Betting the Buzzer

Bhargav Hadya, Jason Huang, John Wang, Samuel Yu

April 2024

1 Introduction

The main motivation for our project comes from sportsbetting. It's very interesting to see the predictions made by professionals who just analyze and predict sports as their job, and we thought it would be a good challenge to try to come up with models that could try to beat the market. We thought this could have some merit since the lines on sportsbooks often end up very far off from the true result, and additionally sometimes different sportsbooks disagree with each other, which displays some level of mispricing.

Although the lines may occasionally be off, we still have a lot of faith in the professionals, so for our project we decided to narrow down our area of research into specifically the Over/Under and Point Spread lines of NBA games to try to beat the market. When we analyzed the Vegas sportsbook lines in this area, they were off by around 17 points on average on their pregame lines!

There are two main challenges that make this task so difficult. First, the lines and outcomes have extreme variance. There are an average of 100 possesssions per NBA game, so we are effectively trying to predict the outcome of 100 weighted coin flips where even the weight changes! The second key challenge is that there are just so many factors that influence the outcome of a NBA game, from game location and environment, to each team's skill level, down to each player's individual performance.

We considered 2 approaches to address these challenges:

In part 1 of our report, we decided to narrow down our scope even further into the point spread in the last 6 minutes of each game. By doing so, we are able to drastically reduce the variance of our responder, while also maintaining useful results since live betting is possible and profitable at the end of each game. Additionally, by using the first 42 minutes as predictors, we are able to factor in many different effects such as player performance and other game-specific details that may otherwise be difficult to predict.

In part 2 of our report, we focused more on the issue of the large number of factors that affect each game. In this part we still focus on pre-game predictions of total score, and with the main goal of performing feature engineering to come up with better predictions. There is so much noise in so many different variables that affect the game, so this approach helps narrow down all the noise to the specific factors that have the most predictive power.

2 Data

The data we used for our analysis of point spreads was sourced from publicly available NBA Play-by-play data from the 2019-2020 season. This dataset contains important details about each play, including home score, away score, time remaining, number of points scored, and others, over the course of 82 regular season games. Given the overall scores, we were able to compile the game scores and spreads at each 6-minute interval and calculate the number of points scored during each interval, which we wanted to use as predictors for the changes in spread during the final 6 minutes.

The data that we used for predicting the over/under was gathered from hoopR. We used NBA 2021-2022 season information, which contained variables on every game, such as score, teams, location, date, and game level statistics. We also pulled advanced statistics per game from the NBA API such as offensive and defensive rating, as well as pace. Offensive rating is defined as the number of points a team scores in 100 possessions, and defensive rating is how many points the team allows in a 100 possessions. Pace is defined to be the number of possessions in a game. We also pulled betting data from ESPN on the game's over/under. We then merged all of these into a single dataframe by using unique identifiers of the date and teams playing. In order to leverage information about earlier parts of the season in later games, we feature engineered average cumulative values for offensive and defensive rating, as well as pace. We also created a 5-game rolling average for offensive and defensive rating, to account for slumps and streaks in the season. These features were engineered for both home and away teams. In order to stabilize the pace values per game, we averaged the cumulative pace values for the home and away teams for that game, and used that as the pace in our methods. For our out of sample evaluation, we queried hoopR for information on the 2022-2023 season, and did the same feature engineering.

Figure 1 displays violin plots of each team's distribution of total points in a game when they are home or away. The lines within the plot are the 95 percent confidence intervals for each team. We can see that some teams, such as Denver, Houston, and the Los Angeles clippers have higher means than thte other teams, and the tails of their distribution are much higher as well. The Washington Wizards seem to have an extremely large spread of points score when they are at home , and a much shorter range when they are away. One way to investigate these differing distributions coudl be to fit random effects for each team, and build a multilevel mdoel based on the feature engineering we have done, since there is a clear grouped structure between the home and away teams, and we have many team level features such as offensive and defensive rating.

