Predicting the best defensive player in the NBA

Jaša Kšela, Faculty of Computer and Information Science Ljubljana, ksela.jasa@gmail.com

Abstract—The goal of this paper was to predict the winner of award for the best defensive player of the year (dpoy.) in American national basketball association (NBA). Firstly we created the data set with the web scraper. Then we remove all unnecessary attributes to simplify and improve our prediction models. We create visual presentation of our data with the help of Python library Matplotlib. At the end we created three different prediction models and compare their success with k-fold cross validation. Results of all methods were almost the same. From the results we can conclude that the prediction of award winner from the data is very hard. Main reason for that is the lack of good statistical metrics for determining quality on the defensive end of the floor.

Index Terms—Machine learning, basketball, web scraping, dpoy.

1 Introduction

THIS paper is about finding the best defensive player of the season in American national basketball association (NBA). The award is given after the end of the regular season each year. The goal of this paper was to predict the winner. Some of the winners were very obvious, but sometimes the winner was hard to predict. First we created database with the web scraper, created with Python. Library Beautifulsoup was used to create web scraper. Since the 1980–81 season, the award is decided by a panel of sportswriters and broadcasters throughout the United States and Canada. After the creation of database we visually present the data with different graphs. Before training our model we normalize all statistical attribute grouped by year attribute. This makes sense because award is given every year and it is not dependent on previous years. We used 5 fold cross validation to evaluate our models and compare them with each other. We use two different model evaluation metrics to rate every one of them.

December 24, 2020

2 Related work

In the past there has been work done on similar problems. A lot of papers were done about predicting the winner of a single game. For example Renato Torres wrote a paper called *Prediction of NBA games based on Machine Learning Methods* (Torres, 2013). He achived a result of 66% accuracy with the linear regression model.

Many papers were done about finding the most valuable player(MVP) in the NBA. That is very similar to our problem. This award is also decided by a panel of sportswriters and broadcasters. In paper *Predict NBA Regular Season MVP Winner* (Chen, 2017) author achived 47% accuracy of predicting the MVP.

3 Predicting the best defensive player

3.1 Database

In this project we used data from the basketball-reference website. The Sports Reference sites are publicly available websites for multiple sports statistics. We used NBA awards voting data from 1980 to 2020. Database is assembled from all candidates that got at least one vote for

the best defensive player in the history. There are 422 different entities. In the figure 1 we saw the table for one of the years.

Defensive Player of the Year Share & more ▾ Glossary

				Voting					Per Game						Shooting			Advanced	
Rank	Player	Age	Tm	First	Pts Won	Pts Max	Share	G	MP	PTS	TRB	AST	STL	BLK	FG%	3P%	FT%	WS	WS/48
1	Sidney Moncrief	25	MIL	14.0	14.0	75	0.187	76	35.7	22.5	5.8	3.9	1.5	0.3	.524	.100	.826	13.2	.233
2	Tree Rollins	27	ATL	10.0	10.0	75	0.133	80	30.9	7.8	9.3	0.9	0.6	4.3	.510	.000	.726	7.3	.142
3T	Larry Bird	26	BOS	6.0	6.0	75	0.080	79	37.7	23.6	11.0	5.8	1.9	0.9	.504	.286	.840	14.0	.225
3T	Maurice Cheeks	26	PHI	6.0	6.0	75	0.080	79	31.2	12.5	2.6	6.9	2.3	0.4	.542	.167	.754	9.4	.183
3T	Michael Cooper	26	LAL	6.0	6.0	75	0.080	82	26.2	7.8	3.3	3.8	1.4	0.6	.535	.238	.785	5.2	.116
3T	Bobby Jones	31	PHI	6.0	6.0	75	0.080	74	23.6	9.0	4.6	1.9	1.1	1.2	.543	.000	.793	6.4	.175

Fig. 1. Table from website

Web scraper was used to create database. We used Python and library called Beautifulsoup to create it. We loop through url addresses for each year and save candidates to database. We also wanted to get some advanced statistic metrics for each player. Because of that we use Selenium library to dynamically load those stats. We saved selected statistical attributes:

- Points, assists, rebounds, steals and blocks per game
- Field goal %
- Team wins
- Defensive win shares
- Box plus/minus
- Defensive box plus/minus
- Team defensive rankings
- Value over replacement player.

