Predicting NBA Playoffs using Machine Learning

Sean Liu [*]

February 18, 2021

Abstract

This project attempts to predict the NBA playoff bracket using machine learning methods. It will consider one self-constructed model and one machine learning model built from various machine learning algorithms. The project will also determine the most efficient model for predicting NBA results and which way to select data gives an accurate and consistent prediction. Finally, the project will investigate the effect of home and away variables on the teams' performance and the model's accuracy.

1 Introduction

The National Basketball Association (NBA) is considered as the premier basketball league for professional male basketball players in USA. It is made up of 30 teams, split into the Eastern and Western conferences [Aut01].

During the playoff, the top 8 teams from each conference (Eastern and Western) are chosen to compete for the championship. The rankings are decided based on the teams' performances during the regular season. Then, the teams play against each other with the 1st place playing against the 8th place, the 2nd place playing against the 7th place, etc. Each game will be a best-of-seven match, and teams will rotate between home and away.

Like AlphaGo in the Go Contest [Sil02], machine learning is a well-known prediction tool for complex process. The question of this research is, can machine learning be used to predict the NBA playoff bracket? And what is accuracy of such prediction compared with the real results? What machine learning method is the best solution for NBA playoff bracket prediction?

To better analyzing and comparing the performance of the machine learning in predicting the NBA playoff bracket, firstly we create a self-made prediction model based on several key game variables that will impact the game result mostly by our best knowledge about the NBA games. These variables include 1) effective field goal percentage, 2) free throw percentage, 3) turn over percentage, 4) Offensive rebound percentage, 5) Defensive rebound percentage. And We

[*] Advised by: Derek Sorensen

focused on working out the probability of Team A winning against Team B, then applying this to every game in the playoff. Base on the historic game data of 2014 to 2018, the playoff prediction of 2018,2017 and 2016 are carried out and comparison with the real playoff brackets are also presented.

As for the machine learning model for the NBA playoff bracket prediction, here, we are focusing on 5 different machine learning models that have already been implemented in the Python Machine Learning Library (Scikit-learn), i.e., Logistic Regression (LR), Linear Discriminate Analysis (LDA), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Classification and Regression Tree (CART) [Dhi03] - [TA11]. For comparison with the self-made model, the same playoff prediction of 2018, 2017 and 2016 are carried out and comparison among different machine learning models are given accordingly.

2 Result

2.1 Exposition of self-made prediction model

Based on the testing results for our self-made prediction model, we have the following prediction results (Table 1). And the predicted playoff bracket with the original ones are shown in Figure 1 and 2, with the prediction difference highlighted in red color.

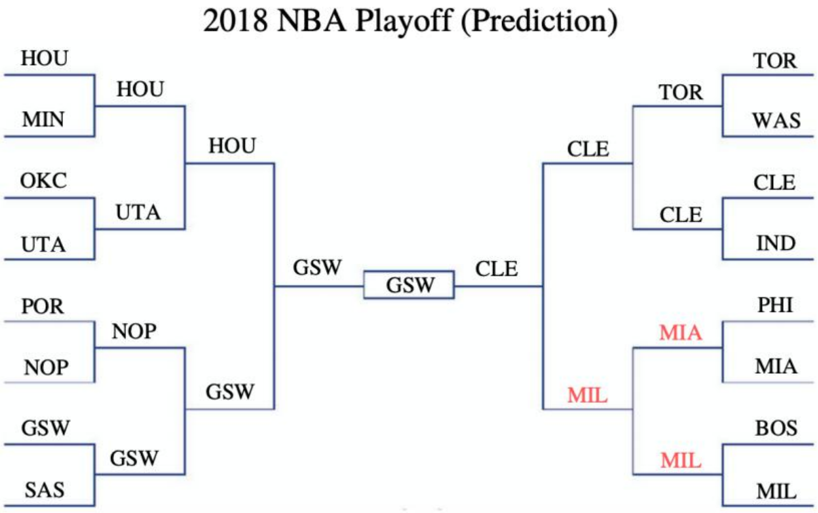


Figure 1: 2018 NBA Playoff (Prediction)

In summary, the self-built prediction model performed best when predicting the 2018 playoff, getting an accuracy of 80%. The second-best prediction was the 2017 prediction, obtaining an accuracy of 66.7%. The worst prediction was for the 2016 bracket, only getting an accuracy of 53.3%. For the 2018 playoff

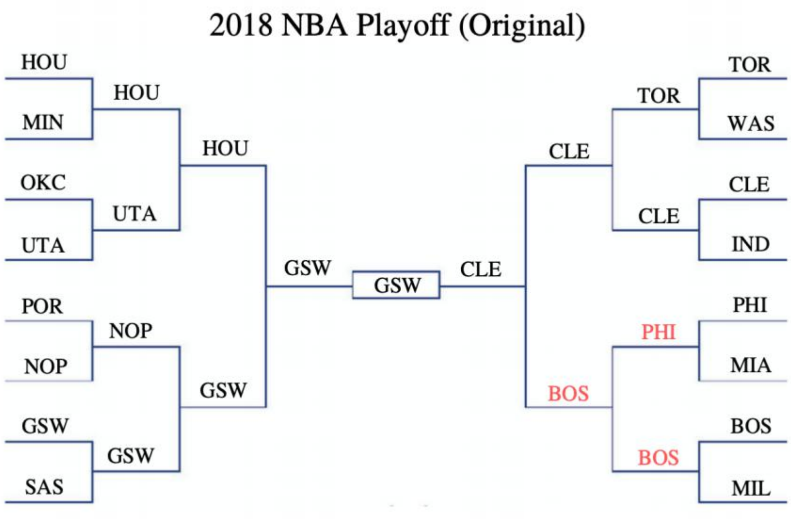
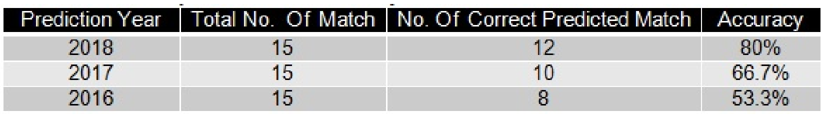


