Using Machine Learning Techniques for Predicting the Winner of NBA Games

Stephen Feddes, Jahi Stewart, Zechariah Rathe, Rebekah Downer

Lewis University Romeoville, United States

{stephenpfeddes, jahikstewart, zechariahdrathe, rebekahgmyrick}@lewisu.edu

*Abstract*—With the increasing popularity of fantasy sports, accurately predicting the winner of an NBA game is essential for making well-informed predictions and providing a better experience. In this paper, we propose using machine learning techniques to predict the winner of an NBA game. In particular, we offer logistic regression and Gaussian Naïve Bayes to predict the winner of an NBA game accurately. In addition, we will be using the Elo rating difference between home and away teams to predict the winner of an NBA game. Furthermore, we will use a synthetic minority oversampling technique to address the class imbalance.

*Index Terms*—logistic regression, Gaussian Naïve Bayes, API, synthetic minority oversampling, Elo rating

# Introduction

Many fans predict the winner of sports games from a gut feeling. Predicting winners from gut feeling has the benefit of intuition and fast decision-making but can suffer from bias and inaccuracy. Taking the opposite approach of analyzing a large dataset of all the advanced statistics for your favorite team and the opponent can be tedious, inefficient, error-prone, and time-consuming. With this in mind, we propose a dynamic, simple, and accurate model.

To make our model dynamic, we integrated it with the balldontlie API [1], which provides data from historical NBA. We are fetching data on all NBA games from 2014 to the present day through the balldontelie API to train the model. Fetching data from an API provides faster, more reliable data retrieval when compared to techniques like web-scraping.

To simplify our model, we combined several features into one feature through a relatively short mathematical formula for calculating the Elo ratings and then getting the difference in the Elo ratings between the home and away teams. Because logistic regression and Gaussian Naïve Bayes are probabilistic models, users can provide the Elo rating difference for a game they are interested in and can be returned a probability for the home team winning.

Utilizing the Elo rating differential feature, we achieved an accuracy of 64% over a 10-fold cross-validation with our Gaussian Naïve Bayes and Logistic Regression models. This accuracy is significant because the outcomes of games can be unpredictable. During the NBA regular season, the upset rate is around 32%, meaning that the non-favored team wins 32% of the time. As such, most models predicting NBA winners achieve an upper bound of accuracy between 66% and 72% [2].

Our model is meaningful with reliable, fast data retrieval through an API, a singular, simple feature, ease of interpretation, and significant accuracy.

# Related Work

Researchers have proposed many methods and techniques to predict the winners of NBA games. Researchers have considered features related to season-long performance, recent team performance, and player performance.

The most widely regarded model, in [3], is the FiveThirtyEight model, which uses team Elo ratings and its own RAPTOR ratings for player performance to predict games with over 70% accuracy.

There are also other notable models. For example, in [2], the author proposes a random forest model with features like Elo ratings, recent team performance, and recent player performance. Elo ratings are a relativistic rating system standard in zero-sum games like chess. It quantifies a competitor’s strength based on the strength of the competition they have beaten. Elo ratings relativize a competitor’s competence. Defeating weaker teams does little to affect one’s Elo rating, but beating superior teams by a considerable margin will. Likewise, losing to higher-rated teams slightly impacts one’s Elo rating. Notably, the author found that a team’s recent statistics and Elo rating were more effective at predicting the winner of NBA games than player performance. They tried using recent player performance and linear regression to predict the scores of the home and away teams and, therefore, predict winners. This model was 58.66% accurate, much lower than their random forest model that utilized Elo ratings and recent team performance in different metrics. That model was 67% accurate.

In [4], the author also utilized Elo ratings, team statistics, and recent team performance to predict winners. He tried models like logistic regression, random forest, and K-nearest neighbors. He found Gaussian Naïve Bayes to be the most accurate, averaging 65.1%.

Finally, [5] had a complex but accurate model. For example, concerning the player statistics alone, they used 30 features. For the team statistics, they had detailed features, such as if the game was back-to-back. Interestingly, they utilized a convolutional neural network trained on 12 NBA seasons. Their model was 71.76% accurate on unseen seasons. It is unclear if they included the playoffs in their dataset, as playoff winner predictions are generally more precise than regular season games [2].

Most of the existing models utilize many complex features. The model we propose has competitive accuracy compared to the others, good f1 scores, a singular feature that is readily derivable, and interpretable models that can retrieve their training dataset from an API.