Database was saved as Comma-separated values (.CSV) file.

	leto	rank	igralec	starost	ekipa	first	pts_won	pts_max	share	g	...	ast	stl	blk	fg%	d
0	1983	1	Sidney Moncrief	25	MIL	14.0	14.0	75	0.187	76	...	3.9	1.5	0.3	0.524	
1	1983	2	Tree Rollins	27	ATL	10.0	10.0	75	0.133	80	...	0.9	0.6	4.3	0.510	
2	1983	3	Larry Bird	26	BOS	6.0	6.0	75	0.080	79	...	5.8	1.9	0.9	0.504	
3	1983	4	Maurice Cheeks	26	PHI	6.0	6.0	75	0.080	79	...	6.9	2.3	0.4	0.542	
4	1983	5	Michael Cooper	26	LAL	6.0	6.0	75	0.080	82	...	3.8	1.4	0.6	0.535	

Fig. 2. Few players from database

3.2 Defining the problem

Our goal was to create three different models to predict the winner of defensive player of the year award. Each sportswriter cast a vote for first to fifth place selection. Each first-place vote is worth 10 points, each second-place vote is worth seven, each third-place vote is worth five, fourth-place is worth three and fifth-place is worth one. Player with the most points total wins the award [3].

In our prediction model we want to predict the share of the votes that each player would get. Based on that we can predict the winner. We created correlation matrix to determine most important attributes for our model.

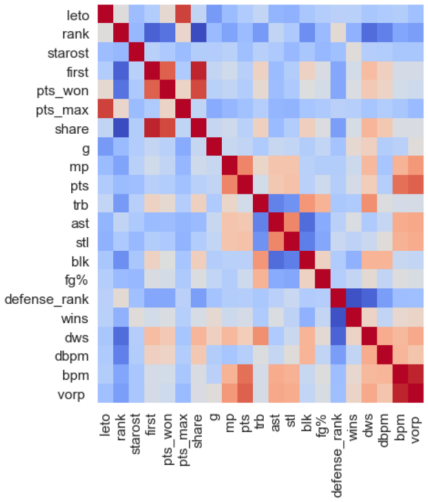

Fig. 3. Correlation matrix

From our model we removed all attributes that have correlation to share attribute lower than 0.15. As expected most important attributes for our model are advanced defensive metrics. Five statistical attributes with the highest correlations are:

- Defensive win shares - 0.48
- Defensive box plus/minus - 0.38
- Blocks per game - 0.34
- Rebounds per game - 0.33
- Team defense ranking - 0.27

From the model we also removed all attributes that directly tell us the result of voting.

We wanted to predict the share attribute which is a numerical value. From this we conclude that our problem falls under regression problems. Attribute share is our dependant continuous variable, Other attributes in our model are independent variables.

3.3 Visualization of our data

After the creation of database we also tried to visualize our data. We grouped data by the win-

ners of the award to visualize the correlation between the winners. From the figure 4 we see that high defensive win shares is common to all the winners. From the previous chapter we also see that this attribute has the highest correlation value to our dependant value (share). It is also very common that the winner has a lot of rebounds and blocks per game.

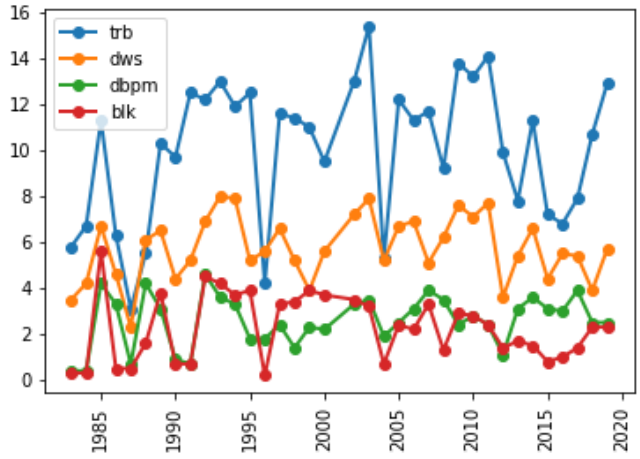

Fig. 4. Stats grouped by the winners

From the figure 4 we can assume that the winners are mostly tall players. Rebounds and blocks are statistical attributes very common for tall players (centers, power forwards). Because of that the height of players was also included in our data. In next figure we show the weight and the height of all candidates depending on the year.