Figure 2: 2018 NBA Playoff (Original)



Prediction Year	Total No. Of Match	No. Of Correct Predicted Match	Accuracy
2018	15	12	80%
2017	15	10	66.7%
2016	15	8	53.3%

Figure 3: Playoff Prediction Accuracy of Self-made Prediction Model

prediction, we used teams' statistics over 4 years from 2014-2018. For the 2017 playoff prediction, we used statistics of teams over a time of 3 years from 2014 - 2017. Finally, for the 2016 playoff prediction, we only used statistics of teams over two years from 2014-2016.

2.2 Exposition of the Machine Learning Models

In order to evaluate the performance of machine learning in NBA playoff bracket prediction, 5 different machine learning models are employed, which have been implemented by Python Machine Learning Library (Scikit-learn), i.e., 1) Logistic Regression (LR), 2) Linear Discriminate Analysis (LDA), 3) Support Vector Machine (SVM), 4) K-Nearest Neighbors (KNN) and 5) Classification and Regression Tree (CART) [Dhi03] - [TA11]. With two different training data selection methods and home/away investigation, the most accurate machine learning model for NBA playoff bracket prediction is finally presented.

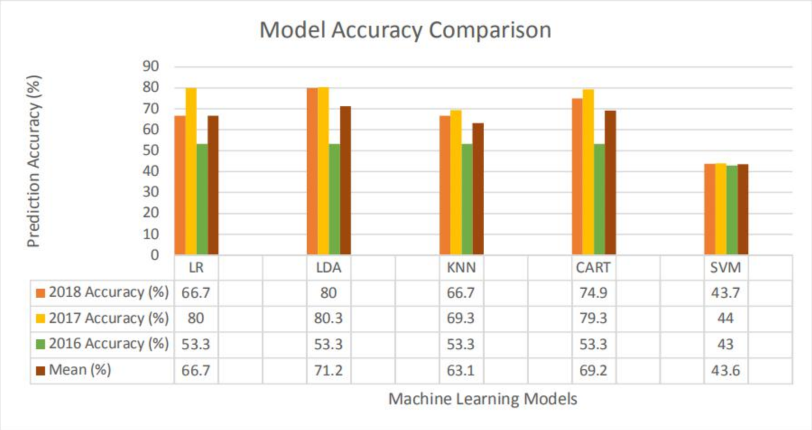
There are two ways in which we chose to select the data to train the algorithm. 1. The first method was to consider each Team's performance when playing against all other teams 2. The second method was to evaluate a team's performance with one other specific Team to determine its win rate against that specific Team

2.2.1 Method 1 – Selecting all data

In this model, we trained the different machine learning algorithms with all the statistics of every Team. Like the self-constructed model, the 2018 prediction used data over four years, the 2017 prediction used data over three years, and the 2016 prediction used data only over two years.

From the above data, algorithms generally performed relatively well and consistent in 2018 and 2017, except for the SVM model and the LR model (Figure 3). The SVM model generally had a low and consistent prediction accuracy in the three years, and the LR did significantly better in 2017 than in 2018.

Overall, LDA had the highest mean accuracy of 71.2%, followed by the CART model, with a mean accuracy of 69.2%. The worst performing model is the SVM algorithm with an accuracy of 0.436 only.



	LR	LDA	KNN	CART	SVM
🟧 2018 Accuracy (%)	66.7	80	66.7	74.9	43.7
🟨 2017 Accuracy (%)	80	80.3	69.3	79.3	44
🟩 2016 Accuracy (%)	53.3	53.3	53.3	53.3	43
🟫 Mean (%)	66.7	71.2	63.1	69.2	43.6

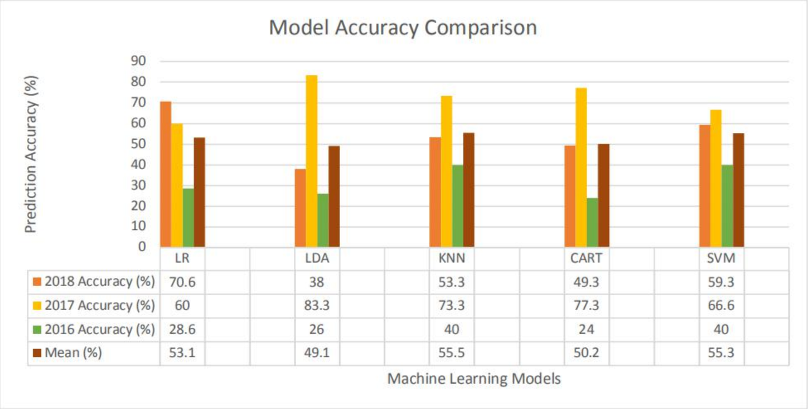
Figure 4: Playoff Prediction Accuracy of Different Machine Learning Models with all data selected and average 30 trial runs

2.2.2 Method 2 – Selecting Partial Data

In this model, we trained the machine learning models with a partial amount of data, which is only based on one Team's performance against a specific opponent. In other words, it is the data between two teams that we are trying to predict.

