NBA 2023 Predictions

Spread, Total and Offensive Rebounds

Weston Murdock, Neil Vakharia, Kevin Pignone

(Image from "2023 Midseason Report")

Data Information

Compiling and Cleaning the Data:

Our biggest priorities when cleaning the data were thoroughness in the predictors and having a large sample size to pull from. Similar to how Nathan Luaga's Kaggle dataset was formatted, we wanted a file with each row representing a game. We wanted columns to include the game's date, the home team name, the away team name, various stats from the home and away teams, and the total points, spread, and total offensive rebounds.

We included games from the 2012-2013 NBA season until the present, specifically cutting off our recent data at games played on March 20, 2023. We chose the 2012-2013 season as the starting point for our data because the NBA went through a change in playstyle during this season, as 2012-2013 marked the beginning of the 3-point-heavy era. Centers began shooting perimeter shots more, proven by the fact that out of the 6 centers featured in the 2012-2013 All-NBA teams, three of them primarily spent their time on the perimeter (Kevin Love, Dirk and Blake Griffin). Stephen Curry, arguably the greatest 3-point shooter of all time, had his first MVP season this year, having nearly 600 3-point attempts. In contrast, in the 2011-2012 season, only one player had shot more than 400 3-pointers. There was a definite culture shift around 2012-2013, so we used that as the furthest season back we pulled from. However, we understand that pulling data from 10 years ago is not a great representation of the current times. To adjust for this while still trying to maintain a large training set, we gradually dropped a proportion of games from the older seasons in our dataset, as is detailed more thoroughly in the "Scope of the Data" subsection found below.

We used the nbastatR package to acquire our games of interest beyond what was provided to us in the Kaggle NBA Games Data dataset. We used the game_logs() function with

the ‘seasons’ parameter set to 2023 to gather data on games that took place between now and the last time the Kaggle dataset was updated, as well as observe the range of game ids corresponding to these games. We then looped through this range, taking each individual game id and getting two rows from the game log, one for the home team and one for the away team. Next, we renamed the column headers to include ‘home’ or ‘away’ respectively, to differentiate between the two teams in the same manner as the Kaggle dataset. Finally, we merged these two games into one row with all predictors, creating new rows that could be bound to the end of the Kaggle dataset. Total Points was calculated as the sum of home points and away points, Spread was identical to the “+/-Team” of the home team (+/- referring to the point differential), and OREB was the sum of home and away offensive rebounds. We repeated this process for every season going back until the 2012-2013 season. Although it may have been more efficient to do this in a single programming loop in R, we had to do this in iterative chunks because the magnitude of data was making our computers crash.

From there, we looked at missing data and outliers. Because the package is so comprehensive, there were no rows with statistics missing. When considering outliers, we identified a couple situations which could potentially classify a game as an outlier. However, through our investigation, we eventually chose not to drop any rows. The first situation was a game where an individual player scores an unexpectedly large amount of points. We chose to look at Devin Booker’s 71 point game. In spite of scoring 71, the team in total scored 120 points and lost to a score of 130. The total score was 250, which is high scoring, but not high scoring enough to be an outlier. We discovered that in most situations where a player scores 50+ points, they take opportunities away from their teammates to score, so the final scores end up being relatively normal. Our second potential source of outliers was if a game went to overtime. Our

two ideas for handling this were standardizing all our variables by the number of minutes or standardizing all our variables by the number of possessions. However, we knew we would be predicting on the naive assumption that no games would go to overtime, and for possessions specifically, there was too much variance in the number of possessions between teams in the same game and number of total possessions between games, indicating that removing overtime games would simply be needlessly reducing our dataset. At this point, our baseline dataset of all games was set.