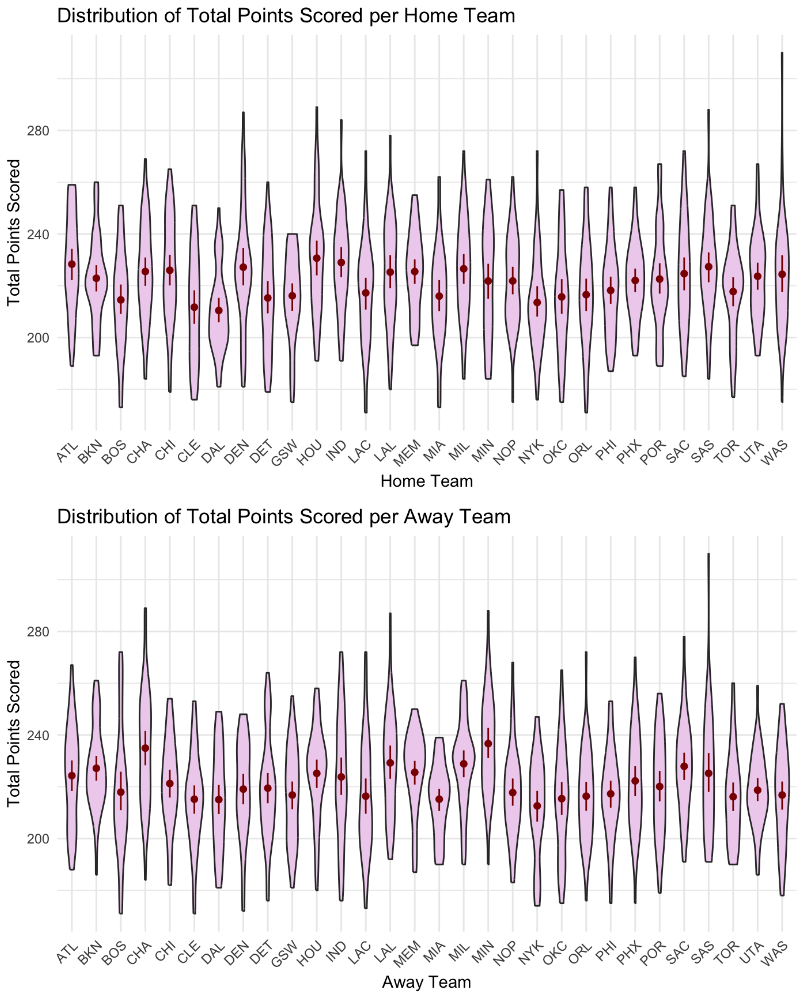
3 Methods

First, we wanted to see at which times in the game did the scoring spread between the home and away teams would have the most predictive effect, if any, and whether this effect differed across teams. We will define the spread as the difference between the home score and the away score. For our feature selection process, we used lasso regression to filter for the point spread and score snapshot factors that are are most likely to be significant and eliminate non-significant factors. We found that the only significant spread factors were the spreads at halftime and with 6 minutes into the game. This does make sense intuitively, as teams will likely adjust their strategy given unique endgame scenarios or when they have sufficient time to do so.

We then fitted a multilevel model using random team intercepts and fixed effects for each of the significant spread factors found by lasso regression as follows:

Level One: Game level information

$$spread_{final,a,h,i} = a_{ha} + \beta_1 * spread_{24m,i} + \beta_2 * scorediff_{42m,i} \beta_{ha} * spread_{42m,i}$$


Distribution of Total Points Scored per Home Team

Distribution of Total Points Scored per Away Team

Figure 1: Distribution of Total Points for Home and Away teams

Level Two: Home and Away team information

$$a_{ha} = \alpha_0 + u_h + v_a$$

$$\beta_{ha} = \beta_3 w_h + y_a$$

where
$\begin{bmatrix} u_h \\ v_a \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_h^2 & 0 \\ 0 & \sigma_h^2 \end{bmatrix}\right)$ and
$\begin{bmatrix} w_h \\ y_a \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix}\right)$

Here, h represents the home team, and a represents the away team.

