NBA Statistic analysis and match prediction

Keyi Ren(A53304040) Bowen Sun(A53319090)

Abstract—In this project, our team made a predictor based on technical statistics from NBA regular seasons from 2000-2018 to predict the winner of a single match. There are two parts in our project, data analysis and classification (or prediction). For the first part, we analyzed some of the most important factors that affect winning rate, including REB, TOV and FG%. Since our dataset is perfectly vectored, we used SVM, KNN, Logic Regression and many other conventional machine learning models to build a match predictor with more than 80% accurancy on testet.

Index Terms—NBA, Data analysis, Data classification, Model prediction, Machine Learning.

1 Introduction

NBA is a world-famous basketball sport league (Fig.1) which contains 30 teams and stands for world's highest basketball level. NBA takes one year as a competition season which contains regular and playoff seasons. As other sport events, NBA teams try their best to win every game, the final champion of a season can win millions of dollars and can gain great honor and attention. Not only the teams themselves, matching results are also important to fans and some gambling websites. As a large amount of people interested in game results, it will be very helpful if it can be predicted.

We suppose everyone knows basketball and only give a brief description on the NBA game rule. A single NBA match is 4 quarter long with first 2 as first half and second 2 as second half, each quarter has 12-minute duration, each half opponents should switch side. The one who gains more points at the end of regular time wins the game. If there is a draw, a 5-minute overtime is added until there is a winner.

Gathering NBA technical statistics started from 1980s, these information are so important for fans, reporters, scout and team managers. Some sports lottery even use them to predict winning rate and make an opening based on that. As a matter of fact, our goal is to make a predictor based on different models to predict winner of a game. But it seems trivial to build such a binary classification based on state-of-the-art algorithms. And many other works have been done on NBA data analysis.

What our work differs from previous work is, we will try to figure out the importance of each factor that mostly affects victories, and make them a visualization. For example, rebound, or REB for short, can represent the defense effectiveness and energy of a team; turnover or TOV, suggests how concentrated the players are and their deployment of tactical coordination. All these statistics are of vital importance and combined to make a game, this makes our advanced analysis valuable.

1.1 Terms

We present some of the most common NBA terms in advanced statistics, which are what we used in our NBA dataset.
PTS: achieved points;

Fig. 1: What a competition

FGM: filed goals made;
FGA: filed goal attempted;
FG%: filed goal percentage;
3PM: three-point filed goals made;
3PA: three-point filed goals attempted;
3P%: three-point filed goals percentage;
FTM: free throws made;
FTA: free throws attempted;
FT%: free throws percentage;
OREB: offensive rebound;
DREB: defensive rebound;
REB: rebounds (OREB + REB);
AST: assists;
STL: steals;
BLK: blocked shots;
TOV: turnovers;
PF: personal fouls;
+/-: scoring margin;

1.2 Teams

One of the most important attributes is team, there are totally 30 teams now in the league, while some of them had changed team name, we then have 36 different unique teams in our dataset, they are:
UTA: Utah Jazz

Fig. 2: Teams now in the League

SAS: San Antonio Spurs
SAC: Sacramento Kings
POR: Portland Trail Blazers
PHI: Philadelphia 76ers
ORL: Orlando Magic
OKC: Oklahoma City Thunder
NYK: Ner York Knicks
MIN: Minnesota Timberwolves
MIL: Milwaukee Bucks
MIA: Miami Heat
MEM: Menphis Grizzlies
LAC: Los Angeles Clippers
IND: Indiana Pacers
GSW: Golden State Warriors
DET: Detroit Pistons
DEN: Denver Nuggets
DAL: Dallas Mavericks
CHI: Chicago Bulls
CHA: Charlotte Hornets
BKN: Brooklyn Nets
ATL: Atlanta Hawks
WAS: Washington Wizards
TOR: Toronto Raptors
PHX: Phoenix Suns
NOP: New Orleans Pelicans
LAL: Los Angeles Lakers
HOU: Houston Rockets
CLE: Cleveland Cavaliers
BOS: Boston Celtics
NOH: New Orleans Hornets (now NOP)
NJN: New Jersey Nets (now BKN)
SEA: Seattle SuperSonics (now OKC)
NOK: Norwgian Krone (not an nba team)
CHH: Charlotte Hornets (now CHA)
VAN: Vancouver Titans (not an nba team)

We introduced the team name since we analyzed the winning rate of themselves and winning rate when a team vs. another between 2000-2018. However, we found the latter intrinsically suffered insufficient samples, which leads the plot to be weird, we them discard them. Due to historical reason, W/L balance, simplicity and personal emotions, we decided to carry further analysis on SAS, LAL, GSW (west region), ORL, BOS and CLE (east regison).

