Predicting the 2020-21 NBA MVP: A Statistical Prediction

Forrest Martin, Ben Kronman, Abdel Diab, and Hyman Abadi

Table of Contents

Introduction

Basketball is one of the most popular sports in the US, and in the world. Growing up, I would watch NCAA and NBA games all the time, and then go try to recreate what I'd just seen in my driveway the next day. As I got older, though, I began to become more interested in the "Why?" of basketball and not just the "What?" or "How?" which is what many people may ask as they see LeBron

James, Michael Jordan, or Kobe Bryant jumping some ridiculous distance and dunking over two defenders. I became interested in "Why" LeBron James always seemed to be struggling to beat the Warriors in the Finals, or "Why" James Harden seemed so inefficient yet incredibly deadly and effective at the same time, or "Why" Giannis Antetokounmpo won the MVP award over LeBron James, James Harden, or Luka Doncic. So when this project was announced, I knew that the data I wanted to analyze was NBA data, to understand why things happened the way they do, and the members of our group were similarly inclined.

Since 1955, the best performing player in the National Basketball Association has been awarded the Most Valuable Player (MVP) title. Over the years, the title has gained a lot of attention and debate among basketball fans and media professionals alike. The race for the title is one of the most intense topics during the NBA season and one that sparks heated debates about who is most deserving of the title.

The narrative of every season plays a big part in the ultimate decision of who will win the title. While the MVP award is awarded to the player who has the largest impact on their team, quantifying that impact is near impossible leading to a lot of subjectivity inevitably taking place in electing the most qualified candidates and choosing the most qualified player out of those players. Typically, the debate for who should win MVP goes on all season, with a few players being championed as favorites by different members of the media. It is contentious and almost always ends with disagreement and disappointment for those fans who believed their player should have won.

Because of all of this, and how much people seem to love debating who should win or who should have won in the past, we thought it would be a good idea to try to quantitatively judge who should be MVP, based on the statistical data, rather than rely on the media narrative as so many fans do. Our objective is to create a predictive model using player stats available to the public that will rank the best players according to who should win the next potential MVP award. We hope that the model we create will have higher than a 50/50 chance of predicting who the next MVP award goes too. Relying on historical data, we will use the model to run the player statistics for the past five years and see how many times our model accurately predicts the player awarded with MVP.

Betting on sports is considered a very popular form of gambling. For a lot of people, betting is simply putting your weight behind your favorite team. It doesn’t matter if your team hasn’t won a championship in over 20 years, or is on a losing streak for the entire tournament. It is incredible the sense of ownership and pride you feel in becoming part of a group. Most Georgians rally behind the Atlanta Falcons, not necessarily because they are the best team on the roster, but because you feel a sense of duty and pride to support your local team. A victory for the Falcons is a victory for Atlanta, and Georgia in general.

As a group, we have a great interest in analyzing betting on sports. We know the average person might just put a couple of dollars behind their hometown team in hopes of winning, but what if there was a way to determine and analyze the scores each player has in order to positively correlate who is winning in the season or which player might get the most recognitions.

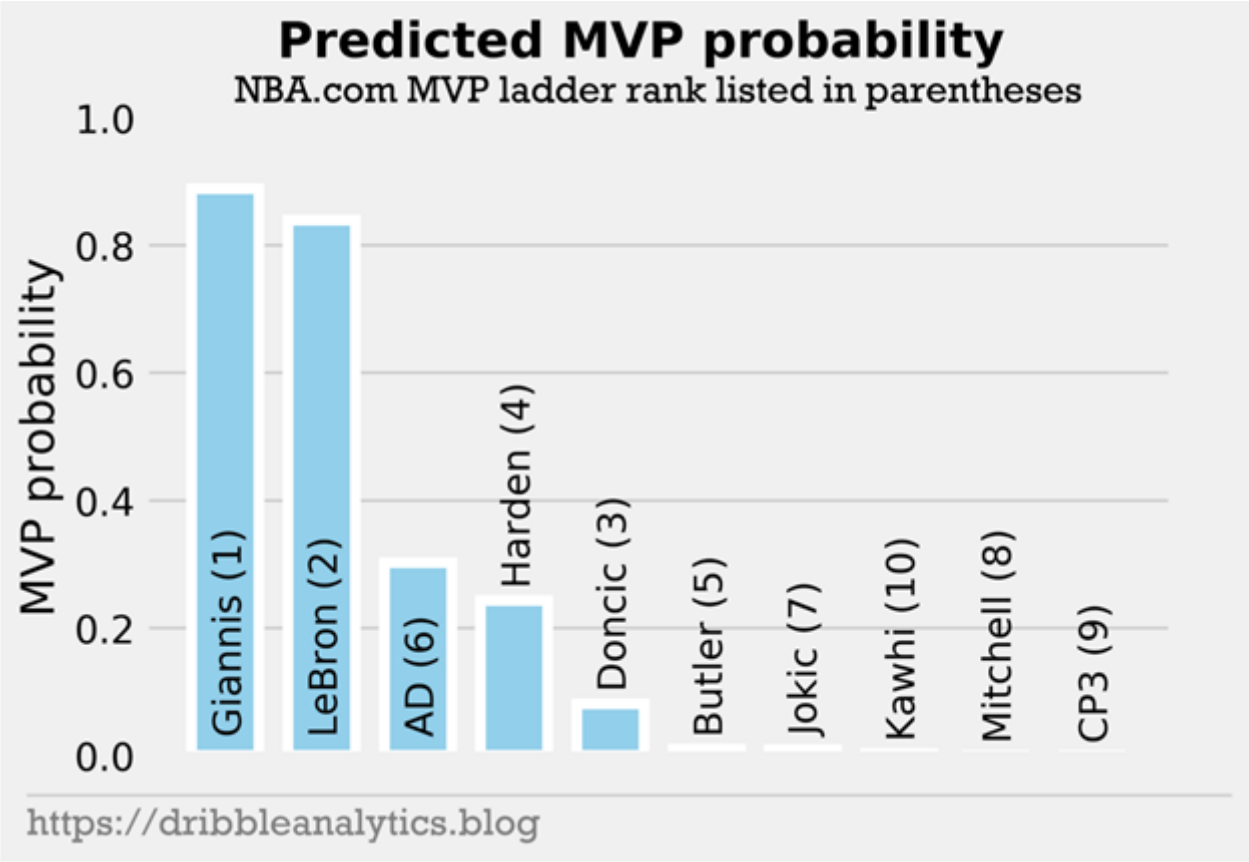
We want to analyze variables such as VORP, BPM, PER (each variable is explained in detail below), and others to see if we can find a significant correlation that can provide more substantive grounds to sports betting. Although these are not the only variables we will analyze and take into account when running our regression, we believe these will be the ones most influential in the final regression results. We don't believe as a group that awards and recognitions in sports are a matter of luck, we believe we will be able to a certain extent to be able to predict which players and which teams are most likely to excel in certain categories. Our biggest motivation is to be able to provide a way for the average basketball fan to plug-in data and statistics about their team and be able to play with the outcomes that are displayed.

We acknowledge that this project is heavily based on the outsourcing of basketball data, and blindly place our trust in this data we found on the internet. Although we do believe this data provides valuable insight and can be traced back to reputable sources, we understand that an error in the data will significantly alter our results. Basketball statistics is not a hard science, and for this reason, results will not always be 100% representative of real-world outcomes. We have tried our best to filter out and select the most reputable sources, but as with anything, we include this caveat into our project to be completely transparent with our users.