The composite model is:

$$spread_{final,a,h,i} = \alpha_0 + u_h + v_a + \beta_1 * spread_{24m,i} + \beta_2 * scorediff_{42m,i} + (\beta_3 + w_h + y_a) * spread_{42m,i}$$

To see if we could achieve a lower MSE on our spread predictions, we also fitted a generalized additive model, or GAM, using the same predictors as our multilevel fit, smoothing the spread terms and including interactions for spread and team factors. The fitted model is:

$$\begin{aligned} Spread_{final} &= s(Spread24m) + s(Spread42m) + s(ScoreDiff42m) \\ &+ HomeTeam + AwayTeam + HomeTeam * ScoreDiff42m + AwayTeam * ScoreDiff42m \end{aligned}$$

We used smoothing splines to model the nonlinear terms.

After looking through the EDA, we identified that offensive and defensive rating would be important towards modeling the over/under at the end of the game. Qualitatively, the speed at which teams play also should impact how many points they score, so we also considered the pace of both teams in the matchup. In our pre-processing step, we created cumulative values for all three attributes, as well as 5-game rolling averages for offensive and defensive rating. The first model that we fit was a simple linear regression in which we had cumulative and rolling offensive and defensive rating as predictors for home and away teams. All of these predictors interacted with average pace, which we defined as the average cumulative pace of the two teams playing. We used cross-validation to find a simpler model with these coefficients.

$$\begin{aligned} TotalPoints &= \beta_0 + \beta_1 * offrtgHome : pace + \beta_2 * defrtgHome : pace \\ &+ \beta_3 * offrtgAway : pace + \beta 4 * rolling\_offrtgAway : pace \end{aligned}$$

Lastly, we leveraged the grouped structure of the data and fit a multilevel model with random intercepts for home and away teams. For this model, we zeroed the overall intercept term The model is as follows:

Level One: Game Level Information:

$$TotalPoints_{h,a,i} = a_{h,a}$$

Level Two: Home and Away Team level information:

$$a_{h,a} = u_h + v_a + \beta_1 * offrtgHome_h : pace + \beta_2 * defrtgHome_h : pace$$
$$+ \beta_3 * offrtgAway_h : pace + \beta 4 * rolling\_offrtgAway_a : pace$$

where $u_h \sim N(0, \sigma_h^2)$ and $v_a \sim N(0, \sigma_a^2)$

The composite model becomes:

$$TotalPoints_{h,a,i} = u_h + v_a + \beta_1 * offrtgHome : pace + \beta_2 * defrtgHome : pace$$
$$+ \beta_3 * offrtgAway : pace + \beta 4 * rolling\_offrtgAway : pace$$

Here, h represents the home team and a represents the away team. For both the multilevel and the simple linear regression models, all the variables represent cumulative season averages up to that game, unless the variable name has rolling in it, in which case it is the last 5 game rolling average.

4 Results

We ran a variety of models and ran cross validation on each of them to calculate their MSE values and compare their predictive accuracy with each other.

For part 1, we ran 2 LASSO regression models and 5 GAM models, with the best 2 being the ones described above. The model performance values are shown in the table below, with baseline being just predicting the mean since we didn't have access to betting data from within each game.

Model	Train MSE	Val MSE
Baseline	34	34
LASSO	30.77 ± 0.395	30.77 ± 1.577
GAM	26.26 ± 0.387	26.266 ± 1.55

Table 1: Model comparison based on Cross-Validation Mean Squared Error (CV MSE) and Standard Deviation (SD)

From this, we can see that the GAM model performs the best, with the smallest MSE value in addition to slightly lower variance. This is to be expected given our prior knowledge of basketball since the spread doesn't scale linearly with the previous spread. For example, there is quite a big difference between being up 1 or 4 points due to the fouling strategies that are present at the end of close games in basketball. So, it's reasonable to observe that GAM is able to capture the nonlinear effects much better than LASSO.

