The Greatest Bet in Baseball

Six Simple Outs

Devin Basley, Zachary Strennen, Vinay Maruri, Daven Lagu

2024-04-30

Introduction

Sports betting is a practice that is both wildly popular and potentially very lucrative for sports fans, but most bettors lose more than they win. Part of the reason for these losses is that bettors place wagers impulsively, making uninformed decisions based on biased beliefs or incomplete knowledge of what bets will hit.

Baseball is an interesting sport from a betting perspective because each inning is essentially a mini game within the full 9 inning game, allowing bettors to place a variety of different bets on each inning, where each wager can have completely different odds or outcomes. One such bet is the NRFI (No Run First Inning) vs YRFI (Yes Run First Inning) bet that wagers either there will be no runs scored by either team in the first inning (NRFI), or at least one run will be scored by either team in the first inning (YRFI).

For this analysis, we will generate our own set of probabilities that a NRFI will happen in a given game. In order for our model to be profitable, we have to aim for at least 60% accuracy to obtain a consistent profit and to account for the different odds these wagers can be set at, generally ranging from -110 to -160. The goal of this analysis is to provide more informed decision making when placing these bets to improve the profitability when gambling.

Data

We pulled MLB Statcast from the 2023 and beginning of the 2024 season using the pybaseball library in python. We then filtered the data to contain only pitches and at-bats from the first inning into a new inning summary data frame and created pitcher and team features from the first inning Statcast data for modeling purposes. After filtering, we were left with 83,064 rows of data for the 2023 season and 9,952 rows of data for the ongoing 2024 season. and created pitcher and team features from the first inning Statcast data.

EDA

After filtering to end of at-bats events, we could see the overall distribution of run scoring for home teams and away teams. As we can see in Figure 1, run scoring appears to have a Poisson distribution.

Methods

Modeling Approach

We treated the first inning as its own game. It is unique to later innings as pitcher performance from an entire appearance may not be fully representative of their performance in the first inning. To account for

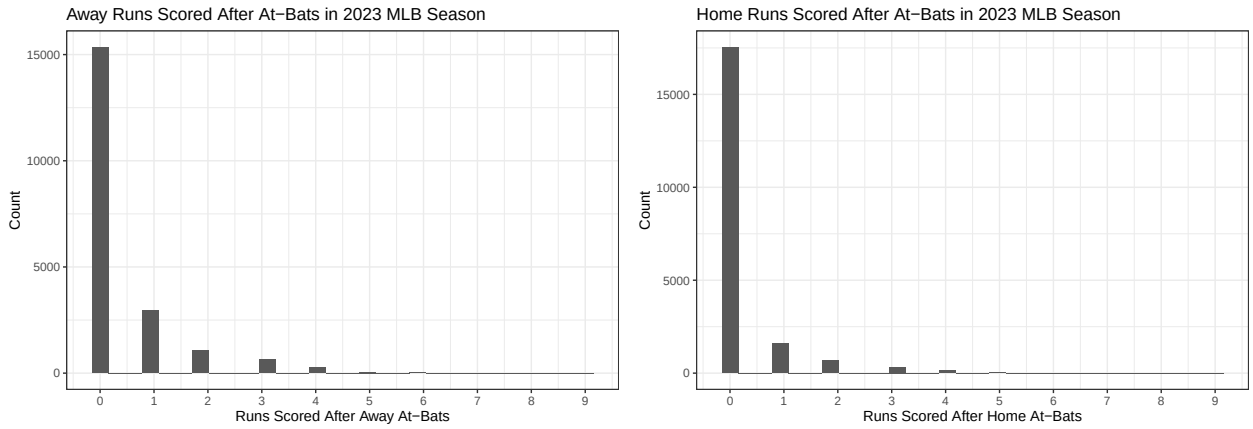

Figure 1: The distributions for away and home run scoring appears to be Poisson.

a minimum 3 plate appearances in the inning, we computed each team's home and away run scoring rate. This feature is appropriate as many teams typically have the same starting lineup throughout the season, therefore we are assessing the first 3 hitters simultaneously.

Initially, we ran simulations using our home and away runs scoring rates. This model was naive as it does not take into account which pitcher is pitching for each team.

We then created a new indicator variable to indicate first innings that ended 0-0 or not. From there we used a multilevel logistic regression with random effects for the home and away teams, the home and away pitchers, and fixed effects for the home and away pitcher FIPs (Fielding Independent Performance), while still including fixed effects for the home and away run rates. This model was assessed using its accuracy score.

To achieve more uncertainty, we then converted this same multilevel model into a Bayesian multilevel model with weakly informed priors. We chose weakly informed priors as there were major rule changes such as the pitch clock and larger bases which provided major impacts on run scoring.

In addition, because we were not satisfied with the performance of our Bayesian model, we also implemented an XGBoost model to also predict whether there would be no runs in the first inning. We did this with the belief that by capturing nonlinear relationships and interactions between our features, we could produce more accurate predictions.

Bayesian Multilevel Model Specification

Level 1:

$y_g = I + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + a_i + b_j + c_k + d_l + \epsilon_g$

Level 2:

$a_i = \alpha_0 + u_i, u_i \sim N(0, \sigma_i^2)$

$b_j = \beta_0 + u_j, u_j \sim N(0, \sigma_j^2)$

$c_k = \phi_0 + u_k, u_k \sim N(0, \sigma_k^2)$

$d_l = \tau_0 + u_l, u_l \sim N(0, \sigma_l^2)$

Priors:

Level 1 Intercept $\sim$ N(0, 6.25)

Level 1 Coefficients $\sim$ N(0, 25), N(0, 216.3841), N(0, 3.1329), N(0, 464.4025)

Covariance Matrix $\sim$ Decov(1, 1, 1, 1) for regularization, concentration, shape, and scale parameters respectively.