Based on the data collected (Figure 4), there are no clear trends or patterns available. All the model predictions have significantly large variations in each year. It also doesn't have a clear correlation to data size. Although 2016 was again the worst year of prediction, the 2017 prediction did significantly better than the 2018 prediction, proving that there are no trends.

In summary, selecting only the data where two teams played against each other resulted in inaccurate and inconsistent predictions. It also means that the models' accuracy in this data selection method will not be considered when calculating the best performing model due to the inconsistency and inaccuracy. Therefore, it can be concluded that selecting all data is a better data selection method.



	LR	LDA	KNN	CART	SVM
🟧 2018 Accuracy (%)	70.6	38	53.3	49.3	59.3
🟨 2017 Accuracy (%)	60	83.3	73.3	77.3	66.6
🟩 2016 Accuracy (%)	28.6	26	40	24	40
🟫 Mean (%)	53.1	49.1	55.5	50.2	55.3

Figure 5: Playoff Prediction Accuracy of Different Machine Learning Models with partial data selected and average 30 trial runs

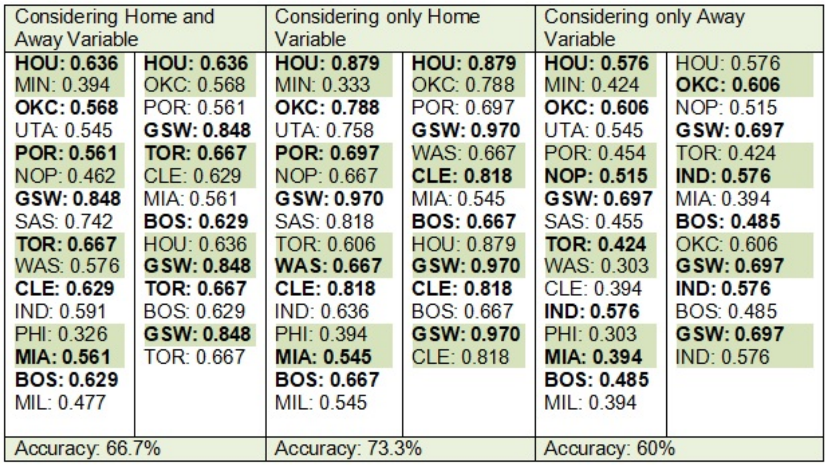
2.2.3 Home and Away Investigation

The home and away variables are widely considered an essential variable on a team's performance and players. The home-court will have more fans, and the positive atmosphere will give the home team a spiritual boost, which may result in a better performance.

An experiment is conducted by training and running the program three times with different data. One will have all data from home and away games, another will have only the data from home games, and the final one will have only the data from away games. Differences between the predicted results are analyzed and evaluated.

Based on the above data (Table 2), generally, teams had much better performance and a higher win percentage when playing as the home team.

To summarize, the home and away variable greatly influenced teams' performance level and win percentage in general. In theory, the variable should not affect model accuracy to a significant extent. But in this case, the model did impact the model accuracy, which can be caused by other factors in real life.



Considering Home and Away Variable		Considering only Home Variable		Considering only Away Variable	
HOU: 0.636	**HOU: 0.636**	**HOU: 0.879**	**HOU: 0.879**	**HOU: 0.576**	HOU: 0.576
MIN: 0.394	OKC: 0.568	MIN: 0.333	OKC: 0.788	MIN: 0.424	**OKC: 0.606**
OKC: 0.568	POR: 0.561	**OKC: 0.788**	POR: 0.697	**OKC: 0.606**	NOP: 0.515
UTA: 0.545	**GSW: 0.848**	UTA: 0.758	**GSW: 0.970**	UTA: 0.545	**GSW: 0.697**
POR: 0.561	**TOR: 0.667**	**POR: 0.697**	WAS: 0.667	POR: 0.454	TOR: 0.424
NOP: 0.462	CLE: 0.629	NOP: 0.667	**CLE: 0.818**	**NOP: 0.515**	**IND: 0.576**
GSW: 0.848	MIA: 0.561	**GSW: 0.970**	MIA: 0.545	**GSW: 0.697**	MIA: 0.394
SAS: 0.742	**BOS: 0.629**	SAS: 0.818	**BOS: 0.667**	SAS: 0.455	**BOS: 0.485**
TOR: 0.667	HOU: 0.636	TOR: 0.606	HOU: 0.879	**TOR: 0.424**	OKC: 0.606
WAS: 0.576	**GSW: 0.848**	**WAS: 0.667**	**GSW: 0.970**	WAS: 0.303	**GSW: 0.697**
CLE: 0.629	**TOR: 0.667**	**CLE: 0.818**	**CLE: 0.818**	CLE: 0.394	**IND: 0.576**
IND: 0.591	BOS: 0.629	IND: 0.636	BOS: 0.667	**IND: 0.576**	BOS: 0.485
PHI: 0.326	**GSW: 0.848**	PHI: 0.394	**GSW: 0.970**	PHI: 0.303	**GSW: 0.697**
MIA: 0.561	TOR: 0.667	**MIA: 0.545**	CLE: 0.818	**MIA: 0.394**	IND: 0.576
BOS: 0.629		**BOS: 0.667**		**BOS: 0.485**	
MIL: 0.477		MIL: 0.545		MIL: 0.394	
Accuracy: 66.7%		Accuracy: 73.3%		Accuracy: 60%	

Figure 6: 2018 LR Prediction Results

2.2.4 Most Accurate Machine Learning Model for NBA Playoff Prediction

To conclude, the most accurate machine learning model at predicting the NBA playoffs is LDA, which reached an accuracy of 71.2%. The performance of the models at predicting with a partial amount of data is neglected since it is considered that the data selection did not give useful information.

