Predicting Win Shares in the NBA

By Anthony Le

Purpose

-Predicting the win shares statistic based on player’s stats

- Find which stats are important for win shares

-Big fan of the NBA

-WIll help appreciate the game more.

-Find which model will be the best to predict the win shares

-Can we predict individual win shares of NBA players using other basketball metrics?

Win Shares

- player statistic which attempts to divvy up credit for team success to the individuals on the team
- An advanced statistic to see if the player is helping their team win.
- Add Offensive Win Shares and Defensive Win Shares together to get the total Win Shares.
- The formula for the WIn Shares are from this source. (https://www.basketball-reference.com/about/ws.html)
- The data collected for this analysis are not related to the formula.
- (PP-0.92*LPPP*(FGA+0.44*FTA+TO))/(0.32*LPPG*(TP/LP))+(MP/TMP*TDP*(1.08*LPPP-DRtg/100)/(0.32*LPPG*(TP/LP)

NBA Data

- Collected from the Basketball-Reference.com (https://www.basketball-reference.com/leagues/NBA_2019_advanced.html)
- Data collected website
- 2018-2019 NBA Season
- 708 players data collected (couple players played for multiple seasons)
- Used the Advanced Stats

Data Attributes

- 20 continuous variables
  - Age- Age; player age on February 1 of the given season.
  - G-Number of Games played
  - MP- Minutes Played
  - PER- Player Efficiency Rating
  - Ts%- True Shooting Percentage
  - 3PAr- 3-Point Field Goal Attempts Rate
  - FTr- Free Throw Attempts Rate
  - ORB%- Offensive Rebound Percentage
  - DRB%- Defensive Rebound Percentage
  - TRB%- Total Rebound Percentage
  - AST%- Assist Percentage
  - STL%- Steal Percentage

Data Attributes(Cont.)

- 20 continuous variables
  - BLK%- Block Percentage
  - TOV%- Turnover Percentage
  - USG%- Usage Percentage
  - OBPM- Offensive Box Plus/Minus
  - DBPM- Defensive Box Plus/ Minus
  - BPM- Box Plus/ Minus
  - VORP- Value over Replacement Player

Source-(https://www.basketball-reference.com/about/glossary.html)

Data Exploration

- Check to see any nulls in the dataset at all

```
nba_data.isnull().sum()*100/nba_data.isnull().count()
```

```
Player    0.000000
Age       0.000000
G         0.000000
MP        0.000000
PER       0.000000
TS%       0.847458
3PAr      0.847458
FTr       0.847458
ORB%      0.000000
DRB%      0.000000
TRB%      0.000000
AST%      0.000000
STL%      0.000000
BLK%      0.000000
TOV%      0.847458
USG%      0.000000
WS        0.000000
OBPM      0.000000
DBPM      0.000000
BPM       0.000000
VORP      0.000000
```