More than anything, as a group, we researched and found great inspirational sources (found below) for our project. The entirety of our group is interested in sports and we found a topic that interested and intrigued us all. We decided to base our project on basketball statistics because it gave us a lens into the inner workings of basketball, all the while being able to keep up with a sport we truly enjoy following.

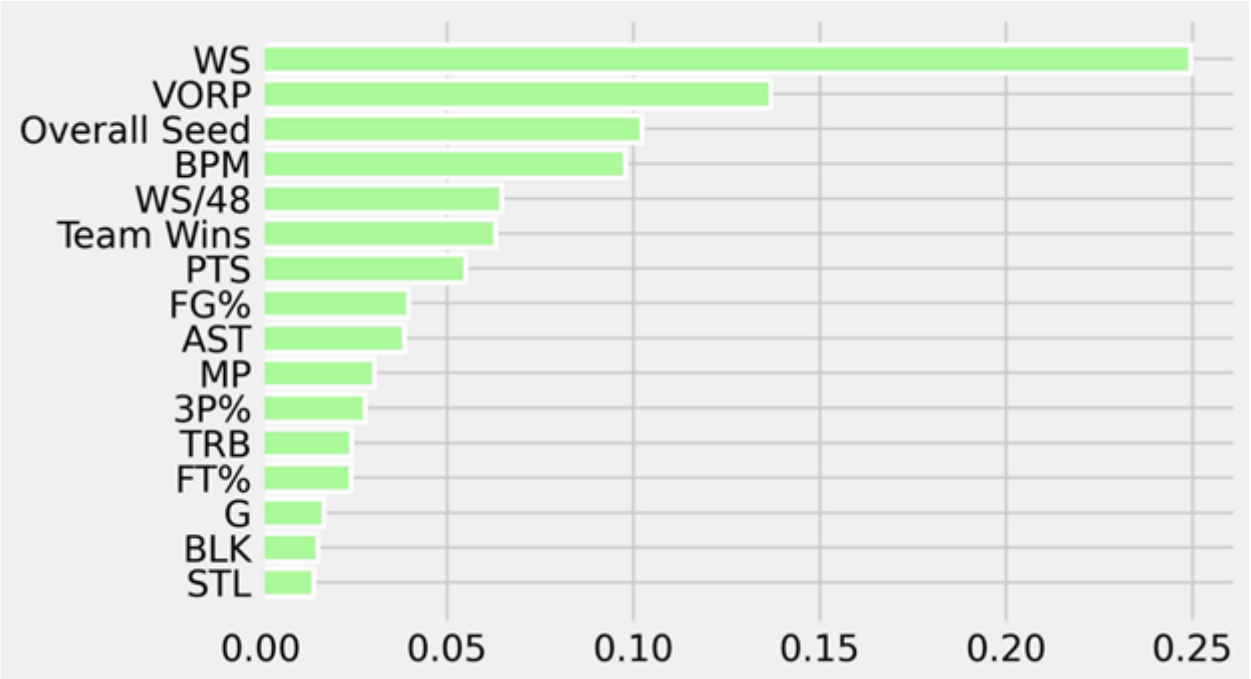
External Sources and Related Literature

Prediction of MVP attribution in NBA regular match based on a statistical model has been previously created using simple linear models such as logistic regression and linear discriminant analysis by Dribble Analytics. In their model, their data set consists of every player in the top-10 of MVP voting since the 1984-1985 season. They collected all the counting and advanced stats available on Basketball-Reference for each player. In total, they have 349 samples. Since team success and narrative play a large role in the voting and are not easy to quantify, the model is limited. Nonetheless, their model predicted about 74.3% of MVP winners and identified a higher percentage of top-3 and top-5 finishers.

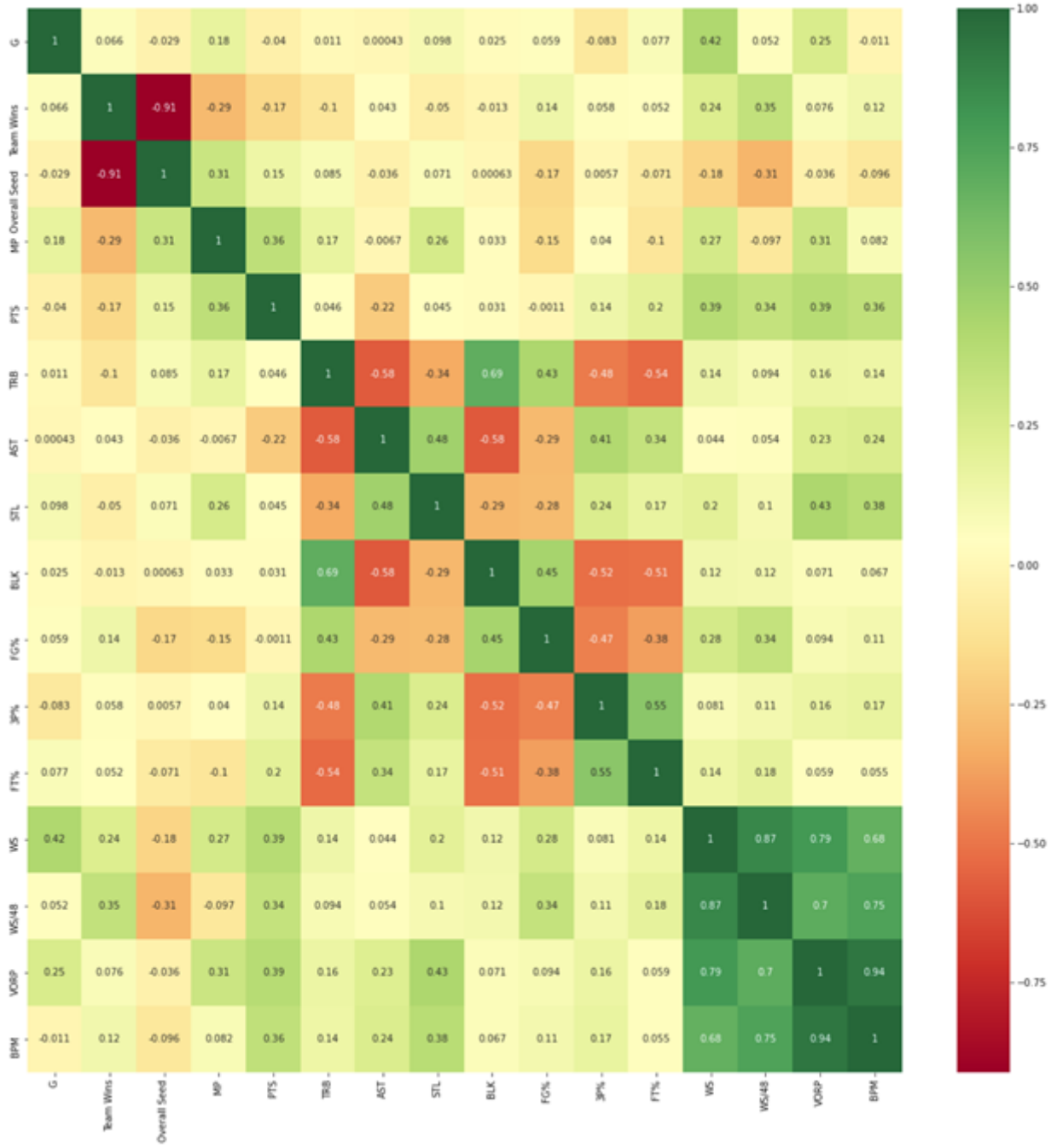


The above graph is the prediction of MVP for the 2020 year. The big drop in MVP probability from LeBron to Anthony Davis is likely due to the model valuing winning a lot. Despite the limitations of the model and the disruption of the sport due to the pandemic, it still correctly predicted the MVP for 2020, Giannis Antetokounmpo.