1.3 Main Contribution and Structure

Our main contribution can summarized as:

- We organized and made a clean NBA statistic dataset from NBA.com, it is a comparable scale, easy handling dataset for data analysis and model prediction.
- We analyzed factors mostly affect winning rate and visualized them.
- We carried experiments on conventional machine learning models, and achieved 85% test accuracy on playoff testset.

Our report is organized as follows: **Section.2** presents a detailed description about our NBA dataset, including our trainset, valset and testset; **Section.3** shows our advancde analysis on 6 NBA teams; while we experiment on our prediction model on **Section.4**; finally we conclude our experiments and make a summary on **Section.5**.

2 NBA Statistic Dataset

Here we introduce our NBA statistic dataset, which is found and collected from NBA Advanced Stats: https://www.nba.com/. Since we don't have off-the-shelf dataset in kaggle, we manually gathered raw data and cleaned them for our prediction part. As previously suggested, we leveraged just the statistics of regular season from 2000-2018, which yield us 22960 games of 18 seasons. Since we are talking about 1 vs. 1 competitions, we totally have 22960×2 samples without repetition(as shown in Table.1). We followed a typical classification baseline, which makes our raw dataset into trainset, valset, and testset. However, in order to evaluate the robustness of classifiers, we randomly divided the raw dataset into trainset and valset only, with each of 36736 and 9184 regular games (Table.1). What worth mentioning is our testset, which are collected 1309 playoff season games from 2000-2018. Since we all know playoff seasons should have different distribution from regular seasons for fierce body contact and more targeted tactics, testing on this kind of set really makes sense.

TABLE 1: NBA Dataset

	Season Type	Size
Trainset	Regular	36736
Valset	Regular	9184
Testset	Playoff	1309

2.1 Attributes

We give a brief view of what our raw dataset looks like in Table.2, each attribute(the first row in Table.2) are illustrated in **Section.1.1**. Besides, the raw data contains complemented **TEAM**, **DATE**, **MATCHUP**, **W/L** which represent team name, match date, match-up information and win or lose respectively. They are not important and may mislead our

TABLE 2: NBA Statistic Dataset(partial)

TEAM	DATE	MATCHUP	W/L	MIN	PTS	FGM	FGA	FG%	3PM	3PA	3P%	FTM	FTA	FT%	OREB	DREB	REB	AST	STL	BLK	TOV	PF	+/-
GSW	06/13/2019	GSW vs. TOR	L	240	110	39	80	48.8	11	31	35.5	21	30	70	11	31	42	28	9	6	16	23	-4
TOR	06/13/2019	TOR @ GSW	W	241	114	39	82	47.6	13	33	39.4	23	29	79.3	11	28	39	25	8	2	12	23	4
...																							
CHH	04/21/2001	CHH @ MIA	W	240	106	37	77	48.1	7	15	46.7	25	29	86.2	9	30	39	21	17	4	15	26	26

*Note that this is only a part data of testset, we are only giving a broad view here, other set such as trainset and valset share the same attributes

classifiers, similarly, **PTS** and **+/-** are most deterministic information that indicates wins or loses which severely biased our classifiers, thus in prediction stage we simply removed all of them.

