Venkat Ramaraju

Topic – Using advanced basketball analytics to predict whether or not a player in the National Basketball Association (NBA) will be selected as an all-star in the current NBA season.

Introduction

For this project, I have developed four supervised learning models that have been taught in class to perform a binary classification on one of my favorite datasets – Advanced NBA analytics. These are second level analytics that have been computed by analytic experts and open-sourced for people to perform analyses on. Advanced analytics are highly valued as they tell a deeper story about a player's performance in the game. This is the reason I chose to use advanced NBA analytics as the dataset on which I would be performing my analysis.

Collection of training data

For this project, I have collected 5 years' worth of advanced analytics from basketballreference.com. This website has been curating basketball statistics of the NBA since the early 1970's and has available a large dataset in CSV formats for its users.

Rk	Player	Pos	Age	Tm	G	MP	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	OWS	DWS	WS	WS/48	OBPM	DBPM	BPM	VORP
1	Precious Achiuwa	PF	21	MIA	53	652	14.5	.557	.000	.515	11.9	19.9	16.0	6.6	1.3	4.2	13.6	19.2	0.3	1.0	1.3	.096	-3.1	-0.7	-3.8	-0.3
2	Jaylen Adams	PG	24	MIL	7	18	-6.5	.125	.250	.000	0.0	16.8	8.7	12.8	0.0	0.0	0.0	18.6	-0.1	0.0	-0.1	-0.246	-14.9	-5.0	-19.8	-0.1
3	Steven Adams	C	27	NOP	52	1455	15.5	.600	.011	.446	15.2	20.2	17.7	9.3	1.6	2.3	17.9	11.9	2.3	1.3	3.5	.116	-0.2	-0.6	-0.8	0.4
4	Bam Adebayo	C	23	MIA	50	1673	22.6	.625	.013	.448	7.9	22.8	15.6	27.0	1.4	3.6	15.0	24.0	4.2	2.6	6.8	.195	2.9	1.8	4.6	2.8
5	LaMarcus Aldridge	C	35	TOT	26	674	15.7	.556	.270	.159	3.0	16.0	9.4	11.0	0.8	3.7	7.9	22.2	0.5	0.7	1.2	.083	-0.2	-0.4	-0.6	0.2
5	LaMarcus Aldridge	C	35	SAS	21	544	15.1	.545	.302	.149	3.2	15.4	9.2	10.2	0.7	2.9	7.0	22.7	0.3	0.5	0.8	.070	-0.2	-1.0	-1.2	0.1
5	LaMarcus Aldridge	C	35	BRK	5	130	18.2	.611	.104	.208	1.8	18.1	10.3	14.2	1.1	7.4	11.8	19.9	0.2	0.2	0.4	.138	-0.2	1.9	1.7	0.1
6	Ty-Shon Alexander	SG	22	PHO	8	23	-3.8	.225	.750	.250	0.0	14.3	7.3	5.1	0.0	0.0	0.0	17.2	-0.1	0.0	-0.1	-0.166	-11.0	-5.0	-16.0	-0.1
7	Nickeil Alexander-Walker	SG	22	NOP	41	851	12.5	.524	.495	.138	1.3	15.0	8.2	13.8	2.3	2.4	12.4	22.8	-0.2	0.8	0.6	.034	-1.6	-0.1	-1.7	0.1