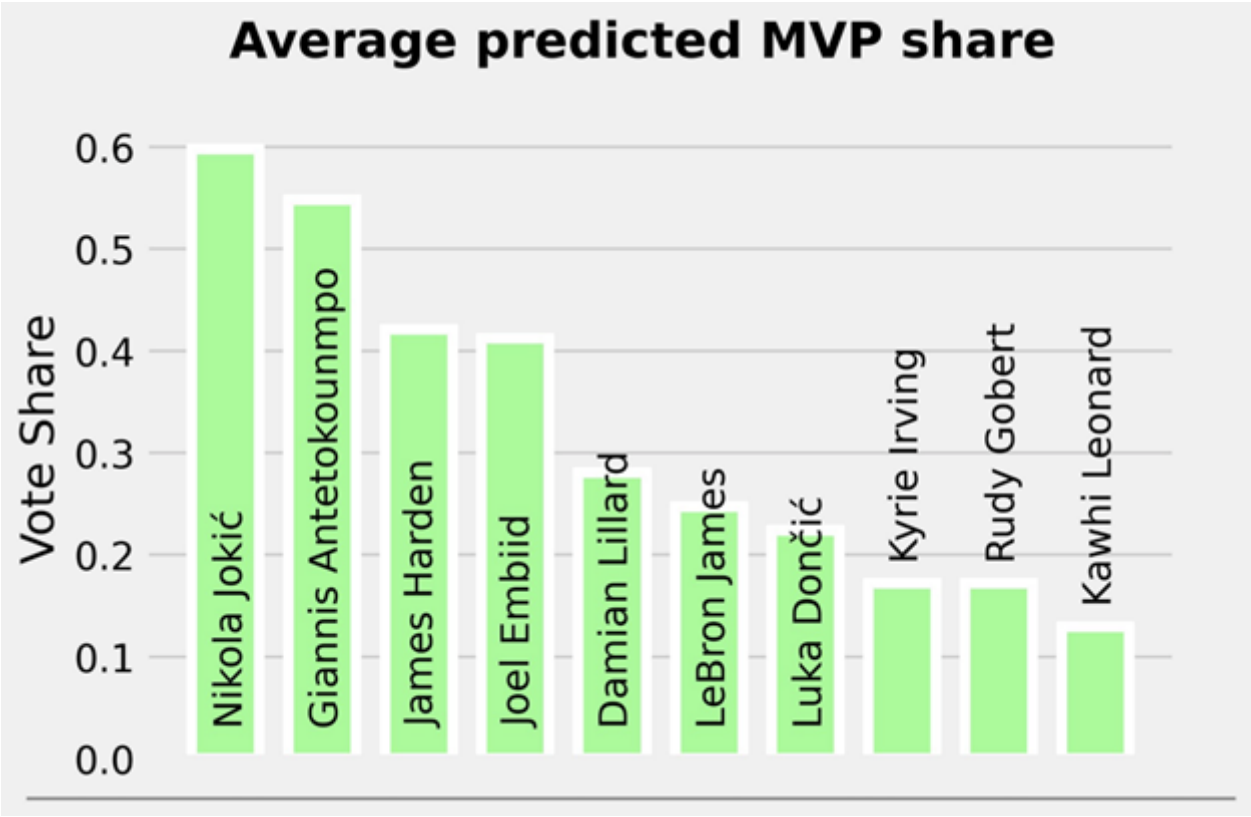
Another predictive model of MVP attribution was done by Towards Data Science. They used data pertaining to the top 10 players in MVP voting of each season, from 1979–80 to 2019–20. The data originally came with 16 variables or features for each player. To optimize the model and inspect which variables or features are most important, a random forest regression was performed:



Furthermore, to find the most correlated variables and avoid feeding the model duplicated information, a correlation matrix was done:


G 1 0.066 -0.029 0.18 -0.04 0.011 0.00043 0.098 0.025 0.059 -0.083 0.077 0.42 0.052 0.25 -0.011
Team Wins 0.066 1 -0.91 -0.29 -0.17 -0.1 0.043 -0.05 -0.013 0.14 0.058 0.052 0.24 0.35 0.076 0.12
Overall Seed -0.029 -0.91 1 0.31 0.15 0.085 -0.036 0.071 0.00063 -0.17 0.0057 -0.071 -0.18 -0.31 -0.036 -0.096
MP 0.18 -0.29 0.31 1 0.36 0.17 -0.0067 0.26 0.033 -0.15 0.04 -0.1 0.27 -0.097 0.31 0.082
PTS -0.04 -0.17 0.15 0.36 1 0.046 -0.22 0.045 0.031 -0.0011 0.14 0.2 0.39 0.34 0.39 0.36
TRB 0.011 -0.1 0.085 0.17 0.046 1 -0.58 -0.34 0.69 0.43 -0.48 -0.54 0.14 0.094 0.16 0.14
AST 0.00043 0.043 -0.036 -0.0067 -0.22 -0.58 1 0.48 -0.58 -0.29 0.41 0.34 0.044 0.054 0.23 0.24
STL 0.098 -0.05 0.071 0.26 0.045 -0.34 0.48 1 -0.29 -0.28 0.24 0.17 0.2 0.1 0.43 0.38
BLK 0.025 -0.013 0.00063 0.033 0.031 0.69 -0.58 -0.29 1 0.45 -0.52 -0.51 0.12 0.12 0.071 0.067
FG% 0.059 0.14 -0.17 -0.15 -0.0011 0.43 -0.29 -0.28 0.45 1 -0.47 -0.38 0.28 0.34 0.094 0.11
3P% -0.083 0.058 0.0057 0.04 0.14 -0.48 0.41 0.24 -0.52 -0.47 1 0.55 0.081 0.11 0.16 0.17
FT% 0.077 0.052 -0.071 -0.1 0.2 -0.54 0.34 0.17 -0.51 -0.38 0.55 1 0.14 0.18 0.059 0.055
WS 0.42 0.24 -0.18 0.27 0.39 0.14 0.044 0.2 0.12 0.28 0.081 0.14 1 0.87 0.79 0.68
WS/48 0.052 0.35 -0.31 -0.097 0.34 0.094 0.054 0.1 0.12 0.34 0.11 0.18 0.87 1 0.7 0.75
VORP 0.25 0.076 -0.036 0.31 0.39 0.16 0.23 0.43 0.071 0.094 0.16 0.059 0.79 0.7 1 0.94
BPM -0.011 0.12 -0.096 0.082 0.36 0.14 0.24 0.38 0.067 0.11 0.17 0.055 0.68 0.75 0.94 1
G
Team Wins
Overall Seed
MP
PTS
TRB
AST
STL
BLK
FG%
3P%
FT%
WS
WS/48
VORP
BPM
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

After adjustments, seven features are removed based on the correlation martix and regression. They then used the three advanced models: Deep Neural Network (DNN), k-nearest neighbors regression (KNN), and Random forest regression (RF). They relied on the Mean Squared Error test (MSE) to assess the accuracy of the four different models. A lower MSE indicates a higher R-squared which means a more accurate model. The most accurate was KNN in their experiment.