3 Advanced Team Analysis

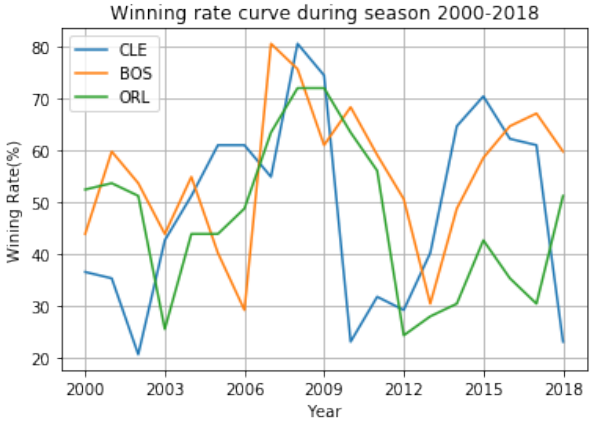
As mentioned above, we carried analysis on San Antonio Spurs, Cleveland Cavaliers, Boston Celtics, Golden State Warriors, Los Angeles Lakers and Orlando Magic. What we cared most are their static statistics and conditional winning rate, e.g., average rebounds, average FG%, winning rate as seasons, winning rate given a value of TOV. Normally teams with good shots or defense win the game, but other information such as game duration, personal fouls may seem not so importance in winning the game. Therefore for our experiments, we presented **Winning Rate as Seaons, Winning Rate given AST, PG%, OREB, REB, DREB and TOV** for each given team.

3.1 Winning Rate as Seasons

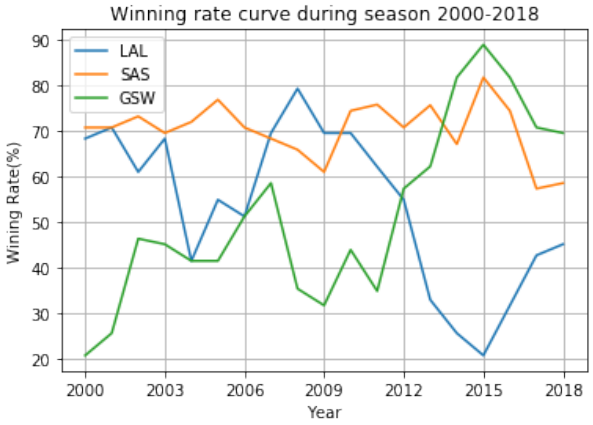
Winning rate of each season can directly reflect the team strength, so we put it on the first section. The figures are shown in Fig.3. From what shown in Fig.3, one can notice that all the other teams, other than SAS, go through many ups and downs, which is intuitive since there are unstable factors such as injuries, trades and lineup depth which play important role in a team's clutch status.

By combining those figures with existing facts, we can found how important a super star plays in leading the team. You can find CLE played high-level games from 2003 to 2009 (50% winning rate), but suddenly dropped to no more than 30% on 2010 and remained in the trough until 2014. What caused such fluctuations is just Lebron James, who was drafted as the 1_{st} pick on 2003 and played as MVP level right after he entered the league. His leaving on 2010 to Miami Heat made CLE a lotto team and his return on 2014 changed it completely to one of the most powerful competitors of NBA final champion. Similarly, you may not be surprised to find the extreme increment of BOS on 2007 compared to 2006 because they obtained two all-star Ray Allen and Kevin Garnett through trades; DPOY Dwight Howard' left on 2012 also made his old owner Orlando Magic went into rebuilt.

Not only the effect of super stars, these curves are also consistent with the injuries of specific player, e.g, LAL's winning rate decreased by 40% after Kobe Brant had a Achilles tendon rupture on 2013. Interestingly, SAS remained a high winning rate (surprisingly 70.23%)from 2000-2018, not only because they got famous GDP threesome, but they owned a strict tactical system leaded by Gregg Popovich. We will not present too much, but they are all consistent with NBA history between 2000-2018.