8	Grayson Allen	SG	25	MEM	41	1027	13.5	.599	.676	.215	1.5	12.3	6.8	11.8	1.8	0.6	9.7	16.9	1.4	1.0	2.4	.113	0.1	0.2	0.2	0.6
9	Jarrett Allen	C	22	TOT	46	1327	21.6	.680	.038	.659	11.6	27.2	19.3	8.8	0.8	5.1	13.6	16.9	3.6	1.7	5.3	.193	2.1	0.1	2.2	1.4
9	Jarrett Allen	C	22	BRK	12	320	21.5	.730	.000	.938	14.0	29.1	21.9	8.2	1.1	5.2	19.3	15.5	1.0	0.5	1.4	.217	1.3	0.6	1.9	0.3
9	Jarrett Allen	C	22	CLE	34	1007	21.7	.667	.046	.594	10.9	26.6	18.5	9.0	0.7	5.0	11.9	17.4	2.7	1.2	3.9	.186	2.4	-0.1	2.3	1.1
10	Al-Farouq Aminu	PF	30	TOT	19	384	9.4	.478	.352	.209	5.3	21.8	13.3	10.8	2.2	2.2	21.4	14.2	-0.4	0.5	0.1	.012	-4.9	1.0	-3.9	-0.2
10	Al-Farouq Aminu	PF	30	ORL	17	367	10.0	.482	.348	.191	5.5	21.0	13.0	11.2	2.3	2.3	20.6	14.3	-0.3	0.5	0.2	.020	-4.4	1.2	-3.3	-0.1
10	Al-Farouq Aminu	PF	30	CHI	2	17	-3.6	.347	.500	1.000	0.0	38.8	19.7	0.0	0.0	0.0	41.0	12.3	-0.1	0.0	-0.1	-0.150	-14.6	-2.4	-17.0	-0.1
11	Kyle Anderson	PF	27	MEM	52	1413	17.2	.569	.400	.267	3.0	20.7	11.7	18.1	2.2	2.7	10.7	18.8	2.0	2.1	4.1	.139	1.2	1.7	3.0	1.8
12	Giannis Antetokounmpo	PF	26	MIL	46	1554	28.8	.630	.203	.534	5.6	28.4	17.4	29.1	1.6	3.5	14.1	32.6	5.1	2.7	7.8	.241	6.0	2.6	8.5	4.1
13	Kostas Antetokounmpo	PF	23	LAL	13	48	2.1	.428	.000	1.375	9.5	29.7	19.8	2.7	2.0	7.4	43.8	20.6	-0.3	0.1	-0.1	-0.140	-12.2	1.1	-11.1	-0.1
14	Thanasis Antetokounmpo	SF	28	MIL	43	474	11.2	.550	.197	.320	10.8	13.2	12.1	9.9	1.8	1.9	21.4	15.7	0.1	0.6	0.7	.069	-3.1	-0.3	-3.4	-0.2
15	Carmelo Anthony	PF	36	POR	54	1356	13.9	.530	.394	.207	2.1	12.0	6.9	9.5	1.4	2.3	7.5	23.3	0.9	0.6	1.4	.051	-0.7	-1.8	-2.5	-0.2
16	Cole Anthony	PG	20	ORL	31	816	11.3	.474	.305	.201	3.3	15.2	9.0	23.7	1.1	1.5	13.8	22.6	-0.6	0.6	0.0	.001	-2.4	-1.1	-3.6	-0.3
17	OG Anunoby	SF	23	TOR	37	1221	14.6	.597	.508	.202	4.6	14.6	9.5	9.2	2.3	2.1	12.4	18.5	1.1	1.4	2.5	.099	0.1	0.3	0.4	0.7
18	Ryan Arcidiacono	PG	26	CHI	32	348	8.7	.500	.571	.176	1.3	14.9	8.2	15.8	1.0	0.0	6.7	13.0	0.3	0.2	0.5	.070	-3.4	-0.2	-3.6	-0.1
19	Trevor Ariza	SF	35	MIA	15	385	10.3	.512	.541	.207	2.7	14.0	8.5	9.5	1.4	2.7	6.9	15.2	0.1	0.5	0.5	.068	-2.4	0.0	-2.5	0.0
20	D.J. Augustin	PG	33	TOT	51	981	11.7	.559	.626	.259	2.0	6.8	4.4	22.2	1.3	0.1	14.7	16.1	1.3	0.6	1.9	.092	-0.7	-1.2	-1.9	0.0

Fig 1: 2021 NBA advanced analytics for 20 players

Each year, there are about 700 players in the NBA on various teams. Basketballreference has tracked 20 different advanced statistics which these players are evaluated against.