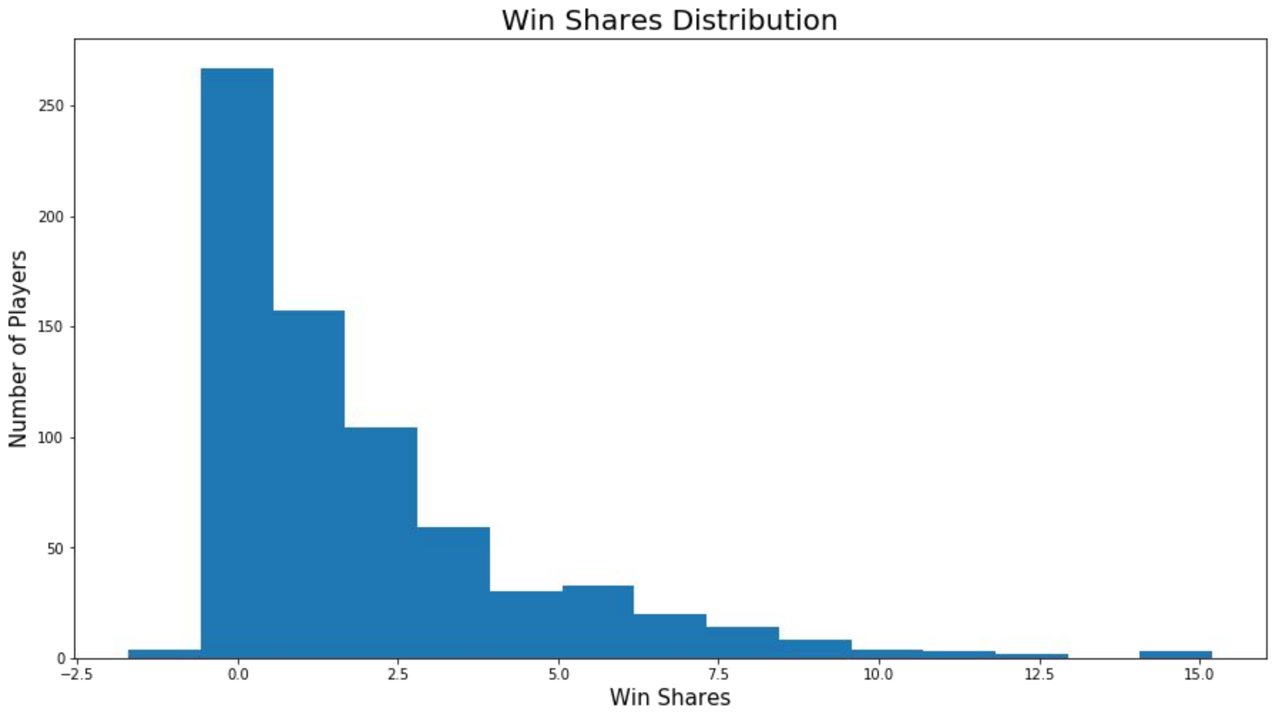
Null

- Replace the null with zero since the null is very low

```
nba_data["TS%"].fillna(0, inplace =True)
nba_data["3PAr"].fillna(0, inplace =True)
nba_data["FTr"].fillna(0, inplace =True)
nba_data["TOV%"].fillna(0, inplace =True)
```

Data Exploration

- Need to see what the Win Shares Distribution looks like.

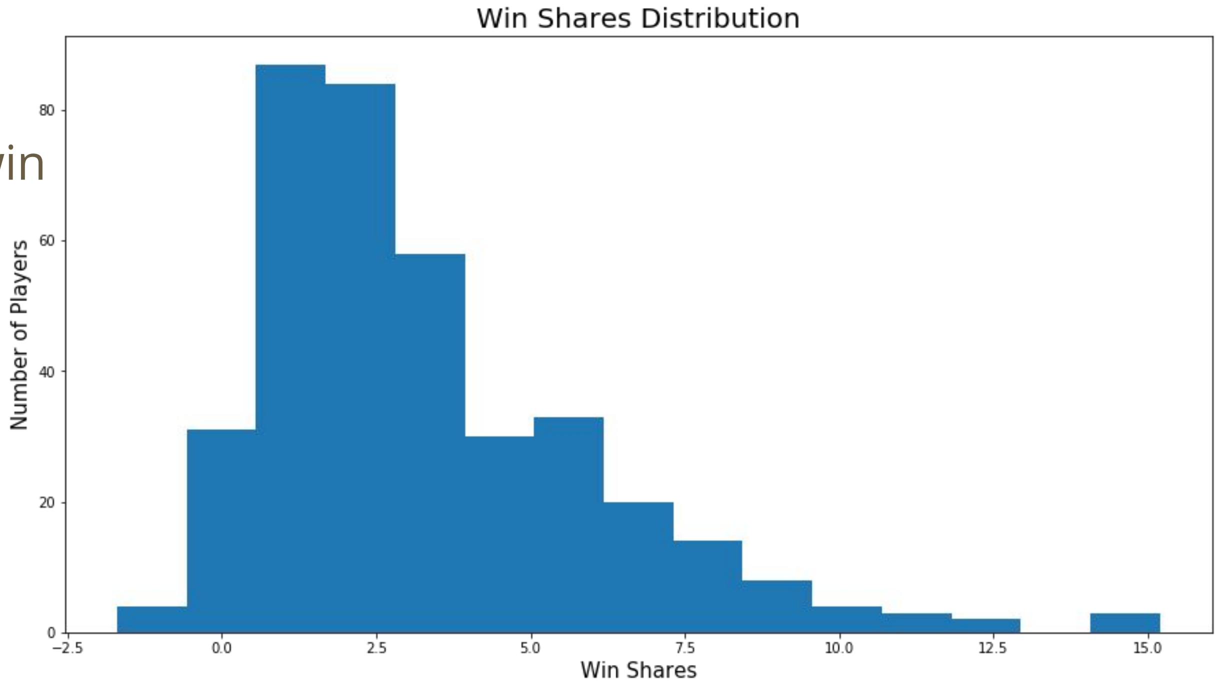


Data Transformation

- Decide to filter out players that barely played
  - 41 games or more
- There’s ton of player with 0 win shares.
- Only 15 players per team
  - Injuries, trades, etc. affects the data set

Data Transformation

- 381 players
- Less players with 0 win shares
- More normalized compared to the last distribution