3 Discussion

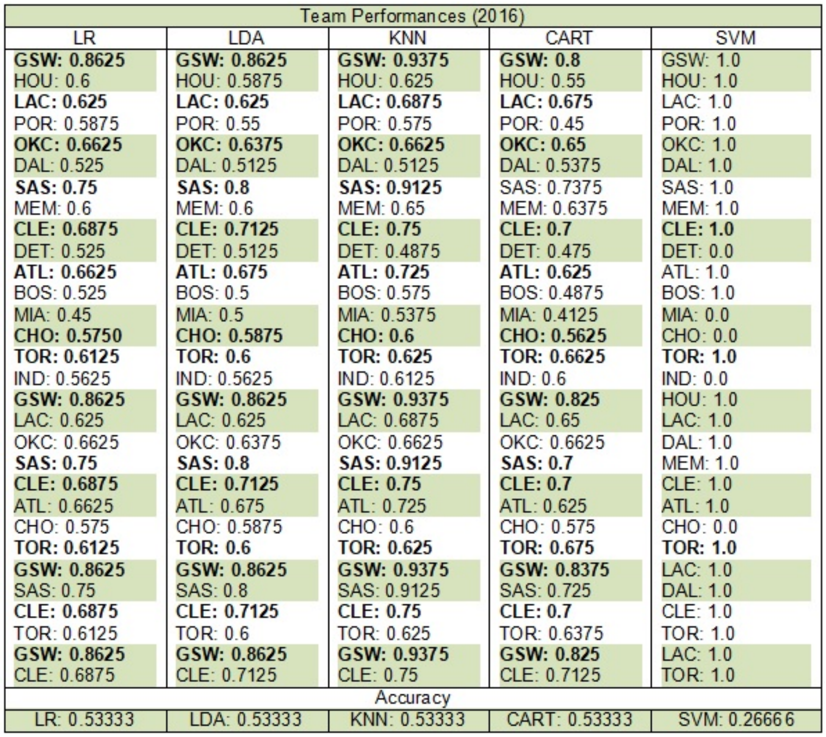
3.1 Self-made prediction model

Based on the model's accuracy and the size of the data, we see a trend between the two variables, with 2018 having the largest dataset and the highest model accuracy and 2016 having the smallest dataset and the lowest model accuracy (Table 1). One possible explanation for the model's changing performance is that it works well with larger datasets while having lower performances when working with smaller datasets.

Another possible reason is that the model is simply not consistent in prediction. It might be a coincidence that there is a correlation between data size and model accuracy since we only have data for three years of prediction. Further investigation can be carried out to confirm the effect of the data size on the model's accuracy. This can be done by running the prediction model for more years with different data sizes to understand the correlation between the two variables better.

3.2 Machine learning prediction model

Similar as the results for the self-made prediction model, the prediction accuracy of 2018 is the highest while the one of 2016 is lowest when all data are selected to train the machine learning models (Figure 3). All algorithms except the SVM model performed significantly worse in 2016. One hypothesis is that the models reached a tipping point in 2016 when the data size is not big enough to support accurate predictions.



Team Performances (2016)				
LR	LDA	KNN	CART	SVM
GSW: 0.8625	**GSW: 0.8625**	**GSW: 0.9375**	**GSW: 0.8**	GSW: 1.0
HOU: 0.6	HOU: 0.5875	HOU: 0.625	HOU: 0.55	HOU: 1.0
LAC: 0.625	**LAC: 0.625**	**LAC: 0.6875**	**LAC: 0.675**	LAC: 1.0
POR: 0.5875	POR: 0.55	POR: 0.575	POR: 0.45	POR: 1.0
OKC: 0.6625	**OKC: 0.6375**	**OKC: 0.6625**	**OKC: 0.65**	OKC: 1.0
DAL: 0.525	DAL: 0.5125	DAL: 0.5125	DAL: 0.5375	DAL: 1.0
SAS: 0.75	**SAS: 0.8**	**SAS: 0.9125**	SAS: 0.7375	SAS: 1.0
MEM: 0.6	MEM: 0.6	MEM: 0.65	MEM: 0.6375	MEM: 1.0
CLE: 0.6875	**CLE: 0.7125**	**CLE: 0.75**	**CLE: 0.7**	**CLE: 1.0**
DET: 0.525	DET: 0.5125	DET: 0.4875	DET: 0.475	DET: 0.0
ATL: 0.6625	**ATL: 0.675**	**ATL: 0.725**	**ATL: 0.625**	ATL: 1.0
BOS: 0.525	BOS: 0.5	BOS: 0.575	BOS: 0.4875	BOS: 1.0
MIA: 0.45	MIA: 0.5	MIA: 0.5375	MIA: 0.4125	MIA: 0.0
CHO: 0.5750	**CHO: 0.5875**	**CHO: 0.6**	**CHO: 0.5625**	CHO: 0.0
TOR: 0.6125	**TOR: 0.6**	**TOR: 0.625**	**TOR: 0.6625**	**TOR: 1.0**
IND: 0.5625	IND: 0.5625	IND: 0.6125	IND: 0.6	IND: 0.0
GSW: 0.8625	**GSW: 0.8625**	**GSW: 0.9375**	**GSW: 0.825**	HOU: 1.0
LAC: 0.625	LAC: 0.625	LAC: 0.6875	LAC: 0.65	LAC: 1.0
OKC: 0.6625	OKC: 0.6375	OKC: 0.6625	OKC: 0.6625	DAL: 1.0
SAS: 0.75	**SAS: 0.8**	**SAS: 0.9125**	**SAS: 0.7**	MEM: 1.0
CLE: 0.6875	**CLE: 0.7125**	**CLE: 0.75**	**CLE: 0.7**	CLE: 1.0
ATL: 0.6625	ATL: 0.675	ATL: 0.725	ATL: 0.625	ATL: 1.0
CHO: 0.575	CHO: 0.5875	CHO: 0.6	CHO: 0.575	CHO: 0.0
TOR: 0.6125	**TOR: 0.6**	**TOR: 0.625**	**TOR: 0.675**	**TOR: 1.0**
GSW: 0.8625	**GSW: 0.8625**	**GSW: 0.9375**	**GSW: 0.8375**	LAC: 1.0
SAS: 0.75	SAS: 0.8	SAS: 0.9125	SAS: 0.725	DAL: 1.0
CLE: 0.6875	**CLE: 0.7125**	**CLE: 0.75**	**CLE: 0.7**	CLE: 1.0
TOR: 0.6125	TOR: 0.6	TOR: 0.625	TOR: 0.6375	TOR: 1.0
GSW: 0.8625	**GSW: 0.8625**	**GSW: 0.9375**	**GSW: 0.825**	LAC: 1.0
CLE: 0.6875	CLE: 0.7125	CLE: 0.75	CLE: 0.7125	TOR: 1.0
Accuracy				
LR: 0.53333	LDA: 0.53333	KNN: 0.53333	CART: 0.53333	SVM: 0.26666