Figure 2 displays the coefficient values of the GAM model for the home and away teams. As we can see, all the away teams have negative coefficients, which means that away teams usually increase the spread at the end of the game. There are a few large values, such as Trail Blazers and Raptors. They were also found to be statistically significant, which means that both teams have a nonzero impact on increasing the spread at the end of games. The coefficients for the rest of the away teams were foudn not be statistically significant, which could mean that they don't really impact the spread. For home teams, we found that only the Oklahoma City Thunder have a statistically significant coefficient. However, just looking at the distribution, it is interesting to see that the mean

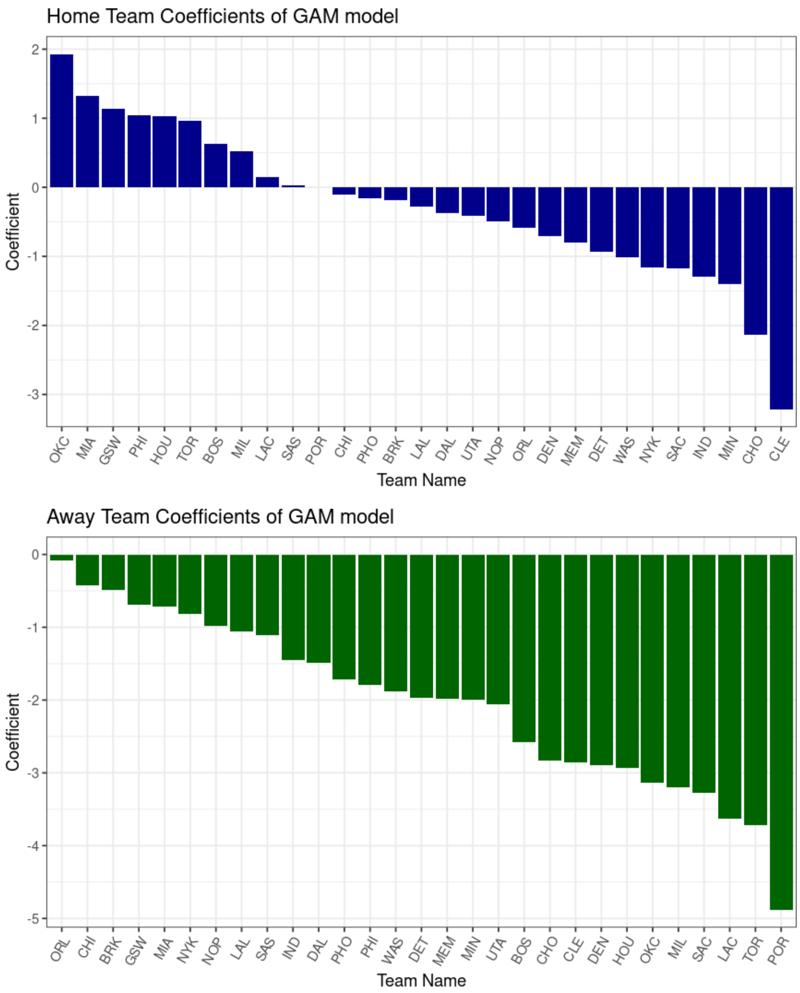


Figure 2: Home/Away team random effects in the over-under multilevel model.

for many of the home teams is negative, which implies that their home-field advantage may not be that good.

4.1 Over/Under Prediction

We compare models using 5-fold cross-validation on games in the 2021-22 season and report the mean and standard deviation of the training and validation MSEs. The results of the models are shown in Table 2. We see that adding in random effects for the home and away teams improves the validation MSE and even the MSE on 2023 data (the test set), despite the random effects being calculated on 2022 data, and teams substantially changing between seasons. However, we note that the validation MSE for the multilevel model is not significant, which may be because we only use one season of data, which is only 1200 games. While there are not confidence intervals for the 2023 MSE, we do note that the values are roughly similar to the cross-validation MSE, which is a good

Model	Train MSE	Val MSE	2023 MSE
Linear (all variables)	326.68 ± 7.18	334.63 ± 30.06	336.70
Linear (handpicked)	330.69 ± 8.27	335.86 ± 32.33	332.10
Multilevel	303.28 ± 8.56	325.77 ± 33.22	324.54
Vegas	–	320.98	332.08

Table 2: Results of several models for predicting over/under in NBA games. We report the mean and standard deviation of 5-fold cross-validation on the 2021-22 season. We also show the results when deploying the model on 2022-23 data. Note that the performance of the two linear models are pretty similar, but adding in random effects for the home and away team decreases the MSE. However, all the validation results are non-significant (within one standard deviation).