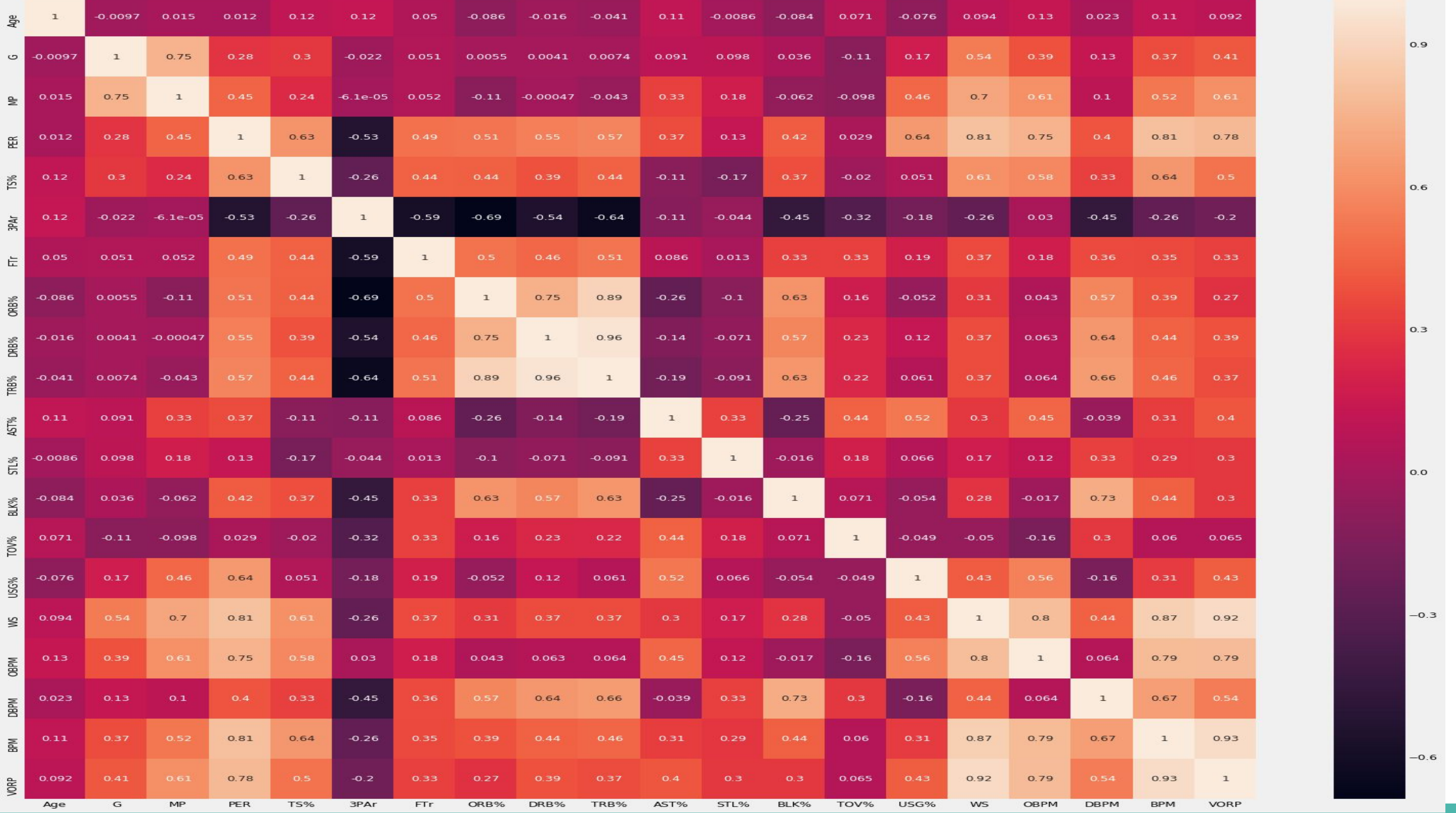
Correlation

- Finding which variables correlated well with WIn Shares.
- Use correlation function and heatmaps to determine correlation.

Correlation-2

- Use variables that correlation that is .5 or higher

	index	WS
15	WS	1.000000
19	VORP	0.924396
18	BPM	0.868633
3	PER	0.813769
16	OBPM	0.795985
2	MP	0.698498
4	TS%	0.612444
1	G	0.537833
17	DBPM	0.441935
14	USG%	0.425792
9	TRB%	0.373042
8	DRB%	0.367111
6	FTr	0.365449
7	ORB%	0.305793
10	AST%	0.298471
12	BLK%	0.276973
11	STL%	0.172317
0	Age	0.094430
13	TOV%	-0.049696
5	3PAr	-0.255229