The above graph shows the average probability of each player winning MVP for 2020. We can see that the above model did not accurately predict the MVP for 2020 but it was very close as the second candidate, Giannis, was the actual winner of the title.

Data

For this project, we will be using data from basketball-reference.com, which is a website that keeps track of team and player stats for every game and every season in the NBA. We compiled the data from this website for the past 15 seasons, including both the Per Game and Advanced data, into a csv file, and also compiled the data from this season so far (as of April 1, 2021) into a csv file. These have both been placed in a Github repository for ease of use and access.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
import statsmodels.api as sm
import scipy.stats as stats
!pip install stargazer
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
```

```
Requirement already satisfied: stargazer in /Users/forrestmartin/opt/anaconda3/lib/python3.8/site-packages (0.
0.5)
```

In [3]:
```
pre_data = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/Overall%20Data%20pre
pre_data.head()
```

Out[3]:

	Player	Pos	Age	Team	Games	Gms_started	MinPerG	FG	FGA	FG_pct	...	OWS	DWS	WS	WSp48	OBPM	DBP
0	Steven Adams\adamsst01	C	26	OKC	63	63	26.7	4.5	7.6	0.592	...	3.8	2.7	6.5	0.185	1.9	1
1	Bam Adebayo\adebaba01	PF	22	MIA	72	72	33.6	6.1	11.0	0.557	...	4.6	3.9	8.5	0.168	1.4	2
2	LaMarcus Aldridge\aldrila01	C	34	SAS	53	53	33.1	7.4	15.0	0.493	...	3.0	1.4	4.5	0.122	1.8	-0
3	Kyle Alexander\alexaky01	C	23	MIA	2	0	6.5	0.5	1.0	0.500	...	0.0	0.0	0.0	-0.003	-6.1	-3
4	Nickeil Alexander-Walker\alexani01	SG	21	NOP	47	1	12.6	2.1	5.7	0.368	...	-0.7	0.4	-0.2	-0.020	-3.2	-1

5 rows × 51 columns

In [4]:
```
current_data = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/2020-21%20Overal
current_data.head()
```

Out[4]:

	Player	Pos	Age	Team	Games	Gms_started	MinPerG	FG	FGA	FG_pct	...	OWS	DWS	WS	WSp48	OBPM	DBPI
0	Precious Achiuwa\achiupr01	PF	21	MIA	57	3	12.0	2.0	3.6	0.538	...	0.2	1.0	1.2	0.085	-3.6	-0.

	Player	Pos	Age	Team	Games	Gms_started	MinPerG	FG	FGA	FG_pct	...	OWS	DWS	WS	WSp48	OBPM	DBPI
1	Jaylen Adams\adamsja01	PG	24	MIL	7	0	2.6	0.1	1.1	0.125	...	-0.1	0.0	-0.1	-0.249	-14.9	-5.
2	Steven Adams\adamsst01	C	27	NOP	57	57	27.7	3.3	5.4	0.620	...	2.4	1.6	4.0	0.120	-0.2	-0.
3	Bam Adebayo\adebaba01	C	23	MIA	58	58	33.5	7.2	12.7	0.569	...	5.1	3.0	8.1	0.200	3.0	1.
4	LaMarcus Aldridge\aldrila01	C	35	TOT	26	23	25.9	5.4	11.4	0.473	...	0.5	0.7	1.2	0.084	-0.2	-0.

5 rows × 51 columns

Methodology/Empirical Model

In the data for the 15 seasons prior to 2020-21, we included a variable, "MVP", which is a binary variable that indicates whether a player won the MVP that season or not (1 for Win, 0 for No Win). In order to determine which of the variables is potentially most important to our model, we ran a correlation test to see which of the variables from the past 15 seasons of data was most highly correlated with a player winning the MVP.

In [5]:
```
cormat = pre_data[['MVP','VORP','BPM','PTS','Age','Gms_started','PER','USG','WS','3P_pct','2P_pct']]

cormat.corr().round(decimals=3)
```

Out[5]:

	MVP	VORP	BPM	PTS	Age	Gms_started	PER	USG	WS	3P_pct	2P_pct
MVP	1.000	0.260	0.041	0.138	-0.000	0.082	0.102	0.103	0.200	0.028	0.028
VORP	0.260	1.000	0.218	0.722	0.067	0.632	0.526	0.382	0.912	0.191	0.201
BPM	0.041	0.218	1.000	0.221	0.035	0.162	0.411	0.096	0.223	0.143	0.219
PTS	0.138	0.722	0.221	1.000	0.026	0.723	0.588	0.606	0.754	0.318	0.247
Age	-0.000	0.067	0.035	0.026	1.000	0.043	-0.005	-0.121	0.064	0.049	-0.014
Gms_started	0.082	0.632	0.162	0.723	0.043	1.000	0.390	0.258	0.757	0.163	0.190
PER	0.102	0.526	0.411	0.588	-0.005	0.390	1.000	0.377	0.551	0.184	0.530

	MVP	VORP	BPM	PTS	Age	Gms_started	PER	USG	WS	3P_pct	2P_pct
USG	0.103	0.382	0.096	0.606	-0.121	0.258	0.377	1.000	0.304	0.184	0.061
WS	0.200	0.912	0.223	0.754	0.064	0.757	0.551	0.304	1.000	0.165	0.284
3P_pct	0.028	0.191	0.143	0.318	0.049	0.163	0.184	0.184	0.165	1.000	-0.015
2P_pct	0.028	0.201	0.219	0.247	-0.014	0.190	0.530	0.061	0.284	-0.015	1.000

With these results, we saw that the top five variables in terms of correlation to MVP were VORP, PER, PTS, USG, and WS. After choosing these variables, we estimated our model to be *MVP = B0 + VORP(B1) + PER(B2) + PTS(B3) + USG(B4) + WS(B5) + u*. Because MVP is a binary variable, we will be running a logistic regression on these variables from the prior 15 seasons to find the coefficients, and then applying those coefficients to the dataset from the current season to see which player has the best score as calculated by the model.

Additionally, below is a description of each of the variables within the model and what they mean.

Variables of Interest

Based on the results of the above correlation table, the variables with the highest correlation with "MVP" will be chosen for inclusion in our model. With that, we have chosen these 7 variables for our model: VORP, BPM, PER, PTS, USG, WS, and AST.

VORP: Value over replacement player - basically do you produce more points/prevent more of the opponents points than the player who comes in for you as a sub.

BPM: Box Plus/Minus - a basketball box score-based metric that estimates a basketball player’s contribution to the team when that player is on the court. BPM uses a player’s box score information, position, and the team’s overall performance to estimate the player’s contribution in points above league average per 100 possessions played. BPM does not take into account playing time, and the league average is defined as 0.0.

PER: Player Efficiency Rating - this stat essentially sums up all of a player's positive accomplishments, subtracts the negative accomplishments, and returns a per-minute rating of a player's performance.

PTS: the average number of points that a player scores per game.

USG: Usage Pct. - calculates what percentage of team plays a player was involved in while he was on the floor, provided that the play ends in one of the three true results: field-goal attempt, free-throw attempt or turnover. On average, a player will have a usage rate of 20 percent, because there are 5 players on the court from each team.

WS: Win Share per 48 minutes - an estimate of the number of wins contributed by a player. This is a measure of a players impact on their teams ability to win. This stat attempts to divvy up credit for team success to the individuals on the team, where the sum of player win shares on a given team will be roughly equal to that team's win total for the season.

AST: Assists - the average number of assists that a player gives per game.

In [6]:
```
past_mvps = {'Year': ['2019-20', '2018-19', '2017-18', '2016-17', '2015-16', '2014-15', '2013-14', '2012-13', '
past_mvp_df = (pd.DataFrame(data=past_mvps))
print(past_mvp_df)
print("*In the 2011-12 season, there was an NBA Lockout which caused the season to start in December rather tha
print("*ASG = All-Star Game, typically considered the point of the season where teams either make a push or the
```