Height of average NBA player is 198 cm. From our data we can see that average height of award winner is 207 cm and average height of all award candidates is 202 cm.

If we look at the weight of players, the narrative is very similar. Weight of average NBA player is 98 cm. The average weight of award winner is 106.1 kg and average weight of all award candidates is 100.5 kg. More statistical indicators about physical attributes of award winners we can see in figure 5.

	age	weight	height
mean	27.5	106.1	207.1
standard deviation	2.7	12.8	9.2
min	23.0	77.0	190.0
50%	28.0	109.0	208.0
max	32.0	125.0	224.0

Fig. 5. Averages of winners physical attributes

In the figure below you can see averages of physical attributes of all candidates grouped by year. From it we can conclude that trends are not changing a lot through the history.

Fig. 6. Averages of candidates physical attributes grouped by year

In the next chapter we are going to explain three different regression methods that were used in our models.

3.4 Prediction models

Our target attribute is continuous value. Because of that we can mark our problem as regression problem. We used three popular regression machine learning methods in our model:

- Linear regression,
- K-nearest neighbors regression,
- Artificial neural networks.

Before training our model we normalize all statistical attribute grouped by year attribute. This makes sense because award is given every year and it is not dependent on previous years. We used 5 fold cross validation to evaluate our models and compare them with each other. We use two different model evaluation metrics to rate every one of them:

- mean squared error,
- mean absolute error.

All negative results were set to zero, because negative values are not possible for dependant variable.

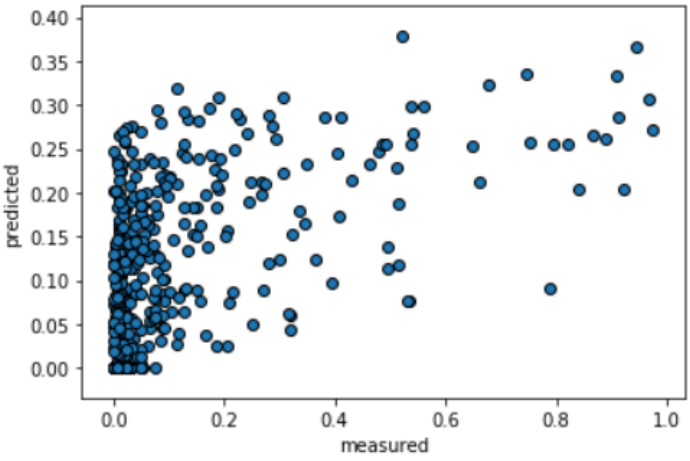

Fig. 8. Measured values on x-axis and predicted values on y-axis for linear regression

3.4.1 Linear regression

Linear regression assumes all attributes to be continuous. That criteria fits our model where all attributes are numerical. This function assumes the existence of linear relationship between independent and dependent variables. The goal of algorithm is to find the coefficients of the linear function in order to minimize the sum of squared errors of regressional predictions evaluated on learning examples. The result of algorithm is a hyper plane that for each entity based on independent variables determines the value of dependent variable [4].

In the figure below we can see the result of our linear regression model. As we see it is quite accurate.

METHOD	MAE	MSE
LINEAR REGRESSION	-0.113006	-0.027464

Fig. 7. Evaluation metrics for linear regression

But the model has a problem with predicting high values. We can see that from the figure 8. The reason for that is that stats of all candidates is very similar. Because of that is very hard to predict high values.

3.4.2 K-nearest neighbors regression

One of the oldest and simplest regression methods is k-nearest neighbors regression. Nearest neighbors algorithm adapted for regression is also called locally weighted regression. In this method, the predicted values for interesting variables are obtained as weighted averages of the values of neighboring observations. Most commonly Euclidean distance metric is used to measure the distance between two instances [5].

Figure 9 shows that the accuracy of K-nearest neighbors regression is almost the same as the accuracy of linear regression.

METHOD	MAE	MSE
KNeighborsRegressor	-0.101121	-0.028802

Fig. 9. Evaluation metrics for K-nearest neighbors regression

Generally the accuracy is better than with linear regression model. However the model is even worse at predicting high values (mean squared error is higher).

