Predicting Outcomes of NBA Playoff Games
CIS 419/519: Applied Machine Learning

Jediah Katz JEDIAHK@SEAS.UPENN.EDU
Varun Ramakrishnan VARUNR99@SEAS.UPENN.EDU
Ishaan Rao ISHAANR@SEAS.UPENN.EDU

Abstract

Sports betting has recently become legalized in several states across the US, leading to a growing interest in being able to predict the outcomes of games. We aim to develop a machine learning model that will be able to predict the winner of an NBA playoff game between two given teams. As a result, bettors may be able to also make more informed bets on the spreads of NBA games. In addition to working with individual games, we can use this model to determine the winner of the NBA playoffs, which start with 16 teams and culminates with one winner. To predict playoff performance of NBA teams, we will use statistics describing a team's overall regular season performance as features for the model.

1. Data Selection

We will first be attempting to learn a binary-valued function, which represents either a win or a loss. We will also potentially attempt to learn an integer-value function to interpret a positive margin of victory to signify that the first team wins, and a negative margin to signify that the second team wins. Part of the challenge involved in this problem is the vast number of potential features from NBA games. We approached the data scraping process by deciding to scrape all data for a team's regular season performance. This included their record, simple stats (such as points, rebounds, steals, blocks, assists, and turnovers), as well as advanced metrics (such as turnover rate, field goal attempt rate, etc.). We scraped our data from the Basketball Reference, a site containing official NBA data on games from every NBA season in history (Sports Reference, 2019). We will be web scraping the site to obtain the features we desire for our model. We also made a decision to only scrape data for NBA seasons from 1980 to the present, because 1980 was the first year that the NBA began keeping track of many of these advanced stats. Additionally, 1980 occurs four years after the NBA-ABA merger, which led to the existence of most of the teams we see playing today. By making this decision, we can ensure we have consistent data across all instances, as well as a strong representation of the current NBA landscape, so our model will be well equipped to predict future NBA games.

University of Pennsylvania, CIS 419/519 Course Project.

2. Cleaning Data

Our initial dataset is comprised of 2894 rows and 174 columns. Each row is comprised of a playoff game from 1980 to 2018. For example, if we want to represent Game 1 of the 2018 NBA Finals between the Golden State Warriors and the Cleveland Cavaliers, the row will contain a description of the game (i.e. Game 1), its date, both team names, and associated team statistics from the regular season for each. Our first process in cleaning the data was to generalize the set of NBA teams. Over the years, teams have moved locations and changed names, leading us to a variety of different teams and names. We made the following merges: New Jersey Nets to Brooklyn Nets, Seattle Supersonics to OKC Thunder, New Orleans Hornets to New Orleans Pelicans, Charlotte Bobcats to Charlotte Hornets, Kansas City Kings to Sacramento Kings, and lastly Washington Bullets to Washington Wizards. In terms of imputing missing values, since we already had a limited dataset, we decided against deleting any rows with missing values. Additionally, we saw that imputing arbitrary numbers such as 0, the maximum value of that feature, or the minimum value of that feature, would be detrimental to the model and would not make sense in the context of NBA statistics. This is because the expected range of certain statistics have changed as the NBA has evolved. Currently, the NBA is very offensively focused, meaning offensive statistics will be significantly higher than before. Since the teams that make the playoffs are considered to be the top teams in the league, their regular performances are relatively similar. So, we decided to impute any missing values with the mean value of that feature, in order to give a general estimate of regular season performance for a certain team.

3. Initial Model and Classifier Selection

The first problem we decided to tackle was to predict the winner of a playoff game. We cleaned the dataset as described above and experimented with various classifiers to train an initial model for this problem (Loeffelholz et al., 2009). As shown in the table, we experimented with 6 classifiers, which are shown below. We found these values by conducting a 10-fold cross validation on the dataset and finding the average values for these scoring metrics. These initial values are derived from training the default classifiers without any hyperparameter tuning.

