NBA Match Outcome Prediction

Neil Gibeaut
Computer Science & Engineering
Texas A&M University
ncg19@tamu.edu

Kevin Thomas Mathew
Computer Science & Engineering
Texas A&M University
kmathew96@tamu.edu

Nishant Barma
Computer Science & Engineering
Texas A&M University
nishant.barma@tamu.edu

***Abstract*—The ability to predict outcomes of a basketball game based on historical data has long been the goal of many researchers, basketball enthusiasts and gamblers. This report discusses the several machine learning models used to accurately predict the winners and losers of the regular season basketball games of the NBA (National Basketball Association) league. For the game outcome prediction, we used five different machine learning classifiers: i) *Logistic Regression*, ii) *Support Vector Machines*, iii) *Random Forest Classifier*, iv) *Neural Networks*, and v) *AdaBoost Classifier*. We evaluated each of these models in a variety of settings, and found them to perform comparably to NBA experts at predicting match outcomes.**

***Index Terms*—classifier, outcome, prediction**

I. Introduction

It has long been the goal of many researchers to correctly predict outcomes based on historical evidence. Sporting games have a record of observable data and a basketball match is no exception. This data can be used extensively for analysis. Although it is an unexpected field, this has led to sports analytics which uses historical data on players, teams, experts and matches to predict outcomes of future games and provide coaches the opportunity to come up data-backed strategies.

In a regular NBA season, over 1200 games are played and the games are recorded with real-time scores and video footage. All of this data provides researchers and analysts with a plethora of information to use in machine learning and models for optimizing the inferences and predictions in a sporting contest. These learning methods can also be used to analyze team and individual performance to develop new strategies. Previously, coaches and players had to watch hours of recorded game footage to come up with new strategies and decisions, but with the help of machine learning methods, this process can become more efficient. This can also change the way viewers watch and interact with a game, while simultaneously giving the players and coaches more control on their performance. The NBA teams can also decide whether to sell or buy a player based on their data, and even encourage sponsorship investment with data-backed predictions taking the sport marketplace and the game itself to another level. Although, even with such rich data available, it is still very complex to analyze and predict a game.

In May 2018, the Supreme Court made a decision to strike down Professional and Amateur Sports Protection Act of 1992 which led to several states legalizing sports betting. According to an article published in nba.com, 32 states would likely offer sports betting in the next five years. In order to capitalize on this new accessibility to gambling, bettors and betting organizations can leverage the knowledge of sports analysis to place smarter bets.

II. Related Work

There are many papers dealing with the topic of predicting the outcome of a sports game. A variety of techniques are used, ranging from data mining to statistical methods. But majority of the work in this field is statistical in nature, with some of it developed in blog posts or web columns [1]. Many of the statistical methods also offer themselves up as machine learning settings, with the expected gain that the burden of specifying the particulars of the model shifts from statistics to algorithm.

Some previous works in sports games (not only basketball) outcome prediction include [2] which provided statistical data as input to artificial neural networks to predict the outcome of soccer games, [3] in which Naive Bayes classifiers were used to predict the Cy Young Award winners in American Baseball and successfully predicted 80% of the prize winners. Also, authors in [4] proposed a system which did a fuzzy evaluation on statistical data to evaluate performance of players and predicted their performance in future games. A solution for supporting the basketball coaches in making tactical decisions during matches and also for pre-game and post-game analysis is also described in [5]

We intend to add to the body of work on sports analytics in the Machine Learning community by building on earlier works, such as NBA Oracle [6] and Predictions of NBA Games by Torres [7], and evaluating different learning settings and classifiers.

[6] uses Machine Learning methods to predict the outcome of an NBA match. The authors used four standard binary classification techniques: i) *Linear Regression*, ii) *Support Vector Machines*, iii) *Logistic Regression*, iv) *Artificial Neural Networks* on the data set acquired from DatabaseBasketball [8]. NBA Oracle's overall accuracy was 70% and in some cases, achieved an accuracy of 73%. The authors also tried to predict optimal player positions with the help of unsupervised learning methods such as K-means Clustering and even tried to predict outstanding players by using Outlier Detection.

[7] also uses Machine Learning methods to achieve a good prediction rate. The prediction only defined the winning team in the NBA match, regardless of the score. The author tried to detect the most important features that helped determine the

winner. These features could later be used to take important decisions in a match. Also, the author tried to predict the position of one player based on his features. The author tested his Machine Learning algorithms on the PCA (Principal Component Analysis) transformed data (taken from the Basketball Reference Website [9]). Accuracies of 67.89%, 66.81%, and 68.44% by using Linear Regression, Maximum Likelihood Classifier and Multi-Layer Perceptron respectively.