```
         Year                   MVP  Regular Season Wins  Wins Pre-ASG
0     2019-20  Giannis Antetokounmpo                   56            46
1     2018-19  Giannis Antetokounmpo                   60            43
2     2017-18           James Harden                   65            44
3     2016-17      Russell Westbrook                   47            32
4     2015-16          Stephen Curry                   73            48
5     2014-15          Stephen Curry                   67            42
6     2013-14           Kevin Durant                   59            43
7     2012-13           LeBron James                   66            36
8   *2011-12*           LeBron James                   46            27
9     2010-11           Derrick Rose                   62            38
10    2009-10           LeBron James                   61            43
11    2008-09           LeBron James                   66            40
12    2007-08            Kobe Bryant                   57            35
13    2006-07          Dirk Nowitzki                   67            44
14    2005-06             Steve Nash                   54            35
*In the 2011-12 season, there was an NBA Lockout which caused the season to start in December rather than Octob
er, shortening the number of games from 82 to 66.
*ASG = All-Star Game, typically considered the point of the season where teams either make a push or the fatigu
e hits them and they decline.
```

In all but one of the past 15 NBA seasons, the regular season MVP award has been given to a player on a team which had over 50 wins, which is a 0.61 win percentage (if the full 2011-12 season had been played, the Miami Heat likely would have had over 50 wins, so this is not including that particular 2011-12 season). Also, in 9 of the past 15 seasons, the MVP's team reached 40 wins even before the All-Star Game, which is a strong early indicator of a good record.

So, while we will not be including team wins into our model because that is a team stat and does not reflect the individual prowess of the player during that season, it is something that should be watched and potentially used as some kind of check on our model. This could potentially prevent us from predicting that a player who is on the worst team in the league, but is scoring at an prodigious rate because no one else on their team is able to, should be MVP. While they may be incredibly valuable to their team, if their team is still

at the bottom of the league they likely shouldn't win the MVP award over a player who has helped make their team a contender for the NBA title.

Analysis

In [8]:
```
pre_data = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/Overall%20Data%20pre
current_data = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/2020-21%20Overal
over20gms_data = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/pre2020-21%20c
```

Model 1

Explanation of Model 1

Our first model is derived from a correlation table in which we sought to find the variables that were most highly correlated with MVP, with data from the previous 15 seasons in their entirety, unfiltered.

In [9]:
```
#Model 1
# loading the training dataset, which is a dataset of all players in the past 15 seasons
# also creating a correlation table to see which variables are best correlated with MVP
train_df1 = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/Overall%20Data%20pr

cormat = train_df1[['MVP','VORP','BPM','PTS','Age','Gms_started','PER','USG','WS','WSp48','3P_pct','2P_pct', 'A

cormat.corr().round(decimals=3)
```

Out[9]:

	MVP	VORP	BPM	PTS	Age	Gms_started	PER	USG	WS	WSp48	3P_pct	2P_pct	AST	STL
MVP	1.000	0.260	0.041	0.138	-0.000	0.082	0.102	0.103	0.200	0.080	0.028	0.028	0.126	0.084
VORP	0.260	1.000	0.218	0.722	0.067	0.632	0.526	0.382	0.912	0.423	0.191	0.201	0.570	0.577
BPM	0.041	0.218	1.000	0.221	0.035	0.162	0.411	0.096	0.223	0.409	0.143	0.219	0.162	0.200
PTS	0.138	0.722	0.221	1.000	0.026	0.723	0.588	0.606	0.754	0.390	0.318	0.247	0.657	0.681
Age	-0.000	0.067	0.035	0.026	1.000	0.043	-0.005	-0.121	0.064	0.071	0.049	-0.014	0.096	0.032
Gms_started	0.082	0.632	0.162	0.723	0.043	1.000	0.390	0.258	0.757	0.298	0.163	0.190	0.526	0.592
PER	0.102	0.526	0.411	0.588	-0.005	0.390	1.000	0.377	0.551	0.890	0.184	0.530	0.362	0.408
USG	0.103	0.382	0.096	0.606	-0.121	0.258	0.377	1.000	0.304	0.009	0.184	0.061	0.383	0.279
WS	0.200	0.912	0.223	0.754	0.064	0.757	0.551	0.304	1.000	0.486	0.165	0.284	0.508	0.566

	MVP	VORP	BPM	PTS	Age	Gms_started	PER	USG	WS	WSp48	3P_pct	2P_pct	AST	STL
WSp48	0.080	0.423	0.409	0.390	0.071	0.298	0.890	0.009	0.486	1.000	0.177	0.519	0.214	0.291
3P_pct	0.028	0.191	0.143	0.318	0.049	0.163	0.184	0.184	0.165	0.177	1.000	-0.015	0.314	0.275
2P_pct	0.028	0.201	0.219	0.247	-0.014	0.190	0.530	0.061	0.284	0.519	-0.015	1.000	0.069	0.157
AST	0.126	0.570	0.162	0.657	0.096	0.526	0.362	0.383	0.508	0.214	0.314	0.069	1.000	0.674
STL	0.084	0.577	0.200	0.681	0.032	0.592	0.408	0.279	0.566	0.291	0.275	0.157	0.674	1.000

After viewing this correlation table, we chose the five variables "VORP", "PTS", "PER", "USG", and "WS". They were the most highly correlated with MVP and we felt that they would do a good job representing the data.

In [11]:
```
#regression of Model 1
# defining the dependent and independent variables
Xtrain = train_df1[['Intercept','VORP','PTS', 'PER', 'USG', 'WS']]
ytrain = train_df1[['MVP']]