	Age	G	MP	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	WS	OBPM	DBPM	BPM	VORP
Age	1	-0.0097	0.015	0.012	0.12	0.12	0.05	-0.086	-0.016	-0.041	0.11	-0.0086	-0.084	0.071	-0.076	0.094	0.13	0.023	0.11	0.092
G	-0.0097	1	0.75	0.28	0.3	-0.022	0.051	0.0055	0.0041	0.0074	0.091	0.098	0.036	-0.11	0.17	0.54	0.39	0.13	0.37	0.41
MP	0.015	0.75	1	0.45	0.24	-6.1e-05	0.052	-0.11	-0.00047	-0.043	0.33	0.18	-0.062	-0.098	0.46	0.7	0.61	0.1	0.52	0.61
PER	0.012	0.28	0.45	1	0.63	-0.53	0.49	0.51	0.55	0.57	0.37	0.13	0.42	0.029	0.64	0.81	0.75	0.4	0.81	0.78
TS%	0.12	0.3	0.24	0.63	1	-0.26	0.44	0.44	0.39	0.44	-0.11	-0.17	0.37	-0.02	0.051	0.61	0.58	0.33	0.64	0.5
3PAr	0.12	-0.022	-6.1e-05	-0.53	-0.26	1	-0.59	-0.69	-0.54	-0.64	-0.11	-0.044	-0.45	-0.32	-0.18	-0.26	0.03	-0.45	-0.26	-0.2
FTr	0.05	0.051	0.052	0.49	0.44	-0.59	1	0.5	0.46	0.51	0.086	0.013	0.33	0.33	0.19	0.37	0.18	0.36	0.35	0.33
ORB%	-0.086	0.0055	-0.11	0.51	0.44	-0.69	0.5	1	0.75	0.89	-0.26	-0.1	0.63	0.16	-0.052	0.31	0.043	0.57	0.39	0.27
DRB%	-0.016	0.0041	-0.00047	0.55	0.39	-0.54	0.46	0.75	1	0.96	-0.14	-0.071	0.57	0.23	0.12	0.37	0.063	0.64	0.44	0.39
TRB%	-0.041	0.0074	-0.043	0.57	0.44	-0.64	0.51	0.89	0.96	1	-0.19	-0.091	0.63	0.22	0.061	0.37	0.064	0.66	0.46	0.37
AST%	0.11	0.091	0.33	0.37	-0.11	-0.11	0.086	-0.26	-0.14	-0.19	1	0.33	-0.25	0.44	0.52	0.3	0.45	-0.039	0.31	0.4
STL%	-0.0086	0.098	0.18	0.13	-0.17	-0.044	0.013	-0.1	-0.071	-0.091	0.33	1	-0.016	0.18	0.066	0.17	0.12	0.33	0.29	0.3
BLK%	-0.084	0.036	-0.062	0.42	0.37	-0.45	0.33	0.63	0.57	0.63	-0.25	-0.016	1	0.071	-0.054	0.28	-0.017	0.73	0.44	0.3
TOV%	0.071	-0.11	-0.098	0.029	-0.02	-0.32	0.33	0.16	0.23	0.22	0.44	0.18	0.071	1	-0.049	-0.05	-0.16	0.3	0.06	0.065
USG%	-0.076	0.17	0.46	0.64	0.051	-0.18	0.19	-0.052	0.12	0.061	0.52	0.066	-0.054	-0.049	1	0.43	0.56	-0.16	0.31	0.43
WS	0.094	0.54	0.7	0.81	0.61	-0.26	0.37	0.31	0.37	0.37	0.3	0.17	0.28	-0.05	0.43	1	0.8	0.44	0.87	0.92
OBPM	0.13	0.39	0.61	0.75	0.58	0.03	0.18	0.043	0.063	0.064	0.45	0.12	-0.017	-0.16	0.56	0.8	1	0.064	0.79	0.79
DBPM	0.023	0.13	0.1	0.4	0.33	-0.45	0.36	0.57	0.64	0.66	-0.039	0.33	0.73	0.3	-0.16	0.44	0.064	1	0.67	0.54
BPM	0.11	0.37	0.52	0.81	0.64	-0.26	0.35	0.39	0.44	0.46	0.31	0.29	0.44	0.06	0.31	0.87	0.79	0.67	1	0.93
VORP	0.092	0.41	0.61	0.78	0.5	-0.2	0.33	0.27	0.39	0.37	0.4	0.3	0.3	0.065	0.43	0.92	0.79	0.54	0.93	1