Variables Engineered:

After getting our baseline data, we wanted to add in other, more complex variables. The first ones we added were coined as STARTERS_PM_home and STARTERS_PM_away. These variables were calculated as the sum of the plus/minuses for the starting 5 on the home team and away team respectively. We hoped that these variables would be useful for predicting spread and total points. We felt that, even if bench players have an impact on the game, basketball games are mostly a product of the starters' production. For example, the Memphis Grizzlies and the Phoenix Suns after the Kevin Durant trade are both topping charts because of their dominant starting 5. We felt that considering the isolated impact of starting-caliber players such as Ja Morant and Jarren Jackson Jr. would be valuable because there is a very strong correlation between the wins of a team and their strength of starting 5. These variables were added in the dataset by referencing the game_details spreadsheet from the Kaggle dataset, and for more recent games that were not in the dataset, we referenced the box_scores() function from nbaStatR, aggregated the +/- from each player for each game, and merged this into our main dataset accordingly.

The choice to aggregate this data was done in order to avoid looking too closely at the impact of specific players, and instead look at general trends across different starting lineups, for several reasons. First, when predicting games in the future, it may be uncertain whether or not a given player is actually going to play in a particular game, and whether or not they will play for the full amount of time they would normally, due both to coaching and management decisions, as well as injuries that may arise naturally through playing such a physically intensive sport. Second, to consider player level statistics would introduce a massive amount of additional variables to our dataset, which would be infeasible to manage without extensive research and further statistical analysis, as including each individual player as a separate predictor would clearly be computationally infeasible, but choosing to only include a subset of players would create an issue in determining which players are important enough to include and which to exclude, on top of still being computationally infeasible if this cut off is not sharp enough. Thus, by aggregating some of the data from the starting lineup, we arrive at a type of compromise that minimizes the number of additional variables we are considering, while also still giving us a broad estimate of how particularly important players are impacting the game as a group.

Although player statistics are important, we wanted to capture metrics that a stat sheet would not tell. To us, a vital part of a basketball team's success is player physicality. Naturally, a taller and heavier player such as Steven Adams is much more likely to get a rebound compared to a shorter, lighter player like Kristaps Porzingis, even though they both play the center position. Even for predicting the amount of points a team puts up (pertinent to Spread and Total), the taller and stronger players tend to dominate their area of the offense. The idea of a metric to capture a team's physical prowess was worth exploring. The way we created a metric for this was we multiplied the average height in inches and weight in lbs of every player that played in the game

for every game to create an interaction term between height and weight. Because these numbers were disproportionately high compared to the rest of the predictors, the interaction term was standardized to a range of [0,1]. We saw this metric as a representation of how good a player in the paint could play "bully ball", so we coined this variable as such.

Bully ball metrics were computed for the home team and away team. Analyses of physicality have been tried in previous EDAs, such as this one: (NBA Player Height and Weight Analysis). The height and weight of the team was retrieved from the player_profiles() function in nbaStatR and merged into the larger dataset the same way that starters' +/- was. The only issue was that every time Jordan Nwora played a game, the API would hang, so we had to find and input his data manually. This issue only persisted for the one player (for reasons we have yet to discover).

Outside Data:

The outside data, as explained in the previous two sections, came from the nbaStatR package. The main advantages to this package include the lack of missing values, the ability to acquire data at the player level of granularity, and the sheer amount of predictors it provides. As seen in the next section, we were able to get multiple box score statistics, numbers for attempts, successes, and efficiency for all types of shots, and more.

Final Variables:

Once all of the previous steps were done, we were left with a dataframe containing the following variables, detailed in **Table 1.1**. Each game had a game_id that was not used in predictions, but was used in order to facilitate merging of statistics between the various dataframes that arose from the different import methods we used for gathering data. There was also a home_id and away_id, corresponding to the home and away teams, neither of which were used directly in making predictions, but which would have been used to generate estimates for the other predictors to arrive at our final predictions for future games. The variables that were used in the predictions are listed below, and each variable was differentiated between home and away statistics - ie. there was a FGM_home and FGM_away, etc.