(a) East Region



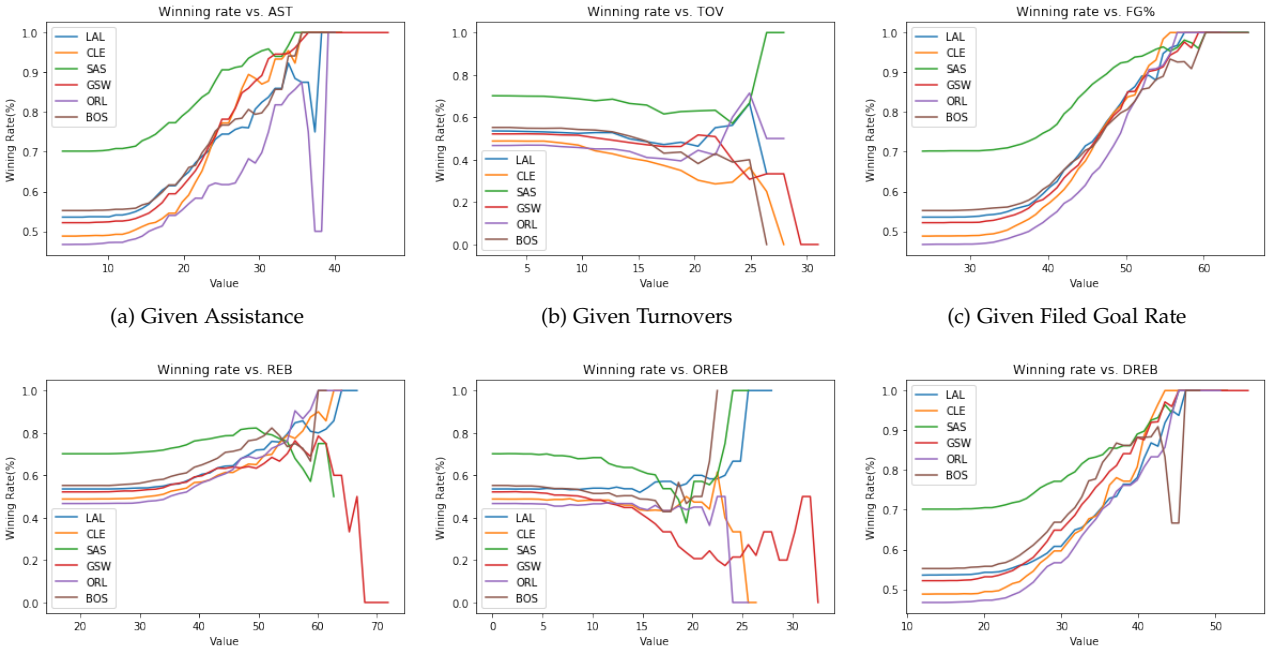
(b) West Region

Fig. 3: Winning Rate as Seasons

3.2 Conditional Winning Rate

Similarly to what reported by ESPN, we analyzed on the winning rate given certain conditions. We call each condition as tag, which is a specific value of attribute. We then counted the winning rate of games with equal or larger value than the given tag and plot them as tag increases, as shown in Fig.4.

From them we conclude that AST, REB, DREB and FG% are almost proportional to a team's winning rate, which is intuitive since they either suggest good shot or tough defense. However, what's interesting is, we witnessed some sudden drop as those number increases, e.g., purple and blue curves in Fig.4-(a), red and green curves in (d) and brown curve in (f). One of the reasons is because some of the teams did not have many games with higher tag. Suppose a team has only 3 games with AST higher than 35, but unfortunately these games are all played with powerful opponents and they lose 2, the winning rate is 33% only, but



(a) Given Assistance (b) Given Turnovers (c) Given Filed Goal Rate

(d) Given Rebound (e) Given Offensive Rebound (f) Given Defensive Rebound

Fig. 4: Conditional Winning Rate

that does not mean AST plays a negative role. Besides, it also depends on the style of certain team, e.g., SAS and GSW are known as playing team basketball, their curves in (a) are strictly monotonically increasing. Additionally, we are not surprised to find winning rate decreases as TOV increases.

We find only higher FG% leads absolute higher winning rate, not FGM, not FGA, only FG%. It may be inconsistent with our belief because we consider more shot made, more possibilities a team should win. But what our data tell us is, we should pay more attention on the pure efficiency. As a matter of that, we hypothesize that, FG% is one of the most important features in our prediction stage, which we will discuss later.

What's more, we find figure of OREB a very interesting one, one may consider OREB as a symbol of energy and initiation, team who gets more OREB is supposed to have more second offenses, which should yield higher points and higher chance of win. Nevertheless, from a different perspective, more OREB also means more missing shot and lower PG%. By what we concluded above that winning rate has closest relation to FG%, it is convincing that winning rate does not have to be positive proportional to OREB.

4 Classification & Prediction

We carried broad experiments on Decision Tree, Random Forest, KNN, Logistic Regression, SVM and MLP. Before classification, we visualized the samples using T-SNE [5], we projected 18-D data into 3-D space in order to see if our data is separable (Fig.5). Even visualization shows the data are densely distributed, one can still tell a boundary between purple and yellow dots. We consider it is because we are facing a binary classification, while T-SNE performs much better in multi-class tasks.