Figure 7: 2016 Team Performance Prediction with All Data Selected

Another hypothesis is that there is an error in the program itself that is causing the 2016 prediction to deviate. This can be seen through the models' accuracy in 2016 (Table 3), except for SVM, all had an accuracy of 53.3%. Here, it demonstrated that each model except for SVM had the same playoff prediction for every round. Although teams have slightly shifting percentages in different models, which may symbolize that there isn't an error and that all models are independent of each other, it is still doubtful that each model had the same win percentage and same prediction. This will be a research question for future investigations to confirm if there is an error in the program causing the deviation in 2016, or the model reached a tipping point in data size that is causing the variation to occur.

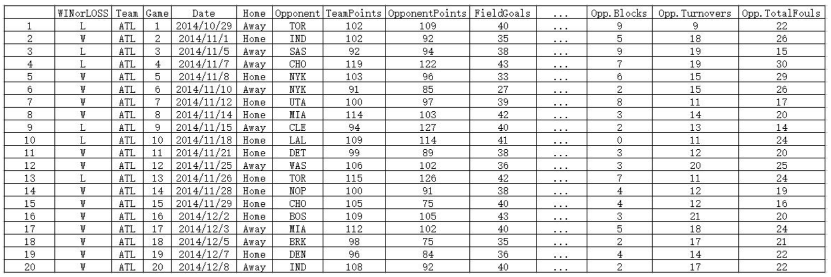
For model training with partial data selection (Figure 4), it can be concluded that partial data selection method gives inaccurate and inconsistent predictions. This is likely because there is very little data for a given pair of teams. To be specific, two teams only play against each other ten times a year, with Team1 playing as the home team for five matches and Team2 playing as the home team for another five games. Additionally, only 80% of the data are used to train, meaning that only eight sets of data are provided for training each year. This resulted in inaccurate predictions with inadequate dataset. It also means that the prediction models will be more likely to give the two teams a 50 percent win rate each due to the small amount of data for testing and training. This will result in the program randomly selecting a winner between the two teams, making the prediction model inconsistent.

Additionally, the two sets of predictions namely home and away should have similar accuracy theoretically. This is because teams typically have a higher win percentage when playing as the home team and a lower win percentage when playing as the away team. If all teams perform better when playing as the home team, they should get roughly the same increase in performance level, so it should not affect the accuracy to a significant extent. This is the same when teams are playing as the away team. They should all perform relatively worse, so the models' accuracy should not shift by a significant amount.

However, in this case (Table 2), the model accuracy did shift significantly, at 13%. This is due to outliers like the team MIN, which had a better performance when playing as the away Team than as the home team. It is also because different teams had different performance levels when playing as the home team. For example, GSW had an increase in a win percentage of 30% when playing as the home team. On the other hand, team PHI only had a 9% increase in win percentage when playing as the home team. One hypothesis is that GSW has more fans than other teams, so they have a better atmosphere when playing as the home team. However, many other factors can decide a team's performance when playing as the home team and the away team. These factors can be further investigated in the future.

4 Methods

To test the effectiveness of our self-made NBA playoff prediction model and all the related machine learning algorithms, certain NBA historic statistics data from 2014 - 2018 are needed, which can be access from many open source NBA statistics. And these historic NBA statistics are usually saved as .csv file format, which we can use the Python pandas library read-csv module to load the dataset from the corresponding csv URL link, the format and header of the dataset is of the following form (in Figure 5.)