I is the fixed intercept term for the level 1 equation, X_1 is a fixed effect for the home team's run rate, X_2 is a fixed effect for the away team's run rate, X_3 is the home team's fielding independent pitching rate (FIP), X_4 is the away team's FIP, a_i is a random effect for the home team, b_j is a random effect for the away team, c_k is a random effect for the home team's pitcher, d_l is a random effect for the away team's pitcher, and ϵ_g is a term that captures the level 1 prediction error.

For the model where we assumed first inning runs were Poisson distributed and predicted the number of runs that would be scored in the first inning, the level 1 response variable y_g is the total number of runs scored in that inning. To make predictions of whether runs were scored or not, we decided that if the predicted number of runs was 0.5 or higher, the prediction was that "Yes Runs First Inning", otherwise the prediction was "No Runs First Inning".

For the model where we assumed first inning runs were Binomial distributed and predicted whether or not runs would be scored in the first inning, y_g is a binary indicator variable with values of 0 (indicating no runs were scored) and 1 (indicating that runs were scored).

XGBoost Model Specification

To take a different modeling approach, we also fit an XGBoost model. XGBoost trains a series of decision trees where each subsequent tree is an improvement of the last. The process then optimizes both the loss function and the regularization term. Using this process is an improvement from our models as both interactions and feature importance will be much more sound than our previous experimentation approach. When training the model, we used 24 features for both home team and away teams to predict a NRFI occurring. We then used an iterative process to find the ideal tuning parameters to return a model that consistently produced the highest accuracy.

Results

The accuracy of these models is assessed by accuracy scores, however we kept in mind that successful betting algorithms must be approximately 60% or better to be profitable as many of these NRFI bets have favorable odds for occurring.

The initial multilevel model failed to hit our benchmark accuracy of 60% as it scored approximately 42% accuracy. With its poor performance, bettor's would be better suited guessing NRFI or YRFI for every game. So then we switched to our 2 Bayesian multilevel models with the binomial and Poisson families.

Our Poisson Bayesian multilevel model performed better than the initial multilevel model with an accuracy score of 56.13%. While this is much better performance, it was still on the edge of profitability. The logistic Bayesian multilevel model we implemented also achieved a barely profitable accuracy score of approximately 58.64%.

After our unsatisfactory results from the Bayesian multilevel modeling, we took a different approach by implementing extreme gradient boosting (XGBoost). The XGBoost model that we implemented was far superior in performance compared to our other modeling attempts, achieving 90% accuracy on the 2023 data it was trained on and 68% accuracy on 2024 games when used as holdout data. This accuracy falls well above our aim of approximately 60% accuracy to be profitable.

When looking at a confusion matrix for the 2024 NRFI predictions in Figure 2, we can see that false negatives and positives are being predicted somewhat evenly. The XGBoost model is also predicting more NRFIs than YRFIs. When examining the ten most important features for the model (Figure 3), we see that fielder independent pitching for both home team and away team (denoted as home_fip and away_fip) are the most significant contributors. Subsequently, different pitch types for both home and away teams impact decisions

at varying levels. In Figure 3, these different pitch types are denoted as away_ff, home_cu, away_cg, etc. Ultimately, XGBoost is telling us that the pitcher and the type of pitches they throw are what matters most. Such feature importance holds true through k-fold cross validation on the 2023 season.

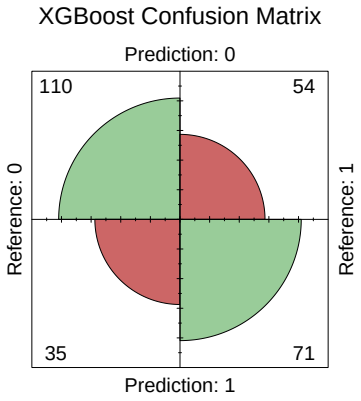

Figure 2: Confusion Matrix shows the predictions vs the actual outcomes of first inning results.

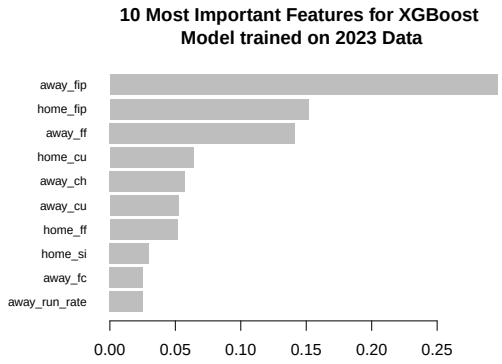

Figure 3: Away pitcher and home pitcher fip are consistently the most important variables when determining the probability of a NRFI.

Discussion

Overall, our findings show the Bayesian models we applied to predict the occurrence of a NRFI does generate accurate enough predictions to be used to slightly profit from bets. While it might be possible to improve this model through accounting for more features, including interactions, or a developing a more strongly-informed prior distribution, the current model does slightly better than suggesting a random guess when it comes to deciding which games will have an NRFI.

A significant challenge present during this project was locating reference material that could be used to improve our model, as this topic will generally only be researched in a sports betting context where bettors will most likely not be inclined to share their betting model, as doing so would potentially reduce the odds and payout of the wager. Additionally, the dynamic nature of baseball means a high probability of an NRFI is not always indicative of the reality, so even when the prediction of an NRFI for a game has a fairly

large probability it is possible that a run may still be scored, making the confidence of the prediction very important when deciding whether or not to place the bet.

Taken together, we conclude that future iterations of this project should include more features and a better informed prior distribution in the Bayesian model to increase the model accuracy, as well as develop a method to increase the probability value confidence to decide when to place the bet. Until the prediction accuracy and confidence meets our previously specified threshold, we should only consider using our XGBoost model to make betting predictions.