Multicollinearity

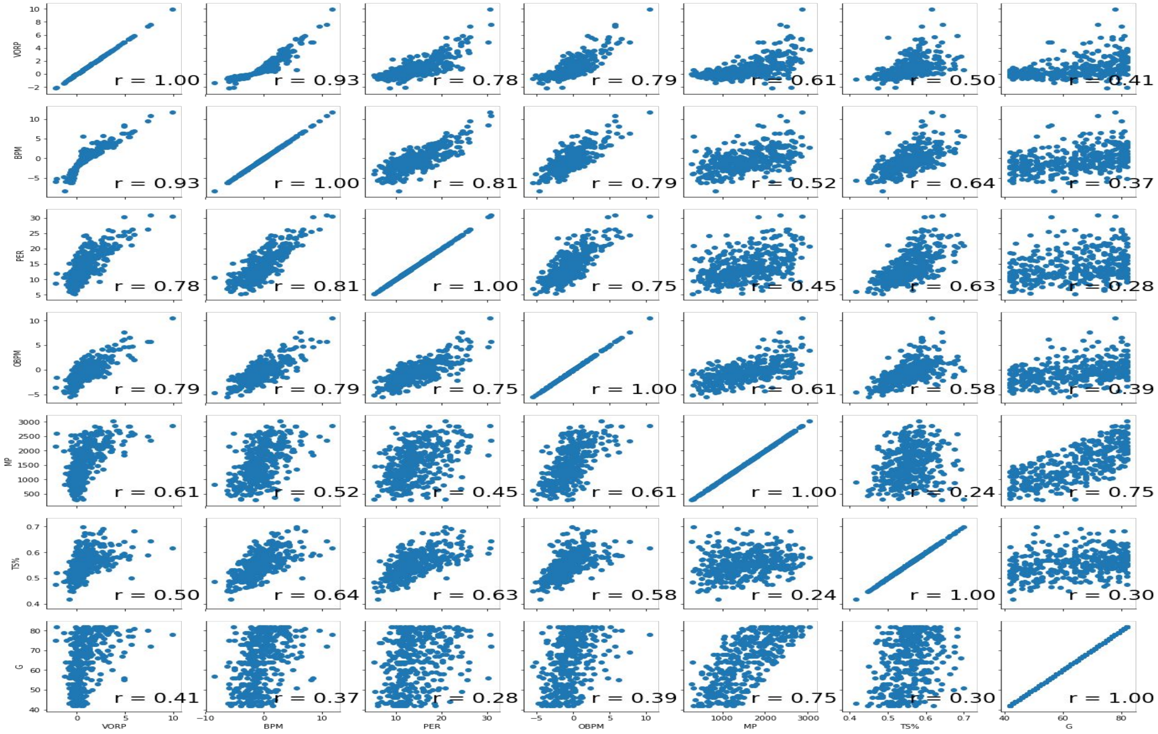
- Need to check that other variables correlated with each too closely.
- Won’t skew the model
- Use graphs and table to check

Multicollinearity

- VORP and BPM have a correlation of .93
- BPM would be dropped because VORP used BPM in their formula

	VORP	BPM	PER	OBPM	MP	TS%	G
VORP	1.000000	0.929908	0.779450	0.794319	0.606268	0.504991	0.408920
BPM	0.929908	1.000000	0.806153	0.785744	0.521236	0.637544	0.371151
PER	0.779450	0.806153	1.000000	0.749606	0.448797	0.625866	0.281484
OBPM	0.794319	0.785744	0.749606	1.000000	0.612428	0.581164	0.387401
MP	0.606268	0.521236	0.448797	0.612428	1.000000	0.235015	0.749390
TS%	0.504991	0.637544	0.625866	0.581164	0.235015	1.000000	0.296773
G	0.408920	0.371151	0.281484	0.387401	0.749390	0.296773	1.000000