# building the model and fitting the data
log_reg1 = sm.Logit(ytrain, Xtrain).fit()

# printing the summary table
st1=Stargazer([log_reg1])

from IPython.core.display import HTML
HTML(st1.render_html())
```

```
Optimization terminated successfully.
         Current function value: 0.003736
         Iterations 14
```

Out[11]:

	Dependent variable:MVP
	(1)
Intercept	-19.472***
	(3.839)
PER	0.013
	(0.030)
PTS	-0.107

	(0.094)
USG	0.324***
	(0.114)
VORP	-0.074
	(0.428)
WS	0.720**
	(0.281)
Observations	9,041
R^2	
Adjusted R^2	
Residual Std. Error	1.000 (df=9035)
F Statistic	(df=5; 9035)
Note:	*p<0.1; **p<0.05; ***p<0.01

Our equation based on the Model 1 regression is:

$$HighScore = \beta_0 + \beta_1 * VORP + \beta_2 * PTS + \beta_3 * PER + \beta_4 * USG + \beta_5 * WS + u$$

High Score is the variable that we created to encapsulate all of the different variables into one score that we could use to determine the strength of a player's season. It is the variable we are using to measure whether a player deserves to win MVP.

Below, we calculate the "High Score" variable and place it into the dataframe, then order the top 10 results from the dataframe by their "High Score" value.

In [14]:
```
#Model 1 Test on 2020-21 data
#using 'Overall pre2020' data generated coefficients on current years data to evaluate the best player/who shou
test_dataset = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/2020-21%20Overal
test_df1 = test_dataset.filter(['Player','Team','Games','VORP','PTS','PER','USG','WS'], axis=1)

test_df1['HighScore'] = log_reg1.params.values[0] + (log_reg1.params.values[1])*test_df1['VORP'] + (log_reg1.pa

test_df2 = test_df1.sort_values(by='HighScore', ascending=False)
test_df2.head(10)
```

Out[14]:

Player	Team	Games	VORP	PTS	PER	USG	WS	HighScore
Nikola Jokić\jokicni01	DEN	64	7.7	26.2	31.1	29.2	14.0	-2.878762
Joel Embiid\embiijo01	PHI	45	3.5	29.3	30.8	35.5	8.1	-5.111384
Giannis Antetokounmpo\antetgi01	MIL	54	5.0	28.4	29.4	32.9	9.2	-5.195044
Luka Dončić\doncilu01	DAL	58	4.8	28.6	25.9	35.9	7.4	-5.573183
Stephen Curry\curryst01	GSW	56	4.9	31.3	26.1	34.0	7.9	-6.120811
Damian Lillard\lillada01	POR	59	3.7	28.3	24.5	31.7	8.2	-6.262211
Zion Williamson\willizi01	NOP	59	3.7	27.0	27.4	29.8	8.6	-6.412292
Anžejs Pasečņiks\pasecan01	WAS	1	-0.1	0.0	-40.7	41.5	-0.1	-6.637890
Donovan Mitchell\mitchdo01	UTA	53	2.5	26.4	21.3	33.6	6.2	-6.838742
Trae Young\youngtr01	ATL	56	2.8	25.5	22.9	33.0	6.3	-6.866084

When we ran that model on this season's data, we could tell that it clearly gives players who should not win MVP. So we decided to create a new model that filters for players that have played at least 20 games, call it "Model 2" and see if it predicts any better.

Model 2

Explanation of Model 2

After viewing Model 1's results, we found that there were a couple of players that were not deserving of winning the MVP award in the top 10 values of "High Score", namely Anžejs Pasečniks and Trae Young. Therefore, we decided to create a new model, based on data that had been filtered for players who had played at least 20 games, and see if it gave better predictions.

Upon beginning to create this model, we also realized that by filtering the dataset we might be changing which variables were most highly correlated with MVP, and so we ran another correlation table to test this.

In [12]:
```
#Model 2
# loading the training dataset, which in this case is a dataset filtered for players who have played at least 2
#also creating a new correlation table to see if the filtered dataset produced different correlations with the
train_df2 = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/pre2020-21%20over%2

cormat2 = train_df2[['MVP','VORP','BPM','PTS','Age','Gms_started','PER','USG','WS','WSp48','3P_pct','2P_pct', '
```

```
cormat2.corr().round(decimals=3)
```

Out[12]:

	MVP	VORP	BPM	PTS	Age	Gms_started	PER	USG	WS	WSp48	3P_pct	2P_pct	AST	STL
MVP	1.000	0.263	0.176	0.149	-0.002	0.080	0.159	0.128	0.210	0.147	0.030	0.044	0.131	0.090
VORP	0.263	1.000	0.850	0.736	0.059	0.604	0.758	0.473	0.919	0.707	0.179	0.275	0.566	0.592
BPM	0.176	0.850	1.000	0.686	0.096	0.507	0.859	0.400	0.801	0.860	0.238	0.423	0.510	0.586
PTS	0.149	0.736	0.686	1.000	-0.017	0.699	0.751	0.758	0.739	0.491	0.262	0.189	0.622	0.649
Age	-0.002	0.059	0.096	-0.017	1.000	0.018	-0.031	-0.126	0.044	0.083	0.041	-0.049	0.073	-0.003
Gms_started	0.080	0.604	0.507	0.699	0.018	1.000	0.488	0.319	0.711	0.387	0.107	0.180	0.491	0.573
PER	0.159	0.758	0.859	0.751	-0.031	0.488	1.000	0.619	0.761	0.835	0.040	0.505	0.449	0.475
USG	0.128	0.473	0.400	0.758	-0.126	0.319	0.619	1.000	0.385	0.199	0.201	-0.026	0.469	0.368
WS	0.210	0.919	0.801	0.739	0.044	0.711	0.761	0.385	1.000	0.772	0.107	0.368	0.473	0.541
WSp48	0.147	0.707	0.860	0.491	0.083	0.387	0.835	0.199	0.772	1.000	0.018	0.601	0.244	0.325
3P_pct	0.030	0.179	0.238	0.262	0.041	0.107	0.040	0.201	0.107	0.018	1.000	-0.171	0.296	0.243
2P_pct	0.044	0.275	0.423	0.189	-0.049	0.180	0.505	-0.026	0.368	0.601	-0.171	1.000	-0.038	0.068
AST	0.131	0.566	0.510	0.622	0.073	0.491	0.449	0.469	0.473	0.244	0.296	-0.038	1.000	0.667
STL	0.090	0.592	0.586	0.649	-0.003	0.573	0.475	0.368	0.541	0.325	0.243	0.068	0.667	1.000

Once we created the correlation table, we found that BPM (Box Plus/Minus) and AST (Assists) had increased significantly in their correlation with MVP, and decided to include them in our second model as well, in order to increase our predictive power.

In [19]:

```
#still Model 2: making a model based on the over20games dataset and including the extra variables, BPM and AST
# defining the dependent and independent variables
Xtrain3 = train_df2[['Intercept','VORP','BPM','PTS', 'PER', 'USG', 'WS','AST']]
ytrain3 = train_df2[['MVP']]

# building the model and fitting the data
log_reg3 = sm.Logit(ytrain3, Xtrain3).fit()

# printing the summary table

st2=Stargazer([log_reg3])
```

```
from IPython.core.display import HTML
HTML(st2.render_html())
```

```
Optimization terminated successfully.
         Current function value: 0.003683
         Iterations 15
```

Out[19]:

	Dependent variable:MVP
	(1)
AST	0.173
	(0.181)
BPM	1.930**
	(0.798)
Intercept	-22.588***
	(6.863)
PER	-0.593
	(0.487)
PTS	-0.281
	(0.234)
USG	0.753**
	(0.310)
VORP	-3.144**
	(1.283)
WS	1.815***
	(0.621)
Observations	7,081
R^2	
Adjusted R^2	

Residual Std. Error	1.000 (df=7073)
F Statistic	(df=7; 7073)

Note: *p<0.1; **p<0.05; ***p<0.01

Our second equation, which is based on the Model 2 regression, is:

$$HighScore = \beta_0 + \beta_1 * VORP + \beta_2 * BPM + \beta_3 * PTS + \beta_4 * PER + \beta_5 * USG + \beta_6 * WS + \beta_7 * AST + u$$

Like we did for Model 1, we now calculate the "High Score" variable and place it into the dataframe, then order the top 10 results from the dataframe by their "High Score" value.

The results for this season's players with the Top 10 High Score values is below:

In [15]:
```
#Model 2 Test on 2020-21 data
#using 'over20gms' data generated coefficients on current years data to evaluate the best player/who should wir
test_dataset2 = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/2020-21%20Overa