# Dataset and Preliminary Results

In this work, we will compare two models that classify the data as a win or loss for the home team. The first approach is to utilize a Gaussian Naïve Bayes model. It is a classification technique used in machine learning based on a probabilistic approach and Gaussian distribution. It assumes the features are independent [6]. The second is the logistic regression model. The results of these two methods are presented in this section. In section IV, we explain the contribution of this work in more detail.

## The Dataset

### The dataset in this work is from the balldontlie API [1]. It consists of 10,750 NBA games from the 2014-15 to 2022-23 seasons with several attributes for each match: team ID, team abbreviation, date, season, home team final score, and away team final score. The dataset is imbalanced, meaning there are more games where the home team won than games where the home team lost, as shown in Fig. 1. The home team won approximately 57% of the games, and the away team won approximately 43%. Thus, although accuracy above 50% is significant, our model should be more than 57% accurate. Initially, the dataset did not have a column indicating whether the home team won or not. In the preprocessing stage, we utilized the final scores from home and away teams to deduce the winner of each game and create a new column that takes one of two values: 1 for home wins and 0 for home losses.

A red and blue squares

Description automatically generated

1. The bar plot of the two classes in the dataset: home win and home loss.

## Gaussian Naïve Bayes Model

Our Gaussian Naïve Bayes model utilizes the Elo rating difference between the home and away teams at

A red and blue dots

Description automatically generated

1. Scatter plot of each game’s home team’s Elo rating and away team’s Elo rating.

the time of their matchup. Several preprocessing steps were taken to get the Elo ratings. These include:

* **Initialization:** In this step, we initialized all the Elo ratings for the teams in their first games during the 2013-14 season to 1500. For the training and testing sets, we decided not to include this season and instead start with the 2014-15 season so that the Elo ratings for that season would be more reflective of each team’s competence to begin that season.
* **Elo Rating Formula and Elo Season Carryover:**

To calculate the Elo rating of a team, we utilized the formula in (1), where k is a moving constant dependent on both the margin of victory and the difference in Elo ratings. It is defined in (2) by [7]. [7] provides their justification for it. Steam is a state variable that can be 0 or 1, and Eteam, defined in (3), is the expected win probability. [8] provides an in-depth justification for the general formula.

Ri+1 = k \* (Steam – Eteam + Ri) (1)

k = 20 (2)

Eteam = (3)

Because good teams tend to stay strong from season to season, as evident in Fig. 3, we needed to ensure a team's Elo rating from the previous season would carry over to the next season. Instead of simply letting the team's Elo rating for their first game in a season be the same as their finishing Elo rating from the previous season, we learned from [7] that it needed to be scaled by a specific factor to account for the tendency for teams to revert to the mean. (4) provides that formula, where R is equal to the final Elo rating a team had in the previous season.

elo\_season\_start = (R\*0.75) + (0.25\*1505) (4)

Our Gaussian Naïve Bayes model estimates the probability that the home team wins, given the Elo rating difference

A graph of a number of different colored lines

Description automatically generated with medium confidence

1. Histograms of the Elo rating differentials for the home loss class and home win class.

between the home and away teams. This model assumes the features are independent and normally distributed. Because we only have 1 feature, we can be assured that there is independence. From Fig. 3, we can be confident that Elo rating differentials are normally distributed for the classes. After feature-engineering the Elo rating differences between the home and away teams for each of the games, we trained and tested our model. The dataset was split into training and testing sets with a ratio of 80:20. Using 10-fold cross-validation, our model had an average accuracy of 65%, but had f1 scores of 54% for the home loss class and 72% for the home win class. We improved the f1 score for the home win class in Section IV.

## Logistic Regression

Logistic regression is a machine learning algorithm for binary classification. Logistic regression uses the sigmoid function, an S-shaped curve that can take any value between 0 and 1. It uses the log odds of a linear equation of inputs to classify the class that is most probable [9]. Using the Elo rating differential feature, the average accuracy in the 10-fold cross-validation for our logistic regression model was 65%. We had an f1 score of 54% for the home loss class and an f1 score of 72% for the home win class. In the next section, we will improve the f1 score for the home loss class.

# Proposed Method and Result

## Elo Rating Differential

As previously discussed, the dataset did not come with Elo ratings for the teams. So, in the preprocessing phase, we feature-engineered them. Motivated by [2], Fig 4, and Fig 2, we utilized the Elo rating differential, which is calculated by subtracting the away team’s Elo rating from the home team’s Elo rating. This produced the one feature our models use. With this feature, our models had comparable accuracy to [2] and [4].