Table 1.1: Predictor Variables Used

Variable Name	Description
TEAM_WINS	The number of wins the team had at the time (count)
FGM	Field Goals Made (count)
FGA	Field Goals Attempted (count)
FG_PCT	Field Goal Completion (%)
FG3M	3-point Field Goals Made (count)
FG3A	3-point Field Goals Attempted (count)
FG3_PCT	3-point Field Goals Completion (%)
FG2M	2-point Field Goals Made (count)
FG2A	2-point Field Goals Attempted (count)
FG2_PCT	2-point Field Goals Completion (%)
FTM	Free-Throws Made (count)
FTA	Free-Throws Attempted (count)

FTT_PCT	Free-Throws Made (%)
MIN	Minutes of Game (count)
OREB	Offensive Rebounds (count)
DREB	Defensive Rebounds (count)
AST	Assists (count)
STL	Steals (count)
BLK	Blocks (count)
TOV	Turnovers (count)
PF	Personal Fouls (count)
PTS	Points (count)
STARTERS_PM	Plus/Minus for Starters, per Team (Sum)
BULLY_BALL	Height * Weight, normalized

In addition to this, our dataset included the 3 target variables, detailed below in **Table 1.2**.

Table 1.2: Target Variables Used

Variable Name	Description
Spread	Home Points - Away Points
Total	Home Points + Away Points
OREB	Home Offensive Rebounds + Away Offensive Rebounds

One caveat is that we did not want to include the target variable itself in the list of predictors. Therefore, when predicting models for spread, we dropped spread from our X matrix and left it as the Y column. Columns for Total and OREB were included in the predictors. Likewise, when predicting Total, we did not use past Total values as predictors but decided to

consider Spread and OREB. This same process was done for OREB. Naturally, there are concerns about multicollinearity among other issues, but we wanted to include as many variables as possible as a baseline. From there, we performed feature selection, so the target variables that are used as predictors may not need to be included in our final models anyway.

Scope of the Data:

Because the nature of the game of basketball has changed over time, and our data spans almost a decade, we used a proportion-based adjustment to pick relevant data points. Old games still have predictive worth, but depending on how old the game was, we wanted proportionally less of them impacting our model training. We randomly sampled 1/3 of games from between the 2013 and 2015 seasons, 2/3 of games from between the 2016 and 2020 seasons, and 100% of games after that in order to build a training and testing set for our models. This new dataframe has dimensions of 4492 rows and 48 columns.

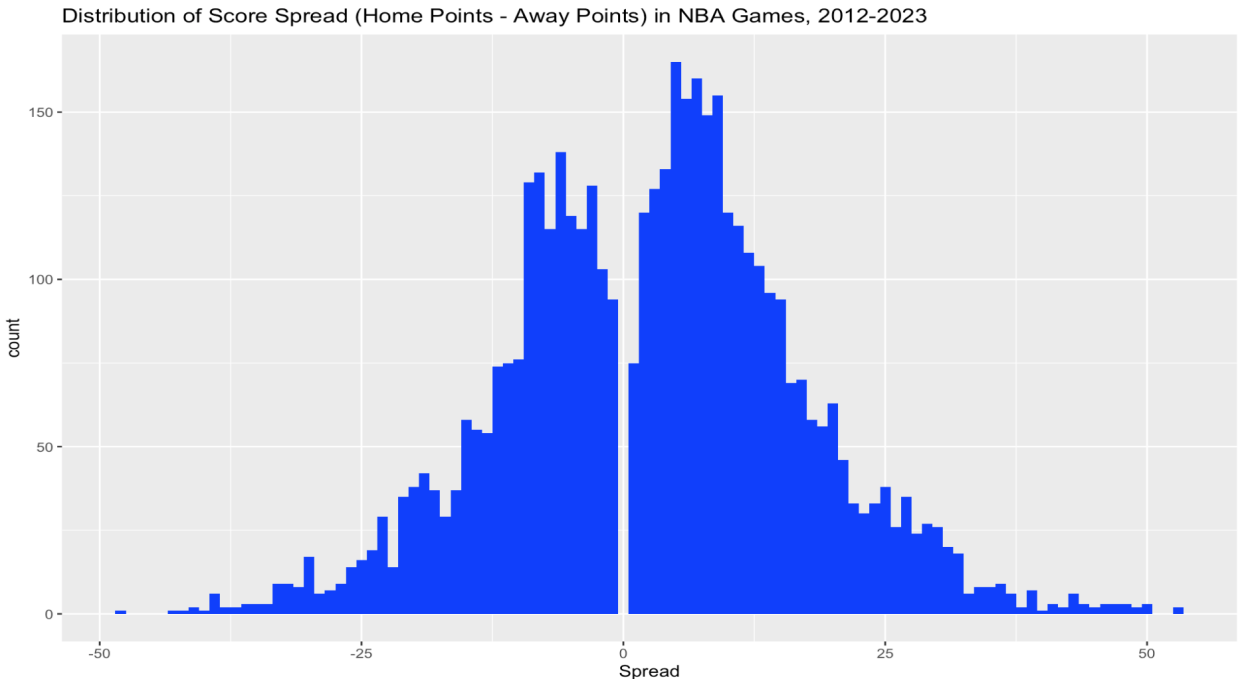
Train/Test Split:

After we randomly sampled our data to pick which games would be considered, we used an 80/20 train/test split to select data points for creating models, the training set, and data points for evaluating models (by means of MAE), the testing set.

Methodology for Spread

Distribution of Spread:

Figure 2.1: Distribution of Spread



As seen in the histogram above, under a binwidth of 1, we see that the distribution of point spread across the sampled NBA games appears to be bimodal. There are no observations for 0, since there are no ties in the NBA. However, most of our data falls within a spread range of (-10, 10) with higher spreads being less and less common. Therefore, it is safe to say that the distribution is very close to normal.

Methodology for Spread:

To begin, we wanted to establish a control for the most basic model. This was a simple linear regression model with all our predictors. In some sense, every variable in our full dataset has some impact on the number of points both sides score, so we wanted to see whether every

variable would be significant to the optimization. When testing such a model, we found a MAE of 15.38. Some games do have a 15 point differential in score, but as an average, this did not feel sufficient. Furthermore, the R^2 for the model was around 1, so there was evidence that there was perhaps overfitting.

Unsurprisingly, one big issue is that there is multicollinearity between our predictors. For example, our dataset includes measures of shots made, shots attempted, and shot percentages - field goal percentage can be derived from field goals made and field goals attempted and this issue of multicollinearity can be seen in other predictors as well. Therefore, we felt inclined to try ridge regression, a form of regression that penalizes terms with high collinearity. We chose a lambda of 0.1 because it tends to balance bias and variance well. Applying a ridge regression model yielded a MAE of 0.15. This result was miles better, and we theorized that the ridge penalty was incredibly helpful in shrinking coefficients of variables that had a dominant weight and it was also helpful in mitigating the effect of overlapping variables.

Table 2.1: Home Correlations for Variables and Spread

Variable Name	Correlation with Spread
HOME_TEAM_WINS	0.80
FGM_home	0.46
FGA_home	-0.01
FG_PCT_home	0.54
FG3M_home	0.24
FG3A_home	0.02
FG3_PCT_home	0.36

FTT_PCT_home	0.10
FG2M_home	0.28
FG2A_home	-0.03
FG2_PCT_home	0.38
MIN_home	-0.04
FTM_home	0.11
FTA_home	0.08
OREB_home	-0.02
DREB_home	0.38
AST_home	0.41
STL_home	0.21
BLK_home	0.17
TOV_home	-0.09

Ridge regression on all of our predictors was relatively successful, although there may have been some overfitting. From here, we wanted to perform some feature selection to reduce the number of predictors, isolating only the relevant ones. To understand which variables might help us predict Spread, we first looked for the variables with the highest correlation coefficients with Spread. For brevity, the coefficients of home variables are listed in **Table 2.1**. Unsurprisingly, the variable in our dataset that most correlated with Spread was Home Team Wins. Other promising variables included the aggregate FG Percentage Made, including 2 point FG completion percentage and 3 point FG completion percentage. Count of FG made, Count of FG attempted, Assists, and Defensive Rebounds were also correlated with Spread. These were all the variables with coefficients above 0.3. We coined this set of predictors as our “correlation set”.