test_df_new = test_dataset2.filter(['Games','VORP','BPM','PTS','PER','USG','WS','AST'], axis=1)

test_df_new['HighScore'] = log_reg3.params.values[0] + (log_reg3.params.values[1])*test_df_new['VORP'] + (log_r

test_df3 = test_df_new.sort_values(by='HighScore', ascending=False)

test_df3.head(10)
```

Out[15]:

Player	Games	VORP	BPM	PTS	PER	USG	WS	AST	HighScore
Nikola Jokić\jokicni01	64	7.7	11.6	26.2	31.1	29.2	14.0	8.5	-1.339691
Joel Embiid\embiijo01	45	3.5	7.7	29.3	30.8	35.5	8.1	3.0	-3.270889
Giannis Antetokounmpo\antetgi01	54	5.0	9.1	28.4	29.4	32.9	9.2	5.9	-3.662609
Luka Dončić\doncilu01	58	4.8	7.3	28.6	25.9	35.9	7.4	8.9	-4.979645
LeBron James\jamesle01	43	3.5	7.5	25.0	24.2	31.6	5.5	7.8	-5.363218
Jimmy Butler\butleji01	48	3.9	7.4	21.5	26.4	26.7	8.5	7.2	-5.480976
Stephen Curry\curryst01	56	4.9	8.1	31.3	26.1	34.0	7.9	5.8	-5.687120
James Harden\hardeja01	34	2.8	7.0	25.4	24.9	28.7	5.7	11.0	-5.922195

Player	Games	VORP	BPM	PTS	PER	USG	WS	AST	HighScore
Kawhi Leonard\leonaka01	47	3.6	6.9	25.5	26.8	29.0	8.1	5.1	-6.221904
Donovan Mitchell\mitchdo01	53	2.5	3.5	26.4	21.3	33.6	6.2	5.2	-6.286621

Based on these results, Model 2 has potential to be better, but it is still yet unclear which model has the best prediction until we look to see how well it would predict the past years MVP's based on those data. Due to that, we now run the two models on the past 15 seasons of data in order to see which model predicts the most MVP winners.

Testing for Validity of Model 1 and Model 2

Theoretically, if our model were perfect, the top 15 values of this dataframe would all have "1" in the MVP column, because it would be so good at predicting MVP that the 15 MVPs in the dataset would all show up in the top 15 rows. But obviously this isn't a perfect model, so instead we just aim to get as good as we can.

In [16]:
```
#using 'Overall pre2020' generated coefficients on its own data to test the models efficacy
confirm_data = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/Overall%20Data%2

confirm_df1 = confirm_data.filter(['VORP','PTS','PER','USG','WS','MVP'], axis=1)
confirm_df1['HighScore'] = log_reg1.params.values[0] + (log_reg1.params.values[1])*confirm_df1['VORP'] + (log_r

confirm_df2 = confirm_df1.sort_values(by='HighScore', ascending=False)
confirm_df2.head(15)
```

Out[16]:

Player	VORP	PTS	PER	USG	WS	MVP	HighScore
LeBron James\jamesle01	11.8	28.4	31.7	33.8	20.3	1	2.618914
Kevin Durant\duranke01	9.6	32.0	29.8	33.0	19.2	1	1.321699
LeBron James\jamesle01	10.3	29.7	31.1	33.5	18.5	1	1.189819
LeBron James\jamesle01	9.9	26.8	31.6	30.2	19.3	1	1.042688
Kevin Durant\duranke01	8.9	28.1	28.3	29.8	18.9	0	0.516717
Stephen Curry\curryst01	9.5	30.1	31.5	32.6	17.9	1	0.488149
James Harden\hardeja01	9.3	36.1	30.6	40.5	15.2	0	0.465227
Kobe Bryant*\bryanko01	8.0	35.4	28.0	38.7	15.3	0	0.090713

Player	VORP	PTS	PER	USG	WS	MVP	HighScore
Dirk Nowitzki\nowitdi01	7.9	26.6	28.1	30.0	17.7	0	-0.051616
James Harden\hardeja01	7.7	30.4	29.8	36.1	15.4	1	-0.100323
Chris Paul\paulch01	9.9	22.8	30.0	27.5	18.3	0	-0.147234
Russell Westbrook\westbru01	9.3	31.6	30.6	41.7	13.1	1	-0.179434
LeBron James\jamesle01	9.4	31.4	28.1	33.6	16.3	0	-0.516962
Dwyane Wade\wadedw01	9.6	30.2	30.4	36.2	14.7	0	-0.683863
James Harden\hardeja01	8.1	27.4	26.7	31.3	16.4	0	-0.685803

Model 1 predicts accurately the MVP for 7 of the 15 seasons. 7 out of 15 is not a terrific prediction accuracy, though, as it is not even above a 50/50 chance.

Now we test Model 2.

In [24]:
```
#using Model 2('over20gms' dataset) coefficients on its own dataset to test the models efficacy
gms20df = pd.read_csv('https://raw.githubusercontent.com/forrestm99/nba-econ320-data/main/pre2020-21%20over%202

confirm_dfnew = gms20df.filter(['VORP','BPM','PTS','PER','USG','WS','AST','MVP'], axis=1)
confirm_dfnew['HighScore'] = log_reg3.params.values[0] + (log_reg3.params.values[1])*confirm_dfnew['VORP'] + (l