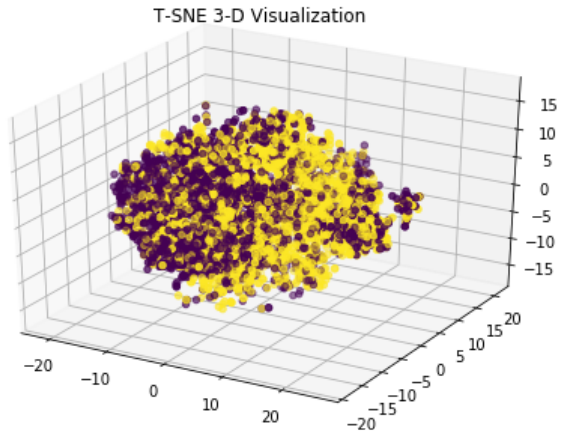


Fig. 5: T-SNE 3-D Visualization

4.1 Decision Tree [6]

The decision tree classifier divides the data based on the features and it tried to divide the data with a combination of thresholds in different features. In decision trees, each node in the tree represents an object, and each branching path represents a possible attribute value, and each leaf node corresponds to the sum of objects from root to that leaf node.

In the Decision Tree, the goal is to find a path with the biggest gain which can be represented as:

$$Entropy(S) = \sum_{n=1}^{c} -p_i log_2 p_i$$
$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{S} Entropy(S) \quad (1)$$

Where $V(A)$is the range of attribute A, S is the set of samples and S_v is a set of S equal to the sample V in the value of attribute A. In each step, the algorithm will cut those with bigger cost to accelerate the process. However, it may trap into local optimum situation. The advantage of this model is intuitive, easy to understand and effective for small data sets. However, performance improved slowly when the training data increases.

4.2 Random Forest [4]

We can regard the Random Forest as a combination of different Decision Trees. In this algorithm, many Decision Trees are generated and trained independently with randomly chosen data. When doing the prediction, the probability is a combination of prediction results from all individual trees.

Compared to single Decision Tree, Random Forest is more stable, less likely to be overfitting and has better performance in large data. However, this model is much more complex and needs more time to train.

4.3 Support Vector Machine [7]

Support Vector Machine is another kind of linear classifier which makes a hyperplane to classify different kind of examples.

We can write the function object function as:

$$argmax_{w,b} \left\{ min(y(w^T x + b)) \frac{1}{||W||} \right\} \quad (2)$$

This objective function can be solved with some techniques such as Lagrange function and relaxation variables. And a traditional SVM is finished after the hyperplane is found.

Typically, compared to Logistic Regression, SVM performs much better to the samples around boundary and it has great generalization ability. What's more, it can solve nonlinear problems with different kernel function. However, the training is much more time-consuming and the results in our experiment show inferior than Logistic Regression.

4.4 KNN [3]

The KNN classifier is the simplest classifier that needs no training. When classifying, the KNN classifier simply search for the nearest neighbor of the unknown sample in the classified samples. The distance between sampled can be defined differently based on different predicting tasks.

4.5 Logistic Regression [2]

In Logistic Regression model, the data and results are assumed to obey a model like:

$$h_\theta(x) = g(\theta^T X)$$
$$g(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

In the above formulas, X is a vector formed with input features, $\theta^T$is the vector of unknown parameters of all features and will be modified in the training to best fit the data (which means minimize the MSE). $g(z)$ is a monotonically increasing function in range(0,1) which has much greater slope in the region closer to 0. The $h_\theta(x)$ stands for the probability of the output being "1".

When using this model to make prediction, it is common to set a threshold since $h_\theta(x)$ has a continuous range. As $g(z)$ has a much greater slope around 0 and intersects with y-axis at 0.5, it is quite reasonable to set 0.5 as a threshold to distinguish positive and negative results.

The most significant advantage of Logistic Regression is its simple and easy to use. There is no need to scale features and the training this model is respectively efficient. However, Logistic Regression has bad performance to the data near the boundary and it is a linear model that cannot solve nonlinear problems. Our experiments suggest that Logistic Regression has the most superior performance on the given dataset.