	WINorLOSS	Team	Game	Date	Home	Opponent	TeamPoints	OpponentPoints	FieldGoals	...	Opp.Blocks	Opp.Turnovers	Opp.TotalFouls
1	L	ATL	1	2014/10/29	Away	TOR	102	109	40	...	9	9	22
2	W	ATL	2	2014/11/1	Home	IND	102	92	35	...	5	18	26
3	L	ATL	3	2014/11/5	Away	SAS	92	94	38	...	9	19	15
4	L	ATL	4	2014/11/7	Away	CHO	119	122	43	...	7	19	30
5	W	ATL	5	2014/11/8	Home	NYK	103	96	33	...	6	15	29
6	W	ATL	6	2014/11/10	Away	NYK	91	85	27	...	2	15	26
7	W	ATL	7	2014/11/12	Home	UTA	100	97	39	...	8	11	17
8	W	ATL	8	2014/11/14	Home	MIA	114	103	42	...	3	14	20
9	L	ATL	9	2014/11/15	Away	CLE	94	127	40	...	2	13	14
10	L	ATL	10	2014/11/18	Home	LAL	109	114	41	...	0	11	24
11	W	ATL	11	2014/11/21	Home	DET	99	89	38	...	3	12	20
12	W	ATL	12	2014/11/25	Away	WAS	106	102	36	...	3	20	25
13	L	ATL	13	2014/11/26	Home	TOR	115	126	42	...	7	11	24
14	W	ATL	14	2014/11/28	Home	NOP	100	91	38	...	4	12	19
15	W	ATL	15	2014/11/29	Home	CHO	105	75	40	...	4	12	16
16	W	ATL	16	2014/12/2	Home	BOS	109	105	43	...	3	21	20
17	W	ATL	17	2014/12/3	Away	MIA	112	102	40	...	5	18	24
18	W	ATL	18	2014/12/5	Away	BRK	98	75	35	...	2	17	21
19	W	ATL	19	2014/12/7	Home	DEN	96	84	36	...	4	14	22
20	W	ATL	20	2014/12/8	Away	IND	108	92	40	...	2	17	22

Figure 8: NBA historic statistics dataset format and headers

4.1 Self-made prediction model

To start, we first created our own prediction model to predict the NBA bracket. We focused on working out the probability of Team A winning against Team B, then applying this to every game in the playoff.

We have to narrow our focus on specific game variables, which significantly impact the game result. After some research, we decided to use the following variables:

1)EFG% effective field goal percentage [Aut12], considers both 2pts field goals and 3pts field goals in one variable and considered their weight with three-pointers worth 1.5 times of a two-pointer.

2)FT% free throw percentage [Aut12], calculates the percentage of free-throw makes for a specific team.

3)TOV% turn over percentage [Aut12], is an estimate of turnovers by a team per 100 possessions.

4)ORB% Offensive rebound percentage [Aut12], is an estimate of the percentage of offensive rebound that a team gets.

5)DRB% Defensive rebound percentage [Aut12], is an estimate of the percentage of defensive rebounds taken by a team.

$$EFG\% = \frac{(2\_point\_field\_goals\_made + 1.5 * 3\_point\_field\_goals\_made) * 100}{Total\_field\_goals\_made}$$

$$FT\% = \frac{free\_throws\_made * 100}{free\_throws\_attempted}$$

$$TOV\% = \frac{number\_of\_turnovers * 100}{field\_goal\_attempted + 0.44 * free\_throws\_attempted + number\_of\_turnovers}$$

$$ORB\% = \frac{offensive\_rebounds * 100}{offensive\_rebounds + opponent\_defensive\_rebounds}$$

$$DRB\% = \frac{defensive\_rebounds * 100}{defensive\_rebounds + opponent\_offensive\_rebounds}$$

4.1.1 Algorithms

The five variables that were chosen are considered the most impactful factors in the game. The second step of our model is to decide on the algorithm we are going to use to calculate the probability of Team A beating Team B; the chosen algorithm was:

$$P_{win} = c_1P_1 + c_2P_2 + c_3P_3 + c_4P_4 + c_5P_5$$

Here, c_i is the proportional correlation of the variable v_i with winning. In other words, the larger the value of c_i, the more variable v_i will contribute to the winning of a game. Pi is the probability that Team A will have a higher score than Team B on variable v_i. By multiplying the probability of the two factors together and adding all the numbers up for all five different variables, we predict Team A beating Team B in a match.

4.1.2 c_i calculation

The formula for c_i is:

$$c_i = \frac{r_i}{r_1 + r_2 + r_3 + r_4 + r_5}$$

Here, r_i represents the Pearson correlation coefficient of the variable v_i with winning. However, winning is a categorical value that cannot be used in the Pearson correlation. Therefore we decided to represent winning with the point difference between the two teams.

4.1.3 P_i calculation

To calculate the value for P_i, we used the principle of confidence intervals, which is defined to be the probability that a parameter will fall between two sets of values with a specific confidence level. [Wil13]

1 - Calculate a 95% confidence interval of for both teams

2 - We defined the confidence intervals for Team A as $[x_A, y_A\ ]$ and Team B as $[x_B, y_B\ ]$

3 – The first case is when the intervals don't overlap. In this situation, the Team with the higher interval has a 95% chance of scoring higher. (Note: The percentage might be slightly higher than 95%, but in this case, we consider it as 95%.)