Overall game outcome prediction is a hard problem, due to various sources of randomness, such as player injuries, player attitudes, team rivalries, subjective officiating, and other non-deterministic factors.

We have used Logistic Regression, Support Vector Machines, Random Forest Classifier, Neural Networks and Adao-Bost Classifier on our data set to compare their performance with previous works.

III. Methodology

The process of developing our machine learning algorithms to test their accuracy consists of three major steps.

We got our data from the NBA stats website [10] where the data is explained in detail. We used a web scraper to download our data from the website. We had to study and select our data and we only chose the Regular Season data.

To prepare our data set, we spent a significant amount of time conditioning our data through cross-validation checks, normalization and feature reduction. Although there is a lot of rich data available, we wanted to select the most important features by performing feature dimension reduction. After the data is processed, we performed an analysis to select the best inputs for our model methods.

Finally, the Machine Learning algorithms were implemented using the pre-processed data set. The following techniques were implemented:

- Logistic Regression: It is a discriminative classifier that shares some similarities with linear regression, but uses a sigmoid function instead of a linear one.
- Support Vector Machines (SVM): Support Vector Machines are a way to build a classifier that maximizes the margin, using the training points closest to the classification boundary. These training points are known as support vectors. This allows the SVM to focus on those training data that actually define the decision boundary, and ignore those data that lie far from the boundary. SVM uses a kernel to map the input features to a higher dimension, which is used to make the data linearly separable.
- Neural Networks: An artificial neural network is a learning method based on the interconnection of actual biological neurons in our brain. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. It is an adaptive system that updates its structure (updates weights) based on information flow during network training phase.
- Random Forest Classifier: [11] The Random Forest Classifier is an ensemble algorithm. Ensembled algorithms are those which combines more than one algorithm of same of different kind for classifying objects. It creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the object.
- AdaBoost Classifier: Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996 [12]. It combines multiple classifiers to increase the accuracy of the classifiers. AdaBoost is an iterative ensemble classifier and the basic concept behind this is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observation.

IV. Data

We used the data from [10] in our machine learning models. We performed the following activities to get our data and prepare it.

A. Data Collection

We used a web scraper called *nba_py* to get our data. It is a python library that can be installed and imported as any other library for use. Using this, we could select what kind of data we needed and created our data sets.They initially consisted of player, team and summary information of all matches played in a season, we used this information to develop more complex features which we felt better captured the correlation between team performance and match outcome.

B. Data Preprocessing

We spent a significant amount of time preprocessing the data to use it in our machine learning models. After obtaining the data through the web scraper, we used the existing features in the data set to create complex features:

- Offensive Rating: Offensive rating for teams is defined as the points scored per 100 possessions[13]. It was calculated using the following formula -

$$Poss = 0.96 * (FGA + TO + 0.44 * (FTA) - OREB)$$

$$Off_{rtg} = (PTS/Poss) * 100$$

  where $Poss$ is the possessions, FGA is the number of field goals attempted, TO is the number of turnovers in the game, FTA is the number of free throws attempted, $OREB$ is the number of offensive rebounds and PTS is the number of points scored.

- Defensive Rating: Defensive rating is defined as the number of points the opponent was allowed to score per 100 possessions. It can be calculated as

$$Def_{rtg} = (Opps' PtsAllowed/Opps' Poss) * 100$$

  this value can be obtained by simply adopting the offensive rating of the opponent as the defensive rating of the current team.

- Net Rating: Net Rating for a game is calculated by simply subtracting the team's Def_{rtg} from their Off_{rtg} for a game. Net Rating for game i is calculated as -

$$Net^i_{rtg} = Off^i_{rtg} - Def^i_{rtg}$$

- Relevant Net Rating: The relevant net rating is a complex feature calculated as the average of the Net_{rtg} for the last game both teams played a common opponent and the last 5 games each team played. Relevant Net Rating for a game k for team A is -

$$Last5Games = \sum_{i=k-1}^{k-6} Net^i_{rtg}$$

$$RelNet^i_A = (Last5Games + Net^{commopp}_{rtg})/6$$

where $Last5Games$ is the sum of the net ratings for team A in the last 5 games they played and $Net^{commopp}_{rtg}$ is the net rating of team A the last time they played the most recent common opponent they share with their current opponent (say, team B).
The most recent common opponent is chosen by iterating through the teams faced by team A and their opponent, team B in the recent past and finding the opponent which was faced most recently with respect to both teams. This is done by calculating "$total - distance$" between a common opponent of A and B. The $total - distance$ is simply the sum of the number of days since the common opponent faced team A ($distance_a$) and the number of days since the common opponent faces team B ($distance_b$). We pick the opponent and respective games with the lowest $total - distance$ value. We pick the lowest value to capture the most relevant match ups for the two teams.