CLASSIFIER	ACCURACY (%)	F1 (%)
ADABOOST	62.96	72.68
DECISION TREES	55.01	63.85
EXTRA TREES	64.03	74.58
GRADIENT BOOSTING	64.55	74.31
MULTI-LAYER PERCEPTRON	66.76	76.84
RANDOM FOREST	64.48	74.87

Table 1. Accuracies and F1 scores for Various Classifiers

We clearly see that the Multi-Layer Perceptron Classifier (MLPClassifier) performed better than all other classifiers, as it has both the highest accuracy and the highest F1 score. Thus, we decided to proceed with our model by using this classifier for our predictions.

4. Feature Selection and Hyperparameter Tuning

Now that we have chosen a classifier, we will attempt to optimize the classifier as much as possible. First, we tuned various hyperparameters, such as the size of the hidden layer (i.e. number of neurons), the maximum number of iterations, and the activation functions. Our results from experimenting with these various hyperparameters are outlined in the tables below. These values were obtained by conducting a 10-fold cross validation on the dataset and finding the average values for these scoring metrics.

HIDDEN LAYER SIZE	ACCURACY (%)	F1 (%)
5	67.24	77.22
10	67.35	77.18
100	67.01	77.23
500	66.11	76.34
1000	66.55	77.32

Table 2. Accuracies and F1 scores for Hidden Layer Sizes

MAXIMUM ITERATIONS	ACCURACY (%)	F1 (%)
500	66.76	76.77
1000	66.97	77.03
2000	66.76	76.51
5000	66.52	76.16
10000	66.83	76.64

Table 3. Accuracies and F1 scores for Maximum Iterations

ACTIVATION FUNCTIONS	ACCURACY (%)	F1 (%)
IDENTITY	61.89	67.36
LOGISTIC	66.76	76.84
RELU	61.96	70.77
TANH	64.37	73.09

Table 4. Accuracies and F1 scores for Activation Functions

We see that we obtained the highest performance with a hidden layer size of 5 neurons, 1000 maximum iterations, and a logistic activation function. Using these three tuned hyperparameter values, our model reached an accuracy of 67.55%. Currently, certain expert models can predict the outcome of general regular season NBA games with an accuracies of 70.3% (ESPN AccuScore) or 70.6% (Vegas Betting Line) (Cheng et al., 2013). We aim to create a model that can perform better than this model. Our tuned model is being trained on a dataset with 169 features and playoff games from the past 40 years. We thought that we may be training our model with too many features, so we aimed to improve our accuracy by reducing the feature set (Jones, 2016). First, we conducted a Principal Component Analysis (PCA) with 25 components on our data to reduce the dimensionality of our feature space. With these results, we were able to identify which features contributed the most to the principal components in the PCA.

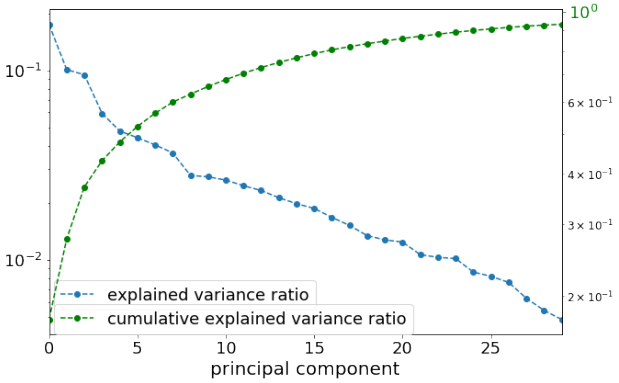

Figure 1. Explained Variance from PCA

The above plot of explained variance from each principal

component seems to indicate that our problem is difficult to learn, or at least that we were missing some features that would allow us to easily predict the winner of a game. The plot was quite smooth, with no clear "elbow". The nearest thing to an "elbow" appears around 5 components, but at this point the cumulative explained variance ratio is only about 0.5, and successive components have a small explained variance.