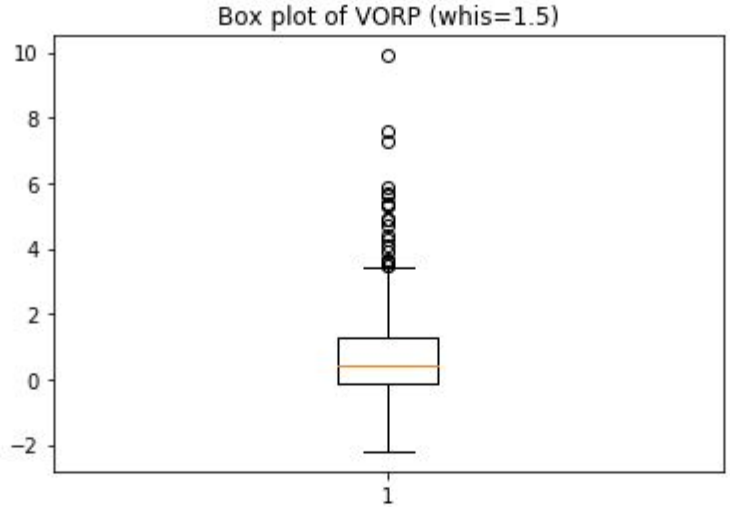
Multicollinearity

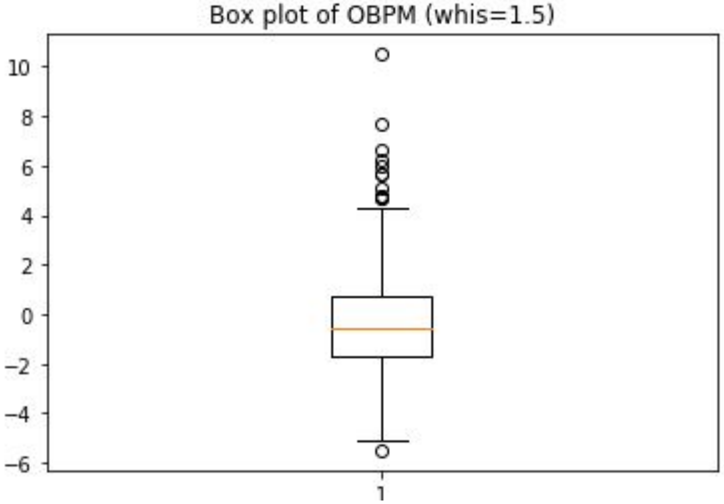


Chosen Features

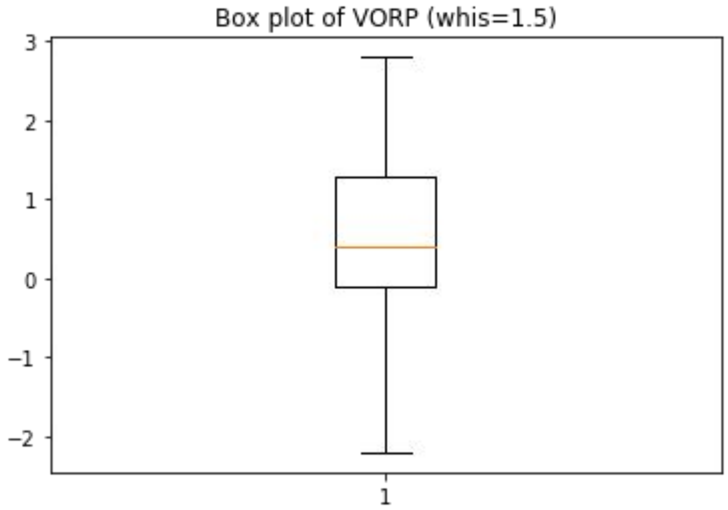
- VORP-Value over Replacement Player
- PER-Player Efficiency Rating
- OBPM-Offensive Box Plus/Minus
- MP- Minutes played
- TS%- True Shooting Percentage
- G- Games Played
- 20% tested and 80% trained