- Win Percentage: It is the percentage of games won by the team in the last 15 games played. This is a tune-able parameter which we believe captures streaks for team playing well, team momentum and can also account for if significant players in the team were injured/suspended (would be seen with a low win percentage). We chose a temporal window of 15 for this parameter because we believe that it effectively captures streaks of good/poor play while being stable enough to not be too over reactive.

- Days since Last Game Played: This feature is the number of days since the team played their last game, this helps us account for teams that may be playing back-to-back games which is not an uncommon occurrence in the NBA and can adversely affect performance of the team.

- Expert Rankings: These were compiled from multiple sports media websites that periodically publish power rankings during a season (ESPN.com, SI.com, etc.). A power ranking is an ordering of the teams from best to worst in the experts opinion. For our models we collected preseason power rankings for each season between 2014 and the present.

V. Training

We used k-fold cross validation to tune the hyper parameters for our classifiers. We used p-values to greedily do backward selection of features to reduce the size of our feature space. This improved the performance of some of our models significantly (Logistic Regression improved 5-7% in some cases).
For the neural network, we train over 500 epochs, using the Adam optimizer minimizing the mean squared logarithmic error at a learning rate of 0.01. This loss metric was chosen empirically, it provides significantly better results than other loss functions considered; mean squared error, mean absolute error, mean absolute percentage error and logcosh. The neural net was also unique in the fact that it did not directly try to predict the outcome of the matches, instead it tries to predict the scores of both teams and the outcome of the match is simply inferred from this estimation.

VI. Results

We calculate accuracy of binary classification (Does the home team win or not?) over the test set in three different settings for all our models. The results vary significantly based on what kind of scenario we were evaluating our models in. We argue the relative importance of one type of result over the other in the later sections.

A. Cross Validation (R1)

This setting involves splitting up the data set into 10 stratified K-folds, where each fold takes a turn being the test data and the others being the training data. The accuracy calculated is the average over all folds, this allows us to eliminate some of the bias of some sections of the data being more easy to predict while others may be harder.
We used a 10-fold cross validation to evaluate the performance of all our models - this results in a 90-10 split of the training and test data respectively.Our models did very well in this setting, with the best performer, Random Forest managing a test accuracy of 0.75, and other classifiers doing similarly well with test accuracy scores ranging from $0.64 - 0.7$ as seen in Fig. 1.
We do not believe that this setting is the most appropriate one, it is an unrealistic use of our model to predict matches that have already occurred. It is reasonable to infer that models would perform better in this setting because it is likely to have been trained on more relevant tuples which aid in predicting the outcome of certain matches.

B. Predicting Half a Future Season (R2)

In this setting, we use data from the 2016-2017 season and the first half of the 2017-2018 season to predict the outcome of the matches in the second half of the 2017-2018 season.

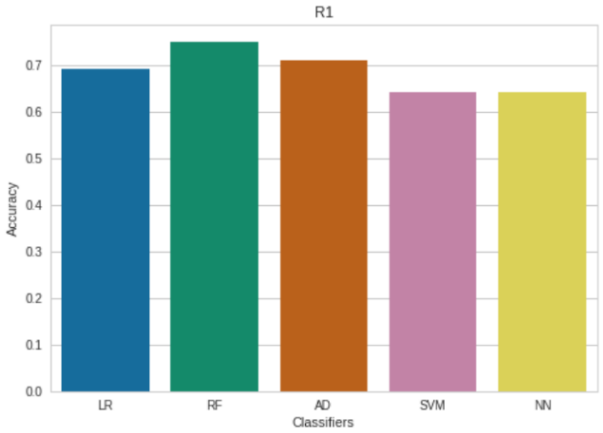

Fig. 1. Accuracy scores for all models during stratified cross validation

This means that all match data from the 2016-2017 season and the first half of the 2017-2018 season are trained on by the models and they are tested on the match outcomes of the latter half of the 2017-2018 season.
As expected, the accuracy scores of our models were lower at this future prediction than the cross-validation. The scores were still very good, with our best performer, Random Forest achieving a test accuracy of 0.67, followed by others which consistently did well in this prediction as seen in Fig. 2.
We believe this setting is more suited to our application but falls short in that even though there are experts who predict entire season results or multiple results, they tend to publish new prediction frequently - updating their prediction using newly available information. So it is natural that this setting would not perform very well, many valuable data points which could aid in prediction are not being used during training.