Hence, 5 years of data * 700 players = 3500 rows

Overall training data size: 3500 rows x 20 columns

Note – I have made a few modifications to the dataset. I have appended a new column called “All Star” which is a boolean of whether or not a player made the all-star in that particular season, given the statistics in the previous columns. This was for training purposes. My modified version of the dataset is also available on my GitHub. However, it is

a derivative of basketballreference.com's datasets, which have been linked in the References section.

During training, I also removed columns that did not play a role in whether a player would make the all-star team or not (position, team, etc.). This provided me with a simplified, scaled version of the dataset that could be used for training, as each number in each row was a relevant statistic towards the binary prediction.

The test data set was the advanced NBA statistics for the current (ongoing) NBA season before the all-star selection. This was done so that there would be an instant verification process with the actual results of the all-star selection vs what my machine learning models would predict.

Models for training

I have implemented most supervised learning classifier models that have been taught in class.

1. K-Nearest Neighbors
2. Logistic Regression
3. Decision Trees
4. Support Vector Machines

In my initial proposal, I had mentioned that I believe Logistic regression would have the best performance, since it is well designed for binary classification via the sigmoid function.

Comparative study

After training each machine learning model, I have obtained the following results

Supervised Learning Model	F1 Accuracy
Support Vector Machines	0.97
K-Nearest Neighbors	0.96
Decision Trees	0.51
Logistic Regression	**0.98**

This has confirmed my initial prediction that **Logistic Regression would be able to provide the best classification accuracy for our dataset.**

	precision	recall	f1-score	support
0.0	0.99	0.98	0.99	515
1.0	0.71	0.89	0.79	28
accuracy			0.98	543
macro avg	0.85	0.94	0.89	543
weighted avg	0.98	0.98	0.98	543

Fig. 2: Classification Accuracy of Logistic Regression

Reasoning: Logistic regression is intended for binary classification problems. There is an inherent reasoning for this – Logistic regression is an improvement on regular linear regression. Instead of using the linear equation ($\beta_0 + \beta_1 x$) directly to create classification boundaries, logistic regression passes this equation to the sigmoid function, which will output a probability between [0,1]. This transformation makes it much easier to create a decision boundary for binary classification, as we are now dealing with probabilities as

opposed to real data. It is, traditionally, much easier to create a decision boundary based on probabilistic outputs.

Reason for lesser accuracies of the other models

- Decision Trees – The biggest pitfall of the decision trees is that our training data is mostly continuous and not multi-class based. Each column is a decimal value, and is different for each row, since these are continuous values that are unlikely to have the same value. Therefore, the decision tree will have a tough time finding examples that can be used to train (for entropy, etc.) Decision trees are traditionally suited for multiclass classification with more balance and less entropy. Additionally, for our dataset, the training examples for each class is not evenly split (Ratio is about 1:30), since only few players should make the all-star team. This is why decision trees have reported an accuracy of only 58%.
- KNN and SVM – Both of these models performed relatively well (>= 95%) but could not match the accuracy of logistic regression since these models are not uniquely suited for binary classification, unlike logistic regression.
    - Imbalance in dataset: Some machine learning models can be affected by the lack of training examples. In our case, only ~3.3% of the training data belongs to the positive class. As mentioned previously, this is due to the fact that only 24 players out of 700+ will make the NBA all-star team in a given year, as it is a prestigious ranking for a player. Only the top NBA players will make the NBA all-star team.

Future Improvements

There are various ways I could improve the project. To begin with, I could develop a convolutional neural network that would simulate the prediction that these supervised learning models are currently doing. I could also improve my training data set to take into account more features – such as amount of team victories, fame level (which is an actual factor in the voting process), etc. Such measures would improve the overall accuracy of the model.

Conclusion

Through this project, I have been able to develop multiple supervise machine learning models that have been taught in class using the Sci-Kit Learn library. All the code I have written, along with the datasets, is available on my GitHub at this link.

References

Training and Testing Datasets: https://www.basketball-reference.com/leagues/NBA_2021.html

NBA Advanced Analytics: https://bleacherreport.com/articles/1813902-advanced-nba-stats-for-dummies-how-to-understand-the-new-hoops-math

SciKit Learn: https://scikit-learn.org/stable/