## Synthetic Minority Oversampling

As mentioned earlier, the f1 scores for the home loss class in both models were poor. To fix this, we introduced synthetic minority oversampling from Python’s imblearn library. Synthetic minority oversampling balances the class distribution for training the model by generating synthetic samples from the minority class [9]. By utilizing this technique, the f1 scores for the two classes became roughly

A graph with different colored lines

Description automatically generated

1. Time series plot of the Warriors, Cavaliers, and Magic Elo ratings from the 2013-14 to 2022-23 NBA seasons.

equal at 64%, as evident in the confusion matrices and classification reports in Fig. 5. Although the accuracy decreased from 65% to 64%, it is more important for the f1 scores to be high. Because of class imbalance, f1 scores are a better indicator of a good model than accuracy.

## Testing On the Currently Unseen Season

To measure how well our model performs in predicting the outcomes of the current NBA season, which the model has yet to see, we compared the game outcomes to the outcomes the model predicted. Of the 307 games played so far, both models correctly predicted 63% of the outcomes. Both models had a f1 score of 60% for the home loss class and 66% for the home win class, as evident in Fig. 6.

# Conclusion

In this work, we introduced different machine learning techniques to predict the winner of NBA games by utilizing the Elo rating differences between the home and away teams. Without synthetic minority oversampling, the Gaussian Naïve Bayes model and logistic regression models were approximately 65% accurate on average, with poor f1 scores for the home loss class.

To improve the f1 score of the minority class, we utilized synthetic minority oversampling, which resulted in both models having f1 scores of 64% for both classes.

When comparing the predicted outcome of the games for the current NBA season, which the model has not been trained on, to the actual results, our model had an f1 score for the home loss class of 60% and an f1 score for the home win class of 66%.

Predicting the outcome of sports games is difficult due to their inherent unpredictability. Many factors influencing a game, such as refereeing, trades, luck, or clutch factor, are difficult or impossible to quantify and predict. Considering the best models are only around 70% accurate, the fact that our models have f1 scores of 64% for both home win and home loss classes by utilizing one feature is a significant feat.

##### References

1. https://www.balldontlie.io/home.html#attributes. [Date accessed Dec. 4, 2023]
2. J. Weiner. “Predicting the Outcome of NBA Games with Machine Learning”. Medium, Towards Data Science. https://towardsdatascience.com/predicting-the-outcome-of-nba-games-with-machine-learning-a810bb768f20 (accessed Dec. 7, 2023).

A screenshot of a graph

Description automatically generated A screenshot of a graph

Description automatically generated

1. Confusion matrices and classification reports for the two models performance on the testing set.

A blue squares with white text

Description automatically generated A screenshot of a graph

Description automatically generated

1. Confusion matrices and classification reports for the two models' performance on the current NBA season
2. N. Silver, J. Boice, N. Paine, R. Best, H. Fuong. “How Our NBA Predictions Work”. FiveThirtyEight. <https://fivethirtyeight.com/methodology/how-our-nba-predictions-work> (accessed Dec. 7, 2023).
3. M. Houde. “Predicting the Outcome of NBA Games”. Bryant University. https://digitalcommons.bryant.edu/cgi/viewcontent.cgi?article=1000&context=honors\_data\_science (accessed Dec. 7, 2023).
4. P. Marin. “Netty – My Personal NBA Game-Winner Predictor”. Medium. <https://medium.com/data-sports/netty-my-personal-nba-game-winner-predictor-63cc728025c5> (accessed Dec. 7, 2023)
5. C. Martins. “Gaussian Naïve Bayes Explained with SciKit-Learn”. Builtin. https://builtin.com/artificial-intelligence/gaussian-naive-bayes (accessed Dec. 7, 2023).
6. N. Silver, R. Fischer-Baum. “How we calculate NBA Elo Ratings”. FiveThirtyEight. https://fivethirtyeight.com/features/how-we-calculate-nba-elo-ratings (accessed Dec. 8, 2023).
7. “Elo Rating System”. Wikiwand. <https://www.wikiwand.com/en/Elo_rating_system> (accessed Dec. 7 ,2023).
8. J. Brownlee. “Logistic Regression for Machine Learning”. Machine Learning Mastery. https://machinelearningmastery.com/logistic-regression-for-machine-learning (accessed Dec. 7, 2023).
9. J. Brownlee. “SMOTE for Imbalanced Classification with Python”. Machine Learning Mastery. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification (accessed Dec. 8, 2023).
10. https://github.com/StephenFeddes/MachineLearningProject [Date accessed Dec. 8, 2023]

\*This is our project repository.