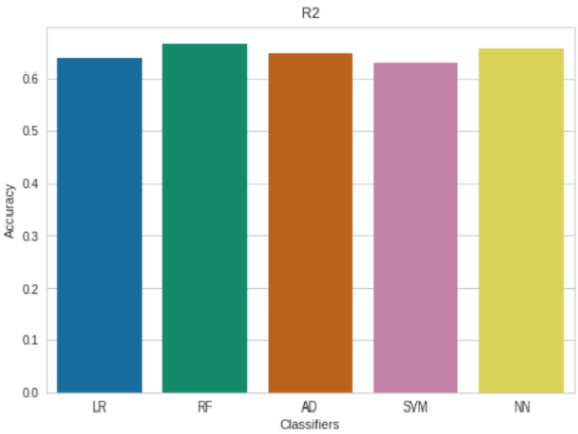

Fig. 2. Accuracy scores for all models attempting to predict the second half of the 2016-2017 season all at once

C. Predicting all games in a Single Day (R3)

Here we emulate a situation in which we predict every day of the 2017-2018 NBA season in chronological order using the most up to date feature values for each day. We retrain our models before predicting the outcomes of each day with data from all the days prior going back to the beginning of the 2016-2017 NBA season. We believe that this is a more valid way to evaluate the effectiveness of our models as this is how they would be put to use in a real situation.
This setting avoids the shortcomings of R2 by leveraging as much data as possible to make better predictions on a day-by-day basis. The accuracy score is calculated for every day and weighted by number of games played that day and averaged over different time periods; full season, two-thirds of a season, one-third of a season. We see our best performer in this setting being AdaBoost at 0.69.
There are usually 4 or 5 games a day, this means that predictions on these days can have a very high volatility in their results as can be seen in Fig. 3. This is simply because there are days where we have perfect solutions and others where we do not, since it is such a small number of games, the variance is to be expected. We do see a trend in the accuracy scores, with better scores generally in the middle and having a slight taper off around the end of the season. This can be observed in Fig. 4. We believe this is because more of the season's trends are captured by the middle of the season, but by the end one of our most valuable features like preseason expert rankings may become outdated and start working against us.

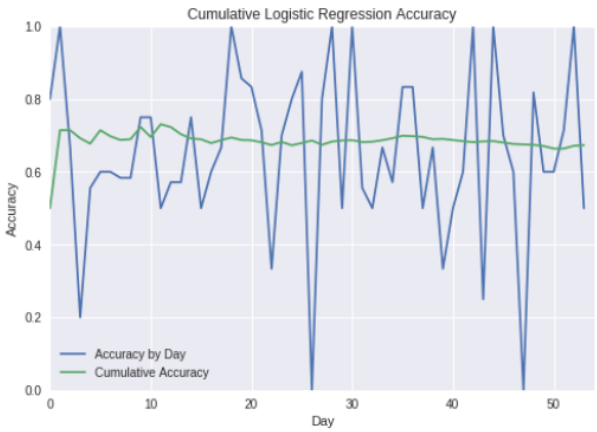

Fig. 3. Accuracy score by day and cumulative accuracy for our Logistic Regression model on the final third of the 2016-2017 NBA season.

Looking through the feature importance for each model, we see that while different models value the features differently, the most common features to see on the top are expert rankings and $RelevantNet_{rtg}$. This provides us some validation that the features we designed were successfully able to capture the nuances of the data required for predicting the outcome of a match. Among the one-hot encoded team

names we noticed a trend that teams that tend to win or lose a lot are more likely to be weighted higher importance than those that have a fairly even split of wins and losses. These feature importances can be seen in Fig. 5.

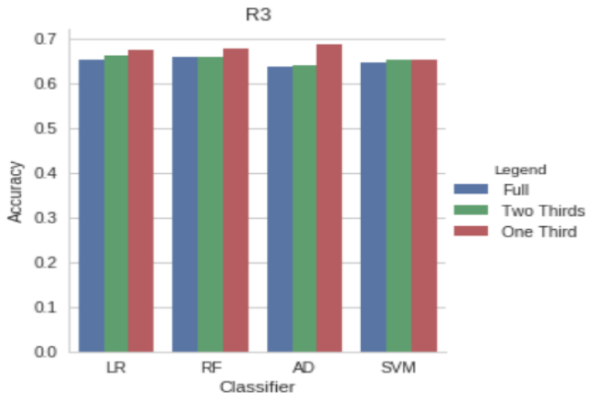

Fig. 4. Accuracy scores for all models segmented by portion of season they are attempting to predict

