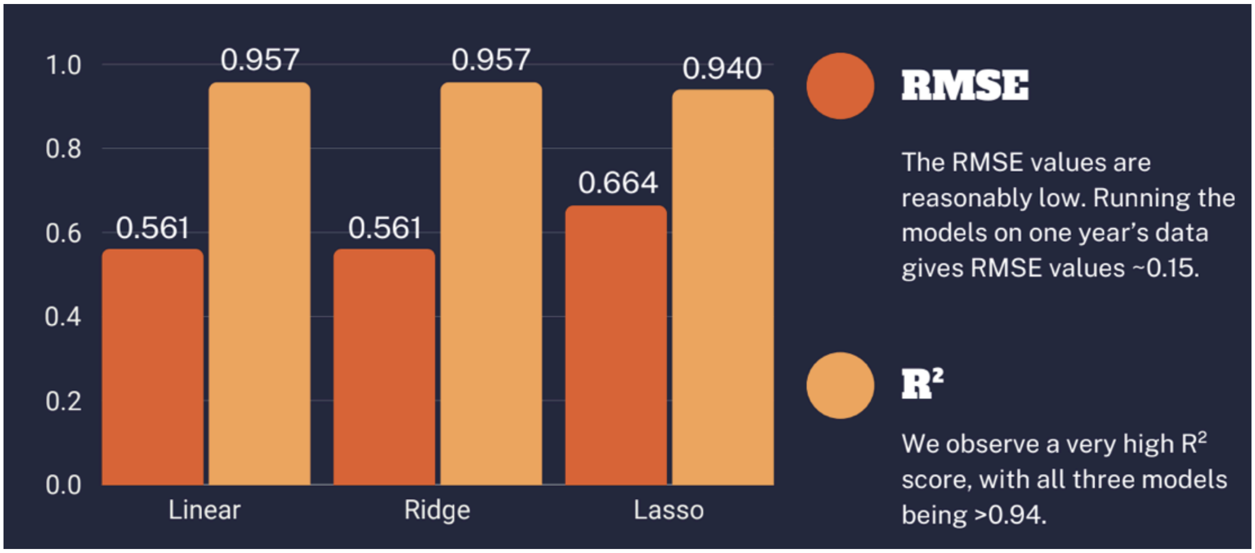
Initially, we were worried that the model was overfitting to the data that we presented, and as such we decided to run two separate models that incorporated L2 regularization (Ridge Regression), and L1 regularization (Lasso Regression). We performed the same steps and gathered the same output metrics as the basic linear regression above. Ridge regression resulted in an RMSE of 0.561 and an r^2 of 0.957, whereas lasso regression resulted in an RMSE of 0.664 and an r^2 of 0.940, both indicating good model performance.

For all 3 models, the prediction for 2023 NBA MVP was Nikola Jokic, who was just announced as the actual NBA MVP weeks after the conclusion of our experiment.

Graphs(s) - a visual representation of analytics

Figure 4 is a visual representation of the r^2 and RMSE of each model

Figure 4



Conclusion

The performance of our overall analytic, as well as its accurate prediction of this year's MVP speaks volumes to the power of big data, as well as the power of the simple linear regression that is often forgotten in favor of more complicated models. We successfully predicted the NBA MVP with great model performance on unseen data. The results of this model could be integrated into decision making for teams across the NBA, and could play a factor in projecting value in player trades, free agency signings, and gametime decisions. Further expansion on this project could include backtesting on more historical data, applying similar methods to college basketball, or using different target variables.

Works Cited

1. Silver, Nate, et al. “How Our RAPTOR Metric Works.” *FiveThirtyEight*, ABC News Internet Ventures, 24 Oct. 2019, fivethirtyeight.com/features/how-our-raptor-metric-works/.
2. Van Curen, Anthony. "The Effect Alternate Player Efficiency Rating Has on NBA Franchises Regarding Winning and Individual Value to an Organization." *Sport Management Undergraduate*, Spring 2012, St. John Fisher University, Fisher Digital Publications, fisherpub.sjf.edu/cgi/viewcontent.cgi?article=1037&context=sport_undergrad. Accessed 12 May 2024.