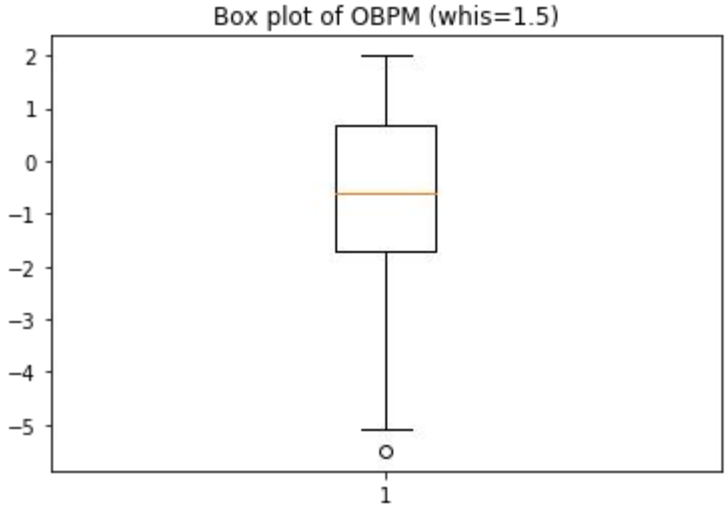
Check For Outliers





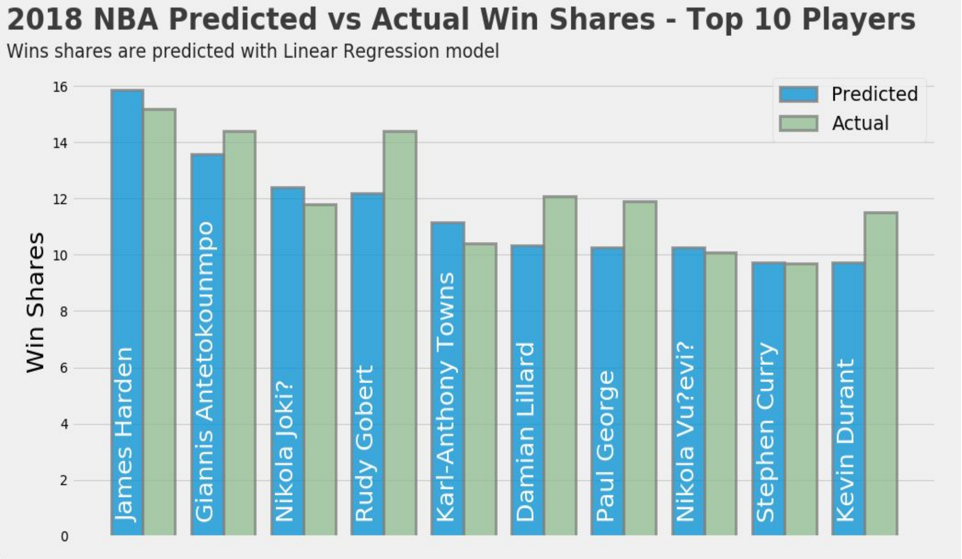
Check For Outliers

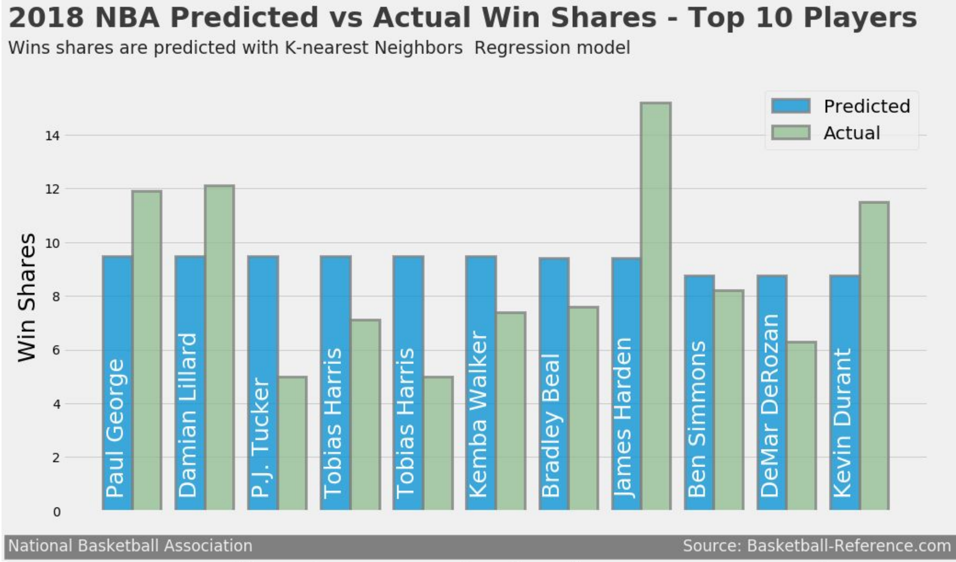


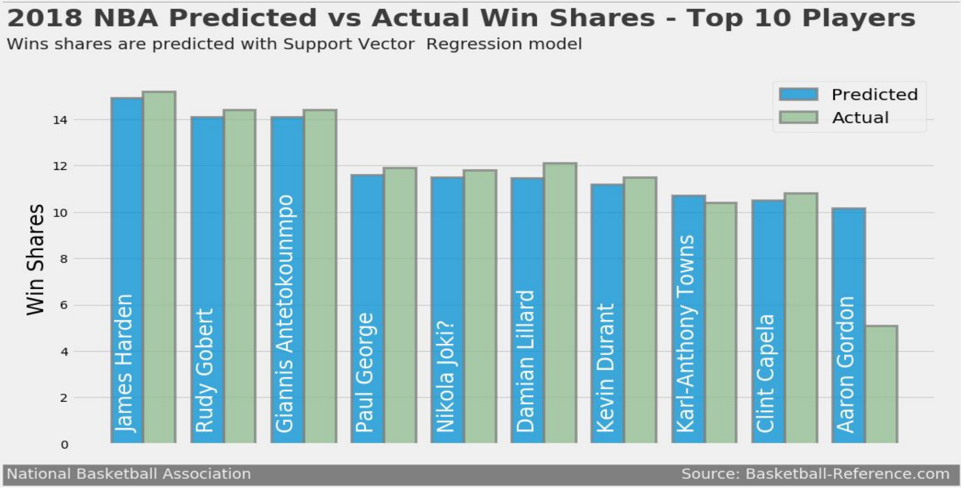


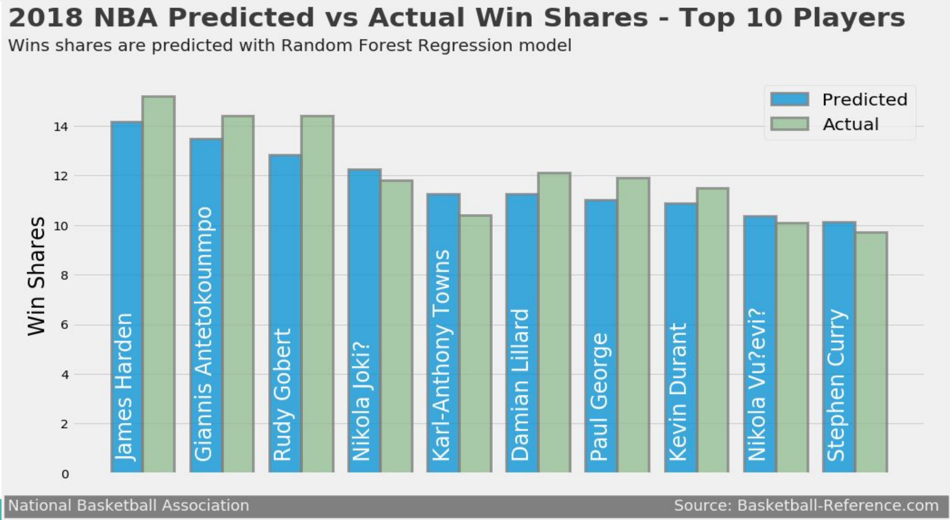
Models

- Linear Regression
- Support Vector Regression
- K-Nearest Neighbors Regression
- Random Forest Regression


2018 NBA Predicted vs Actual Win Shares - Top 10 Players
Wins shares are predicted with Linear Regression model
Predicted
Actual
Win Shares
0
2
4
6
8
10
12
14
16
James Harden
Giannis Antetokounmpo
Nikola Joki?
Rudy Gobert
Karl-Anthony Towns
Damian Lillard
Paul George
Nikola Vu?evi?
Stephen Curry
Kevin Durant


2018 NBA Predicted vs Actual Win Shares - Top 10 Players
Wins shares are predicted with K-nearest Neighbors Regression model
Predicted
Actual
Win Shares
0
2
4
6
8
10
12
14
Paul George
Damian Lillard
P.J. Tucker
Tobias Harris
Tobias Harris
Kemba Walker
Bradley Beal
James Harden
Ben Simmons
DeMar DeRozan
Kevin Durant
National Basketball Association
Source: Basketball-Reference.com


2018 NBA Predicted vs Actual Win Shares - Top 10 Players
Wins shares are predicted with Support Vector Regression model
Predicted
Actual
Win Shares
0
2
4
6
8
10
12
14
James Harden
Rudy Gobert
Giannis Antetokounmpo
Paul George
Nikola Joki?
Damian Lillard
Kevin Durant
Karl-Anthony Towns
Clint Capela
Aaron Gordon
National Basketball Association
Source: Basketball-Reference.com


2018 NBA Predicted vs Actual Win Shares - Top 10 Players
Wins shares are predicted with Random Forest Regression model
Predicted
Actual
Win Shares
0
2
4
6
8
10
12
14
James Harden