Feature	Estimate	Standard Error	t Value	$Pr(>\|t\|)$
intercept	-83.39	23.16	-3.60	0.00
rolling_home_offrtg × pace	-0.09	0.11	-0.80	0.42
rolling_away_offrtg × pace	1.10	0.11	10.09	0.00
rolling_home_defrtg × pace	0.22	0.11	1.95	0.05
rolling_away_offrtg × pace	-0.27	0.11	-2.40	0.02
cumulative_home_offrtg × pace	0.48	0.14	3.45	0.00
cumulative_away_offrtg × pace	0.51	0.13	4.08	0.00
cumulative_home_defrtg × pace	0.42	0.15	2.77	0.01
cumulative_away_dffrtg × pace	0.43	0.16	2.73	0.01

Table 3: Coefficients for the linear model predicting over-under. We note that most of the coefficients are significant, but some are hard to interpret, such as having a negative estimate for rolling_away_offrtg × pace. Perhaps it is a signal of some sort of mean reversion.

sign that our models are consistent.

We also look at the models more closely in Table 3, which show the coefficients for the first linear model, and Table 4, which show the coefficients for the multilevel model. We note that the features we made for the linear model seem to be pretty good overall, with most estimates being significant. However, some of the features don't particularly make sense, as it is expected that all the coefficients should be positive (higher offensive/defensive rating means more points scored), but rolling_home_offrtg × pace and rolling_away_offrtg × pace both have negative coefficients. Otherwise, the rest of the coefficients are in line with what is expected; they are all roughly 0.4-0.5, so the model is taking a portion of "expected points" scored/allowed by each of the teams. Another interesting coefficient is the rolling_away_offrtg × pace, which is much larger than the other coefficients, which means that the strongest indicator of points scored is how the away team's offense has been doing recently.

We keep the most significant ($P < 0.01$) features for the multilevel model and observe similar estimates as that of the linear model, where the coefficient for rolling_home_offrtg × pace is the largest. We also visualize the random effects in Figure 3. We can see that for example, when Houston is the home team there is a large positive random effect, which can make sense because the Rockets had a poor defense as shown in our EDA, and Dallas had a very good defense, which is