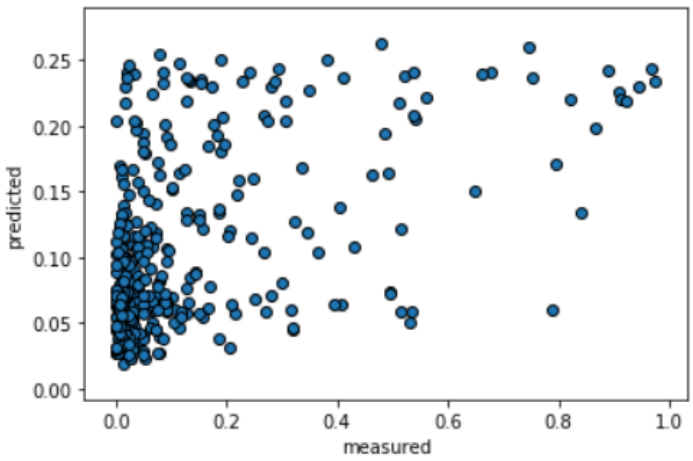

Fig. 10. Measured values on x-axis and predicted values on y-axis for K-nearest neighbors regression

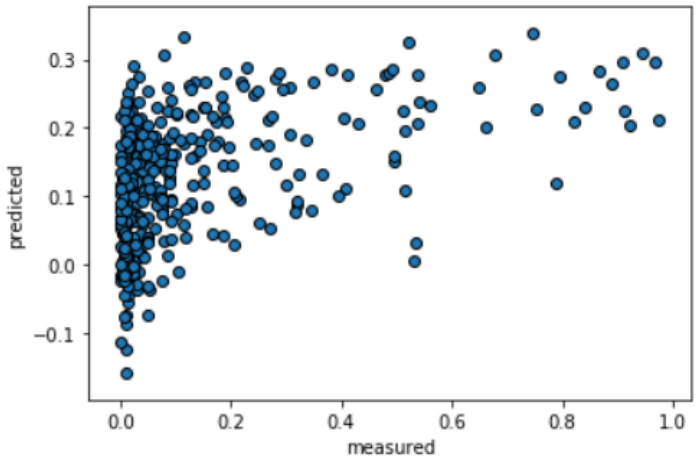

Fig. 12. Measured values on x-axis and predicted values on y-axis for neural networks

3.4.3 *Artificial neural networks*

This model can be used for classification and also for regression. Networks are inspired by the biological neural network that constitute animal brains. Neurons are organized in a layered way. There is input layer, one or more hidden layers and and output layer with a single neuron corresponding to the regressional variable. Goal of the learning algorithm is to determine weight on connections between neurons to match the minimal regressional error [4].

Figure 11 shows that accuracy of neural networks is almost the same as the accuracy of previous two models.

METHOD	MAE	MSE
Neural networks	-0.112542	-0.028618

Fig. 11. Evaluation metrics for neural networks

As we see in the figure below our model is pretty good at recognizing low values but it has problem with high values similar as previous two models.

4 Conclusion

From the results we can see that all methods have similar accuracy. We can conclude that it is very hard to predict the winner of the award with those machine learning methods and with the available data. We tried to get as many defensive statistical metrics as possible, but we can see that most of the candidates have similar values at those attributes. Because of that it is very hard to recognize the winner of the award. For example finding the best offensive player is a lot easier because there are more statistical metrics available for the offensive end of the floor. Our conclusion is that award is not always given to the player with the best statistical metrics, more important is the narrative behind the player. Voters are not always objective. Because of that they sometimes vote for more popular players even if less known players has better statistics. From all the reasons stated above we conclude that it is very hard to predict the winner of the award for the best defensive player of the year.

References

[1] Chen M., *Predict NBA Regular Season MVP Winner.*, 2017.
[2] Torres R., *Prediction of NBA games based on Machine Learning Methods.*, 2013.
[3] Corvo M., *How voting is done for the NBA MVP and its evolution.*, https://clutchpoints.com/how-voting-is-done-for-the-nba-mvp-and-its-evolution/, 2020.
[4] Kononenko I. and Kukar M. *MACHINE LEARNING AND DATA MINING: Introduction to Principles and Algorithms*. Horwood Publishings, 2007.

[5] Nguyen, Morell and De Baets, *Large-scale distance metric learning fork-nearest neighbors regression*, 2016.