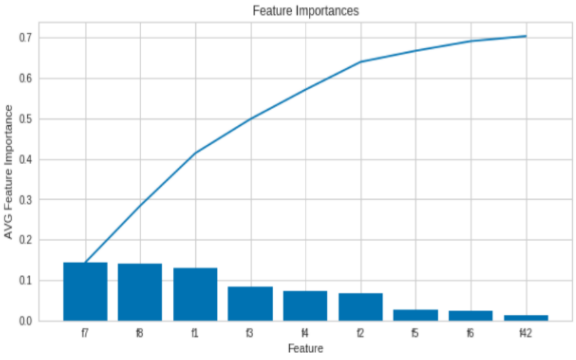

Fig. 5. Graph showing feature importance values from our Random Forest model. The features displayed from left to right are as follows: Expert rank for home team, expert rank for away team, relevant net rating for home team, win percentage for home team, win percentage for home team, relevant net rating for away team, days since last game played by home team and days since last game played by away team

Work Distribution

- **Kevin Matthew** - Handled tuning model hyper-parameters, model evaluation, and all things related to our neural net model.
- **Neil Gibeaut** - Wrote scripts for data collection and feature derivation. Also assisted with model tuning, feature selection, and evaluation.
- **Nishant Barma** - Directed our background research, created figures of end results, and assisted with feature selection methods.

Conclusions

Having compared our model to baseline of experts who on average have accuracy scores of around $0.68 - 0.71$, we can see that while it is not easy to outperform experts it is possible to get comparable results. The model is very susceptible to becoming obsolete as the state of the league can change quite rapidly with changing teams, players, rules and strategies. This kind of work has the potential to not only aid in a gambling situation, but also with team strategies and providing a basis for analyzing odds. The models as they are now struggle to perform well on the games at the start of the season.

We believe we have created a sophisticated and challenging project which performs quite well. There is definitely room for improvement but we believe this was a great first step into the world of predictive models for sports.

Future Work

We believe that there is a large amount that could still be done to improve the predictive accuracy of our models. More experimentation in regards to the tune-able temporal windows of our features such as relevant net rating, and win percentage could be done to optimize results. Also, new features which capture information about player availability or distance traveled would likely be beneficial to our models.

A large area of potential improvement we have identified in our methods is collecting more up to date expert rankings. Currently we are only utilizing expert rankings from before each season begins. By the end of the season, there are typically multiple teams that have exceeded or fallen below the expectations set for them. By continuing to use expert opinions from the beginning of the season to predict games occurring at the end of a season, we are not providing our model with the most up to date information and likely sacrificing performance. Despite this shortcoming of our methodology, expert ranks was still the feature that provided us with the largest jump in accuracy, so by improving this we would expect to see an even large accuracy gain.

We would also like to expand the usefulness of our models by training them to predict more than just the outcome of a game. For example, point spreads, or player stats a are two potential candidates that would be useful to anyone attempting to gamble on the NBA or set their fantasy basketball lineup.

More work could also be done to improve our testing framework. Currently it is difficult to quickly test the performance of different combinations of our tune-able feature parameters(N for N games, etc.). A more API-like implementation or even a graphical user interface that allows a user to more easily experiment with these parameters would greatly increase the usefulness of our models.

References

[1] Miljkovi, D., Gaji, L., Kovaevi, A., Konjovi, Z. (2010, September). The use of data mining for basketball matches outcomes prediction. In Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on (pp. 309-312). IEEE.

[2] McCabe, A., Trevathan, J. (2008, April). Artificial intelligence in sports prediction. In Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on (pp. 1194-1197). IEEE.

[3] Smith, L., Lipscomb, B., Simkins, A. (2007). Data mining in sports: Predicting cy young award winners. Journal of Computing Sciences in Colleges, 22(4), 115-121.
[4] Atlas, M., Zhang, Y. Q. (2004, September). Fuzzy Neural Agents for Online NBA Scouting. In Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (pp. 58-63). IEEE Computer Society.
[5] Polese, G., Troiano, M., Tortora, G. (2002, July). A data mining based system supporting tactical decisions. In Proceedings of the 14th international conference on Software engineering and knowledge engineering (pp. 681-684). ACM.
[6] Beckler, M., Wang, H., Papamichael, M. (2013). NBA oracle. Zuletzt besucht am, 17(20082009.9).
[7] Torres, R. A. (2013). Prediction of NBA games based on Machine Learning Methods. University of Wisconsin, Madison.
[8] http://www.databasebasketball.com
[9] http://www.basketball-reference.com
[10] http://stats.nba.com
[11] https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1
[12] https://www.datacamp.com/community/tutorials/adaboost-classifier-python
[13] Dean Oliver, Basketball on Paper: Rules and Tools for Performance Analysis