Giannis Antetokounmpo
Rudy Gobert
Nikola Joki?
Karl-Anthony Towns
Damian Lillard
Paul George
Kevin Durant
Nikola Vu?evi?
Stephen Curry
National Basketball Association
Source: Basketball-Reference.com

Performance Evaluation

Model	Mean Squared Error	Mean Absolute Error	Variance Score
Linear	0.458	0.448	0.905
Support Vector	2.834	1.282	0.415
k-Nearest Neighbors	3.560	1.332	0.265
Random Forest	0.323	0.440	0.933

Best Model

- Random Forest proven to be the best model
- Predict the top ten players accurately
- Has the lowest value of Mean Squared Error and Mean Absolute Error
- Has the highest value of Variance score.

Random Forest

- Need to see the Random Forest was overfitted or not
- To see there was a good generalization or not.
- Going to test 10%, 50%, 90%

Random Forest

Test	Mean Squared Error	Mean Absolute Error	Variance Score
10%	0.310	0.467	0.942
20%	0.323	.440	.933
50%	0.647	0.557	.0908
90%	1.000	0.705	0.866

Random Forest

- At 10% seem to be the closest for performance for 20% .
- The values of the mean squared error and mean absolute error when the percentage of the data set gets tested
- Except for Mean Absolute value for 20% because it has the lowest value.
- The variance score gets lower when the percentage of the dataset is increased

Conclusion

- Random Forest best model for this dataset based on performance
- VORP has the most correlation
- Tend to favor offensive stats
- Maybe use log transformation instead of Winsorization
- Not the best stat to judge individuals' performance
  - Team fit and player personnel
  - Team success
- Need more datasets
  - Used past seasons
  - Used more features