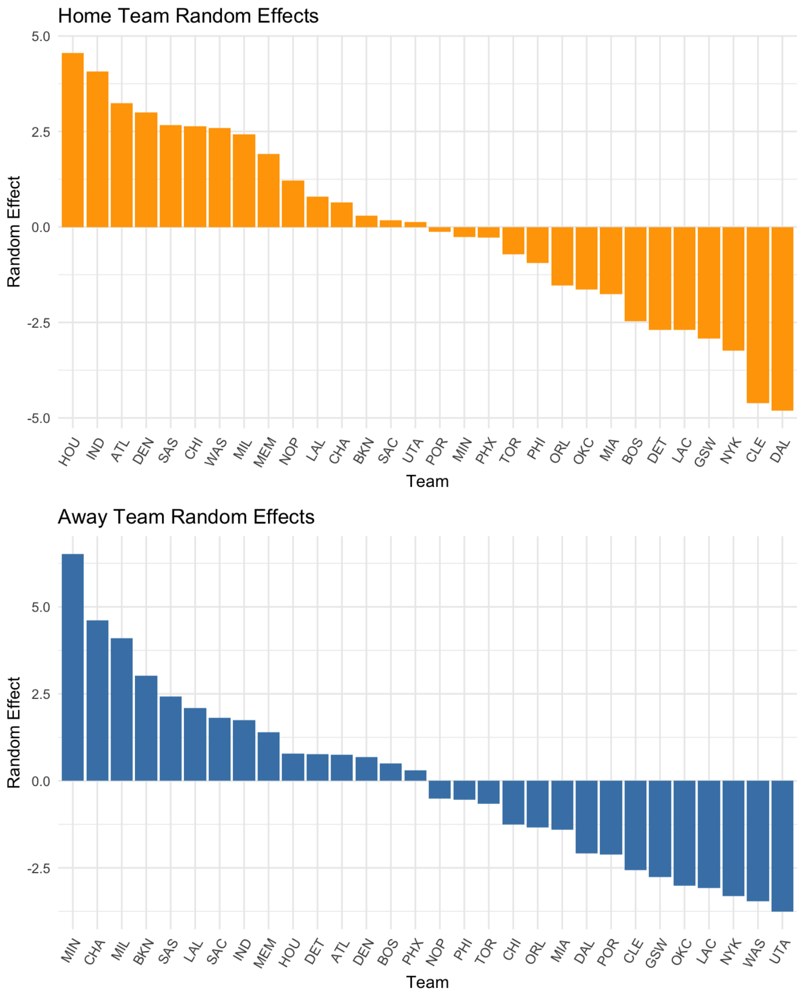


Figure 3: Home/Away team random effects in the over-under multilevel model.

Feature	Estimate	Standard Error	t Value
rolling_home_offrtg × pace	1.01	0.09	10.98
cumulative_home_offrtg × pace	0.32	0.14	2.22
cumulative_away_offrtg × pace	0.29	0.15	1.88
cumulative_home_defrtg × pace	0.41	0.14	2.88

Table 4: Coefficients for the multilevel model predicting over-under. We only use a subset of features that were significant and interpretable from the previous linear models. We also don't include an intercept because the random effects serve as intercepts.

Model	Profit/Loss	Accuracy
Linear (all variables)	$12620	56.59%
Linear (handpicked)	$11990	56.34%
Multilevel	$12200	56.42%

Table 5: Betting results on the over-under in the 2023 season.

demonstrated by the large negative random effect.

Finally, we also run a simulation, using models trained on the 2021-22 season to predict and bet on the over/under in games in the 2022-23 season and calculate the profit/loss of each model across the whole season, which is shown in Table 5. Interestingly, all the models are profitable, and all have an accuracy of roughly 56.5%. Also, the linear model including all the offensive and defensive ratings did the best, over the multilevel model. This can make sense as well because the multilevel model uses random effects from 2022, but teams in the 2023 season are wildly different, so the random effects are not applicable. For better evaluation, the model should be retrained on a rolling basis as well.

5 Discussion

Although our models do a good job of beating the baselines, it's important to note that there are quite a few caveats, although our results are still useful. We will now consider the statistical results above in the context of sportsbetting.

In part 1, we obtain Val RMSE of 5.12 vs the baseline of 5.83, which means our predictions are 0.71 closer than the baseline on average. Based on empirical evidence, at the end of the game the markets price each spread point at roughly 6% chance of occuring(alternate line odds shift roughly 25 per point). So, we can approximate that this model gives us $0.71 \cdot 6\% = 4.26\%$ of edge. However, this isn't enough to be a profitable strategy since most markets at this time charge around 3.5% fee (-115/-115 odds) and we are also using quite a weak baseline. Instead, this model is much more useful as a sort of tiebreaker when markets disagree: at the end of games, sportsbooks frequently disagree on what they think is the fair, in which case there is an opportunity to make a trade if we can predict which book is more likely to be correct using our model.

In part 2, our models achieve comparable MSEs to the Vegas line, and actually make a profit in betting simulations with an accuracy of roughly 56.5% for all the models. We note that this is possible because the Vegas line is set to balance the amount of money the public bets on each side, and so it is more reflective of the public sentiment rather than the actual 50/50 line. As such, even though our model doesn't necessarily beat the Vegas line in terms of MSE, it has an edge because the line isn't perfect and our model is somewhat orthogonal to the Vegas line. However, we do not think that any of these models should be actually used for sports betting because we lack extensive backtesting, and the multilevel model is somewhat hard to interpret.

For future work, it is actually quite promising to research the combination of the two parts: using the engineered features in the last 6 minutes of the game to predict lines for live betting. We would be curious to obtain historical live sportsbook odds in order to backtest this model. We have seen a lot of profitability from purely arbitraging books that disagree in live betting without ever taking an opinion, so this model would help generate even more opportunities to take more directional trades rather than hedging fully when these situations arise. We would also want to get more data, such as information from the last 15 seasons, such that we have more seasons to cross-validate on. We would also be able to develop a pipeline that continuously retrains the models as the season goes on, which would likely generate the most accurate predictions, and is the most representative of how the model(s) would be used in the real world.

6 Appendix