4 - The second case is when the two intervals overlap, and Team A has a higher upper limit (y_A ¿ y_B). Here, the formula to calculate Pi is:

$$P_i = 0.95\frac{y_A - y_B}{y_A - x_B}$$

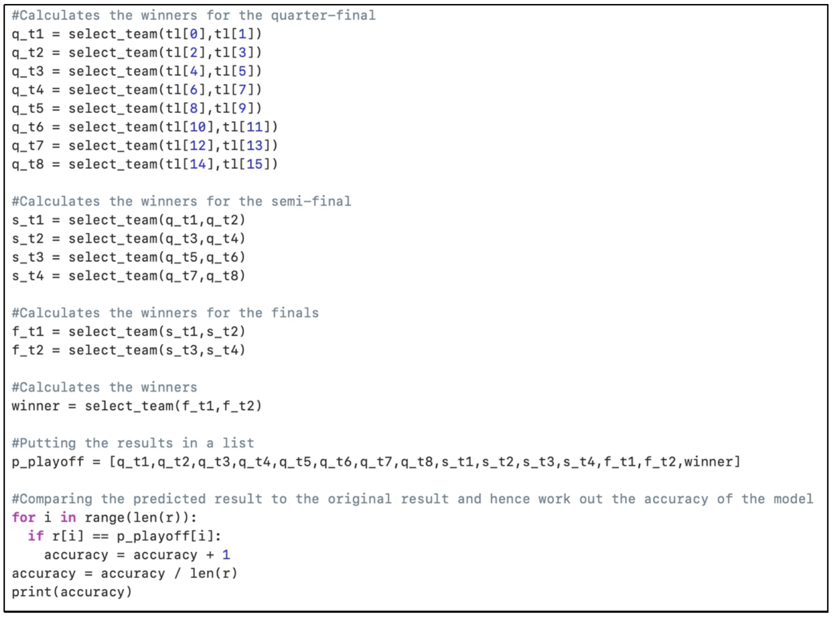
5 - The third case is when Team B has a higher upper limit (y_B¿y_A). Here, the formula to calculate P_i is:

$$P_i = 1 - 0.95\frac{y_B - y_A}{y_A - x_B}$$

6 – The last case is when the two sets of data have the same upper limit (yB=yA). Then Pi = 0.5 in this case.

4.2 Predicting the playoff bracket

In order to predict the playoff bracket, we created Python's function to calculate the probability of Team A defeating Team B, and we applied it to predict the playoff bracket for 2018.



```python
#Calculates the winners for the quarter-final
q_t1 = select_team(tl[0],tl[1])
q_t2 = select_team(tl[2],tl[3])
q_t3 = select_team(tl[4],tl[5])
q_t4 = select_team(tl[6],tl[7])
q_t5 = select_team(tl[8],tl[9])
q_t6 = select_team(tl[10],tl[11])
q_t7 = select_team(tl[12],tl[13])
q_t8 = select_team(tl[14],tl[15])

#Calculates the winners for the semi-final
s_t1 = select_team(q_t1,q_t2)
s_t2 = select_team(q_t3,q_t4)
s_t3 = select_team(q_t5,q_t6)
s_t4 = select_team(q_t7,q_t8)

#Calculates the winners for the finals
f_t1 = select_team(s_t1,s_t2)
f_t2 = select_team(s_t3,s_t4)

#Calculates the winners
winner = select_team(f_t1,f_t2)

#Putting the results in a list
p_playoff = [q_t1,q_t2,q_t3,q_t4,q_t5,q_t6,q_t7,q_t8,s_t1,s_t2,s_t3,s_t4,f_t1,f_t2,winner]

#Comparing the predicted result to the original result and hence work out the accuracy of the model
for i in range(len(r)):
  if r[i] == p_playoff[i]:
    accuracy = accuracy + 1
accuracy = accuracy / len(r)
print(accuracy)
```

Figure 9: Python Code Snippet for Self-made Prediction Model

In the python code snippet (Figure 6), the function select_team(), which predicts the winner between Team A and Team B, is called many times. This calculates the winners for the quarter-final, the semi-final, the finals, and in the end, it calculates the winner of the year. This predicted playoff is then appended to a list and compared to the actual result of the 2018 playoff to calculate a prediction accuracy. The original playoff is pre-loaded into the program beforehand.

4.3 Machine learning prediction models

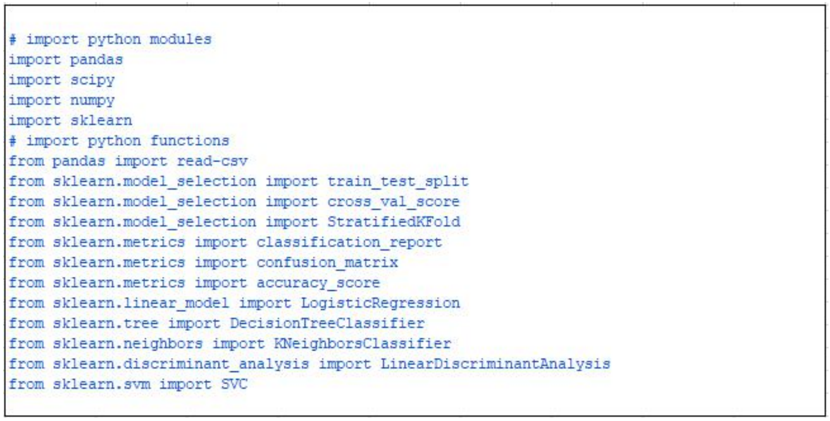
In order to test if machine learning algorithm can be used to predict the NBA playoff bracket and evaluate which machine learning model has the best prediction accuracy, 5 different machine learning models that have been implemented in Python Scikit-learn machine learning library are employed with two linear (LR and LDA) and three nonlinear (KNN, CART and SVM) ones (Figure 6).