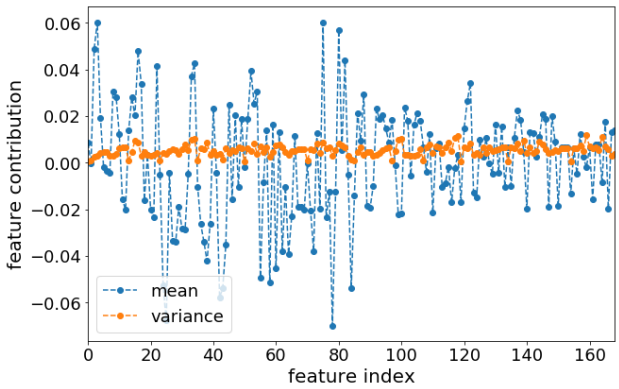

Figure 2. Feature Contribution from PCA

The above figure shows each feature's mean and variance of their contributions. We sorted our feature set in descending order of absolute contribution and experimented with using a select number of these features. We found our model reached the highest accuracy when using approximately the top 95 features. After conducting feature selection, our model's accuracy peaked around 69-70%, which was still slightly worse than the expert models mentioned above. Additionally, in the past few decades the NBAs offensive play style has dramatically changed, as the 3 point shot has become increasingly popular. This has led to higher point totals, pace of play, and 3 point attempts. Therefore, we decided to use training instances from the 2000 season onwards, rather than the 1980 season, because we believe this will better predict future NBA games. After altering our dataset to only includes games occurring after 2000, our tuned MLPClassifier reached a peak performance of approximately 75% accuracy over a 10-fold cross validation. We now have a model that can predict the winner of an NBA game better than some prominent models.

5. Extension: MOV Prediction

As an extension to our results, we attempted to generalize our work to predict not just the winner of a game, but the margin of victory (MOV); i.e., a new feature defined as the difference between the home team's score and the away team's score. (Cheng et al., 2013)

To solve this regression problem, we experimented with neural networks and many of *scikit-learn*'s generalized linear models, but ultimately settled on Support Vector Regression, a generalization of Support Vector Machines to solving regression problems. We also settled on using 20 features with the highest absolute contribution from PCA, since experimenting with larger numbers of features did not yield any significant improvement.

Here are the results of our training using 10-fold cross validation with a variety of SVR kernels. We calculated the mean absolute error between the predicted MOV and the true MOV.

SVR Kernel	Mean Absolute Error
Linear	9.64
Poly ($d = 3$)	9.82
RBF	9.80
Sigmoid	11.52

Table 5. Mean absolute error for various kernels

After settling on a linear kernel, we performed some experimentation with the regularization parameter C:

C	Mean Absolute Error
0.01	9.90
0.10	9.65
1.00	9.64
10.0	9.66
100.	9.79

Table 6. Mean absolute error for values of C with linear kernel

Our best mean absolute error of 9.64 is not particularly remarkable, since NBA games are often decided with a margin of victory of less than 10 points, so our model would probably not be useful towards actual sports betting. However, the publicly released work on this problem that we were able to find was similarly unsuccessful (Cheng et al., 2013), indicating that a model to predict the margin of victory is difficult to learn.

6. Case Study: 2018 NBA Playoffs

As mentioned before, the main aim of our project is to use our models to predict the outcome of NBA playoff games. Thus, we used our final models in order to predict the results of the 2018 NBA Playoffs. All 30 teams in the NBA are grouped into two conferences based on their geographic location. After the regular season is finished, the top 8 teams in both the Eastern Conference and Western Conference make it to the playoffs. They are seeded based on

the number of wins in the regular and are arranged within the playoff bracket as shown in Figures 3 and 4. Within each matchup, the two teams play a best-of-7 game series to determine who moves on to the next round.

As for the data, we already had the data for all the first round matchups, since these games actually occurred. However, things got a little more tricky in future rounds. If we correctly picked a matchup within the second round and onwards, we could simply use the data we scraped. However, if we predicted a matchup then never actually occurred, we just spliced together the data from two separate matchups into one, since the data didn't depend on who exactly the team played. After preparing all the data, we used our final models to predict only the binary outcome (win/loss) of every game. Figure 3 shows our predicted bracket, while Figure 4 contains the actual results.