confirm_df3 = confirm_dfnew.sort_values(by='HighScore', ascending=False)
confirm_df3.head(15)
```

Out[24]:

Player	VORP	BPM	PTS	PER	USG	WS	AST	MVP	HighScore
James Harden\hardeja01	7.7	9.9	30.4	29.8	36.1	15.4	8.8	1	2.757508
LeBron James\jamesle01	11.8	13.2	28.4	31.7	33.8	20.3	7.2	1	2.557464
LeBron James\jamesle01	9.9	11.7	26.8	31.6	30.2	19.3	7.3	1	1.637148
Stephen Curry\curryst01	9.5	11.9	30.1	31.5	32.6	17.9	6.7	1	1.574116
LeBron James\jamesle01	10.3	11.8	29.7	31.1	33.5	18.5	8.6	1	1.310512
Giannis Antetokounmpo\antetgi01	6.6	11.5	29.5	31.9	37.5	11.1	5.6	1	1.008587
Kevin Durant\duranke01	9.6	10.2	32.0	29.8	33.0	19.2	5.5	1	0.905328

Player	VORP	BPM	PTS	PER	USG	WS	AST	MVP	HighScore
James Harden\hardeja01	9.3	11.0	36.1	30.6	40.5	15.2	7.5	0	0.497802
Kevin Durant\duranke01	8.9	9.3	28.1	28.3	29.8	18.9	4.6	0	0.245430
LeBron James\jamesle01	7.6	10.9	27.1	30.7	32.0	14.5	6.2	1	0.226261
Stephen Curry\curryst01	7.9	9.9	23.8	28.0	28.9	15.7	7.7	1	-0.015658
James Harden\hardeja01	8.1	8.8	27.4	26.7	31.3	16.4	7.0	0	-0.052778
Giannis Antetokounmpo\antetgi01	7.4	10.4	27.7	30.9	32.3	14.4	5.9	1	-0.404592
Russell Westbrook\westbru01	9.3	11.1	31.6	30.6	41.7	13.1	10.4	1	-0.450461
James Harden\hardeja01	8.0	8.7	29.1	27.4	34.2	15.0	11.2	0	-0.455900

Now we find that Model 2 accurately predicts 11/15 MVPs for the past 15 seasons. That's a big improvement over Model 1, a nearly 30% jump in accuracy. Because of these results, we will stick with Model 2, and use it to declare who should be the winner of the MVP award for the 2020-21 NBA regular season.

Additionally, below you will find a visual representation of Model 2's validity test that was just run above.

Visual Representation of Model 2's Validity Results

As you can see, 11 of the 15 bars have been labeled with the suffix "MVP" in their label. These represent the 11 out of the 15 players that our Model 2 accurately predicted as MVP winners.

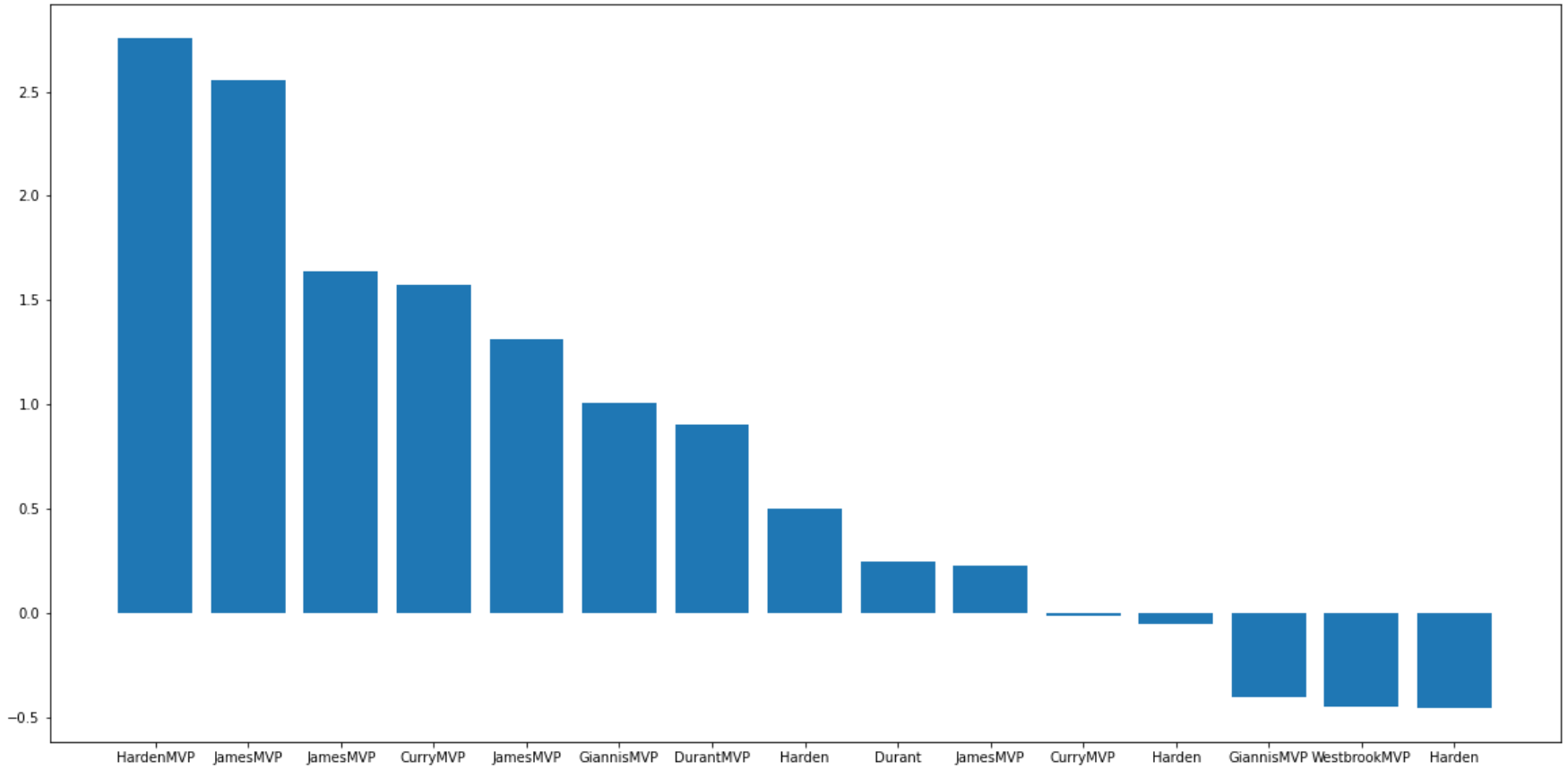
In [40]:
```
# Make the dataset:
height = [2.757508, 2.557464, 1.637148, 1.574116, 1.310512, 1.008587, 0.905328, 0.497802, 0.245430, 0.226261, -
bars = ('HardenMVP', 'JamesMVP', 'JamesMVP', 'CurryMVP', 'JamesMVP', 'GiannisMVP', 'DurantMVP','Harden','Durant
y_pos = np.arange(len(bars))


plt.figure(figsize=(20,10))
# Create bars
plt.bar(y_pos, height)

# Create names on the x-axis
plt.xticks(y_pos, bars)

# Show graphic
plt.show()
```

```
#This visual shows model 2 with the players and High Score.
```



Conclusion

Our model correctly predicts the Most Valuable Player of the National Basketball league in 11 out of the past 15 years. The model operating at greater than 70% accuracy allows us to make reasonable predictions about who should be the Most Valuable Player in any given year. Based on this, we predict that the MVP of the 2020-21 NBA regular season will be:

Nikola Jokić

In the future we think our model can effectively be applied to predicting the MVP race. The main concept holding our model back is the idea of a narrative behind a player's season. Our model does not control for narrative and media bias, and it would be very

difficult to measure how much a player is positively, or negatively, covered in the news. In theory, the MVP race is meant not to be influenced by media coverage throughout the season, but to be performance based. Our model provides a purely performance based prediction of the MVP award. The NBA MVP is selected by a vote casted by sportswriters and broadcasters across the U.S. and Canada. Our model would best be applied as a predictor, but if utilized to select the MVP it would inevitably be combined with narrative based opinions and take into account media grabbing events.

Our model would best be implemented as a compliment to the current MVP vote, whether bolstering support for the most statistically worthy candidates or adding a score of "votes" to an indiviual's tally.

References

Freire, D. (2020, March 24). Predicting 2020–21 NBA's most Valuable player using machine learning. Retrieved April 26, 2021, from https://towardsdatascience.com/predicting-2020-21-nbas-most-valuable-player-using-machine-learning-24aaa869a740

Predicting the 2020 MVP with linear models. (2020, January 17). Retrieved April 27, 2021, from https://dribbleanalytics.blog/2020/01/2020-mvp-mid-season/

Sports Reference LLC, Basketball-Reference.com - Basketball Statistics and History. "*NBA Player Data, Per Game and Advanced*", www.basketball-reference.com/.