4.6 MLP [1]

At last, we tried a Multi Layer Perceptron, which contains at least three layers: an input layer, some(equal to or more than 1) hidden layers and a output layer. The function of a hidden layer is:

$$X_1 = h_1(W_1 X + b_1) \quad (4)$$

Where X is the input from input layer, W_1 is parameter of X and b_1 is bias, f can be sigmoid or ReLU function:

$$sigmoid(a) = \frac{1}{1 + e^{-a}}$$
$$ReLU(a) = max(a, 0) \quad (5)$$

After several hidden layers (2 in our experiment), the function of output layer can be represented as:

$$G(X1) = softmax(h_2(h_1(X)))$$
$$softmax(x_i) = \frac{e^{-x_i}}{\sum_i^C e^{-x_i}} \quad (6)$$

where C is the number of classes

5 Experiment

5.1 Implementation Detail

We simply leveraged **sklearn** implementation for the first 5 models and implemented MLP on deep learning framework PyTorch. For sklearn models, we used different configuration of parameters and found them had vital influences on the performance, thus at the very end we simply used default configuration. For PyTorch implementation of MLP, since we removed some unrelated and biased attributes

TABLE 3: Prediction Performance

Model	Val Acc	Test Acc
Random Forest	79.78	80.14
KNN	75.98	75.78
Decision Tree	73.11	74.79
Logistic Regression	**84.45**	**86.17**
SVM	70.93	64.32
MLP	83.88	85.41

and left 18 features for prediction, we implement a 2-layer MLP with the first as $Linear(18, 10)$ and second as $Linear(10, 10)$, where $Linear(input, output)$ is a learnable linear function. At training stage, we trained the classifier with Adam optimizer and Cross Entropy loss function with a 256 mini-batch for 30 epochs, which is close to convergence. We validated on Valset each epoch, choosing the best model to test on Testset.

5.2 Results

The performance of six classifers are shown in Table.3.

5.3 Further Analysis

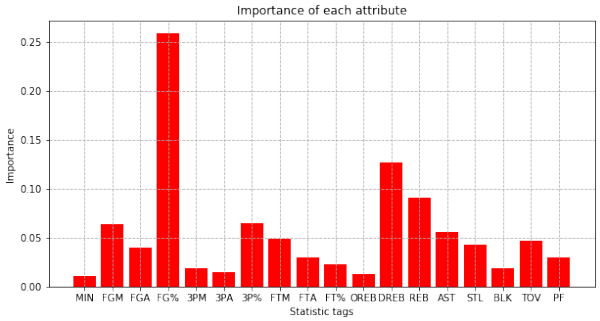
One can see from Table.2 that Logistic Regression performs the best, then MLP, with test accuracy of over 85%. And we mentioned before that, even though we trained and validated on regular season games, it makes no difference from test performance which indicates that our model can generalize pretty well. We then plot the feature importance. We could have plot the gradient of MLP and then calculate the importance of each feature, but due to time limits, we simply plot those of Random Forest and Decision Tree as Fig.6 by using $feature\_importance\_$ function in sklearn.

We found that FG% is the most important feature among 17 other features, followed by DREB, REB, TOV and 3P%. This is consistent with our analysis in **Section 3.2**, where we hypothesized that FG% is one of the most important features.

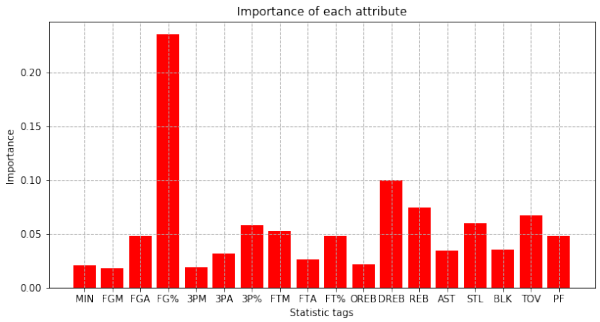
Also, all of our classifiers cannot achieve more than 90% accuracy, though they are really capable in dealing with such kinds of data. That reveals statistics are not the only prediction of wins or loses, we still have something untraceable that influences the game. Also, in such a competitive league, any team has the chance to win, any team can compete for the final champion.

6 Conclusion

Through this project, we analyzed NBA teams' performance, we found SAS remained high-level from 2000-2018, which made them 4 champions. We also revealed some factors that affect winning a game, such as REB, PG%, TOV and AST, PG% is the most important factor among them, a team gains the largest chance to win if they make efficient shots.



(a) Random Forest



(b) Decision Tree

Fig. 6: Feature importance

References

[1] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[2] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[3] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[4] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[6] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[7] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.