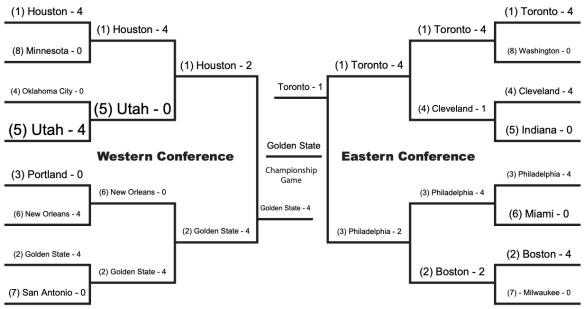

Figure 3. Predicted 2018 NBA Playoffs outcomes

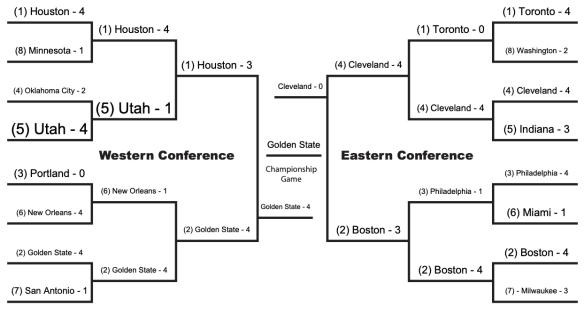

Figure 4. Actual 2018 NBA Playoffs outcomes

Comparing the actual and predicted series winners, we see that our model correctly predicted almost every series correctly. Within the Eastern Conference, we see that our model incorrectly predicted both the winners for the semifinal rounds. However, it still predicted that the Warriors would win the finals, which they did. We can attribute this discrepancy to a phenomenon we have called the "LeBron Factor." Last year, the Cleveland Cavaliers had LeBron James, a player whom many regard as the greatest NBA player of all time. Throughout his career, James has been known to carry mediocre teams deep into the playoffs, often to the NBA Finals. Thus, while the team's overall statistics are extremely average, they are still able to win many games in the postseason. Our model was unable to account for this phenomenon and thus incorrectly predicted the Cavaliers to lose in the Eastern Conference Semifinals.

Another issue our model had was predicting the correct outcomes for each individual game. For example, it predicted that every series in the first round was a 4-0 sweep. In reality, this situation is not very common, and traditionally only occurs in matchups involving the 1st and 8th seeds. However, in future rounds, where teams were more evenly matched, the outcomes were far more realistic. These results suggest that our model has its best performance with evenly-matched teams, while it tends to overfit with high-powered teams and almost always predicts them to win in uneven matchups. Overall, we had a series-win accuracy of 80%.

7. Important Applications

The most important application of this project is in sports betting, which directly involves predicting margins of victory between teams. In addition, this type of work would be extremely beneficial for sports teams in that they could forecast their performances. By doing so, they can change their training methods and improve their performances against teams with specific styles of play. Finally, this type of data would be useful for TV networks who broadcast NBA games. Because they seek to maximize revenue with each broadcast, networks hope to select games that they believe would be competitive. Using this model would give them the games with the smallest spread, thus enabling them to select the most competitive and exciting games for prime time.

8. Conclusion

As seen from our results above, using the multi-layer perceptron neural network is able to predict the winners of NBA playoff games with accuracies of approximately 75%, which exceed those obtained by popular models, such as those made by ESPN and FiveThirtyEight. The NBA Playoffs have always been considered to be much more variable than the NBA regular season, so our usage of regular season data to predict playoff games with a high accuracy is particularly notable. We are currently building upon our model to develop a new model that will predict the margin of victory of these NBA games and plan to work on this model in the future.

References

Cheng, B., Dade, K., Lipman, M., and Mills, C. Predicting the betting line in nba games. 2013.

Jones, E. S. *Predicting Outcomes of NBA Basketball Games*. PhD thesis, North Dakota State University of Agriculture and Applied Science, 2016.

Loeffelholz, B., Bednar, E., and Bauer, K. Predicting nba games using neural networks. *Journal of Quantitative Analysis in Sports*, 5(1), 2009.

Sports Reference, LLC. *Basketball Reference*, 2019. Data retrieved from https://www.basketball-reference.com/.