After loading the dataset from the historic NBA statistics CSV file, depending on the two different data selection methods, all data or partial data, together with home or away analysis, data related to year of 2016, 2017 and 2018 for NBA playoff prediction can be split into different data arrays, so that the prediction accuracy of different machine learning models can be analyzed accordingly.

To test the prediction accuracy for each different machine learning model, the dataset needs to be split into two sets, one for the model training and one for the model prediction on 8:2 randomly selection basis, which means 80% of the data will be used as training data and 20% will be used to evaluate the prediction accuracy and the data is randomly selected.

After the dataset is split for training and validation, the fit function for each machine learning model will be called to train each individual models. After model training, the predict function for each machine learning model will be called to make the final prediction based on the validation dataset generated before and the prediction accuracy for each models will also be calculated by the accuracy$_{s}corefunction(Figure7)$.

Note that in Python, loc function is a frequently used function to retrieve partial data in the dataset related to certain variable value, like certain year, certain team, etc.

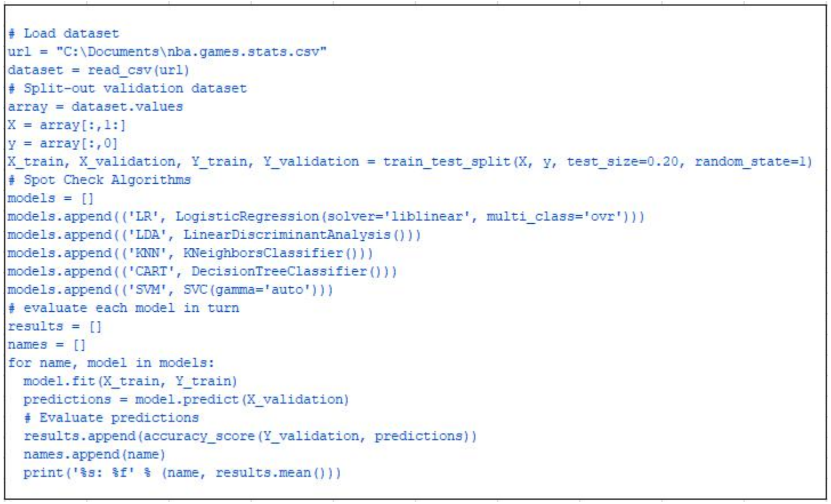


```
# import python modules
import pandas
import scipy
import numpy
import sklearn
# import python functions
from pandas import read-csv
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.svm import SVC
```

Figure 10: Python code snippet for importing modules, functions and models

5 Acknowledgement

I would like to thank Mr. Derek Sorensen of Horizon Academic Research Program (HARP), together with my parents, my sister for the support.



```
# Load dataset
url = "C:\Documents\nba.games.stats.csv"
dataset = read_csv(url)
# Split-out validation dataset
array = dataset.values
X = array[:,1:]
y = array[:,0]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, test_size=0.20, random_state=1)
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
  model.fit(X_train, Y_train)
  predictions = model.predict(X_validation)
  # Evaluate predictions
  results.append(accuracy_score(Y_validation, predictions))
  names.append(name)
  print('%s: %f' % (name, results.mean()))
```

Figure 11: Python code snippet for dataset split and model cross-evaluation

References

[Anu06] Mehta Anukrati. A beginner's guide to classification and regression trees. *https://www.digitalvidya.com/blog/classification-and-regression-trees/*, 0006.

[Aut01] No Author. National basketball association. *https://en.wikipedia.org/wiki/NationalBasketballAssociation*, 0001.

[Aut09] No Author. Machine learning - logistic regression. *https://www.tutorialspoint.com/machinelearningwithpython/machine-learningwithpythonclassificationalgorithmslogisticregression.htm*, 0009.

[Aut10] No Author. Advantages and disadvantages of logistic regression. *https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/*, 0010.

[Aut12] No Author. Glossary. *Basketball Reference, No Date, https://www.basketballreference.com/about/glossary.htmltovpct*, 0012.

[Cor05] Maklin Cory. Linear discriminant analysis in python. *https://towardsdatascience.com/linear-discriminant-analysis-in-python-76b8b17817c2*, 0005.

[Dhi03] K Dhiraj. Top 5 advantages and disadvantages of decision tree algorithm. *https://medium.com/@dhiraj8899/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a*, 0003.

[Jos08] Starmer Josh. Linear discriminant analysis (lda) clearly explained. *https://www.youtube.com/watch?v=azXCzI57Yfct=516s*, 0008.

[Nar04] Kumar Naresh. Advantages and disadvantages of knn algorithm in machine learning. *http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html*, 0004.

[Rus07] Pupale Rushikesh. Support vector machine (svm) – an overview. *https://towardsdatascience.com/https-medium-com-pupalerushikesh-svmf4b42800e989*, 0007.

[Sil02] David Silver. Mastering the game of go without human knowledge. *ARTICLE, doi:10.1038/nature24270*, 0002.

[TA11] Ibrahem Abdlehameed Hassanien Aboul Ella Tharwatt Alaa, Gaber Tarek. Linear discriminant analysis: A detailed tutorial. *https://www.researchgate.net/publication/316994943Linear-discriminantanalysisAdetailedtutorial*, 0011.

[Wil13] Kenton Wil. Confidence interval. *https://www.investopedia.com/terms/c/confidenceinterval.asp*, 0013.