We wanted to see if regression was more or less successful on our correlation set of predictors. We created a linear regression model to test that. We calculated a MAE of 14.88. This was marginally better than our linear regression with all predictors, but still an unsatisfactory error. We transitioned to ridge regression to see if penalizing multicollinearity was equally successful for this set of predictors.

Our ridge regression model evaluated on our correlation set, contrary to what we expected, performed horribly. Our MAE was 77.47. A 70+ difference in points is astronomically rare in the NBA. We investigated why this model performed so poorly. The most likely reason is that ridge regression penalizes multicollinearity, but the variables in this set of predictors are mostly unrelated. This could lead to an overpenalization, which is why the MAE was so high.

Our next step was to try a more precise method for feature selection. We tried a stepwise selection process. Our stepwise process led us to a variable set of: HOME_TEAM_WINS, FG3A_home, FTT_PCT_home, PTS_home, FG2A_home, MIN_away, OREB_away, PTS_away, BULLY_BALL_home, and Spread. Once we had this set, we wanted to repeat our model building process. Our stepwise linear regression model had a MAE of 15.87. This is marginally worse than the linear regression model for all predictors. However, all 3 linear regression errors are all around 15. This suggests that linear regression is simply not good enough to predict spread.

We also calculated MAE for a ridge regression model based on stepwise selection. This model performed similarly well to the ridge regression with all predictors. We observed an MAE of 0.77. This was nearly our best model, but it is slightly worse than the ridge regression for all predictors.

We wanted to explore non-regressive models as well. We interpreted a basketball game as a set of events, each with probabilities. Based on this probabilistic approach, we tried using a Naive Bayes algorithm, an algorithm that uses the argmax function to pick the score with the highest probability calculated under the naive assumption that all our variables are independent. Our model did not perform well. In retrospect, we should have expected this because many of the variables are very dependent on each other. Furthermore, each basketball game is different so deriving probabilities from disparate and wildly varied game stats may lead to bad results. We received a MAE of 45.72, one of our worst models thus far.

A common trend among many of our models is that there are a large number of predictors, so there are many outcomes to consider based on a team's play in vastly different categories. We want a model that is sensitive to data that can take thousands of different states. A model that handles such data well is random forest. The idea of using a random forest model is not uncommon in prior NBA predictions. Our model was based on pre-existing work from William Li that predicts whether or not NBA teams will beat the Vegas Spread Moneyline (Using Machine Learning to Predict Over / Under Moneylines). Our random forest model has 500 trees, so it is very well suited to classify hundreds of different end states that a game can take. It provided a MAE of 0.87, which performed very well. The final random forest performs so well because it makes so many splits at such a high granularity for each variable.

A final table of each model and their respective MAE's is listed below in **Table 2.2**.

Table 2.2: Spread Model Performances

Model	MAE
Linear Regression (All Variables)	5.94
Ridge Regression (All Variables)	**0.15**
Linear Regression (Correlation Set)	14.88
Ridge Regression (Correlation Set)	77.47
Linear Regression (Stepwise Selection)	15.87
Ridge Regression (Stepwise Selection)	0.77
Naive Bayes	45.72
Random Forest (nforest=11, ntree=500)	0.87

Best Model for Spread:

After generating all of the above models, we found that the model with the best MAE on the test set was the Ridge Regression (All Variables) model. This model led to an MAE of only 0.15, outperforming the other ridge regression models, and the random forest model that did similarly well. Ridge regression followed the formula given in Figure 2.2.

Figure 2.2: Ridge Regression Formula

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

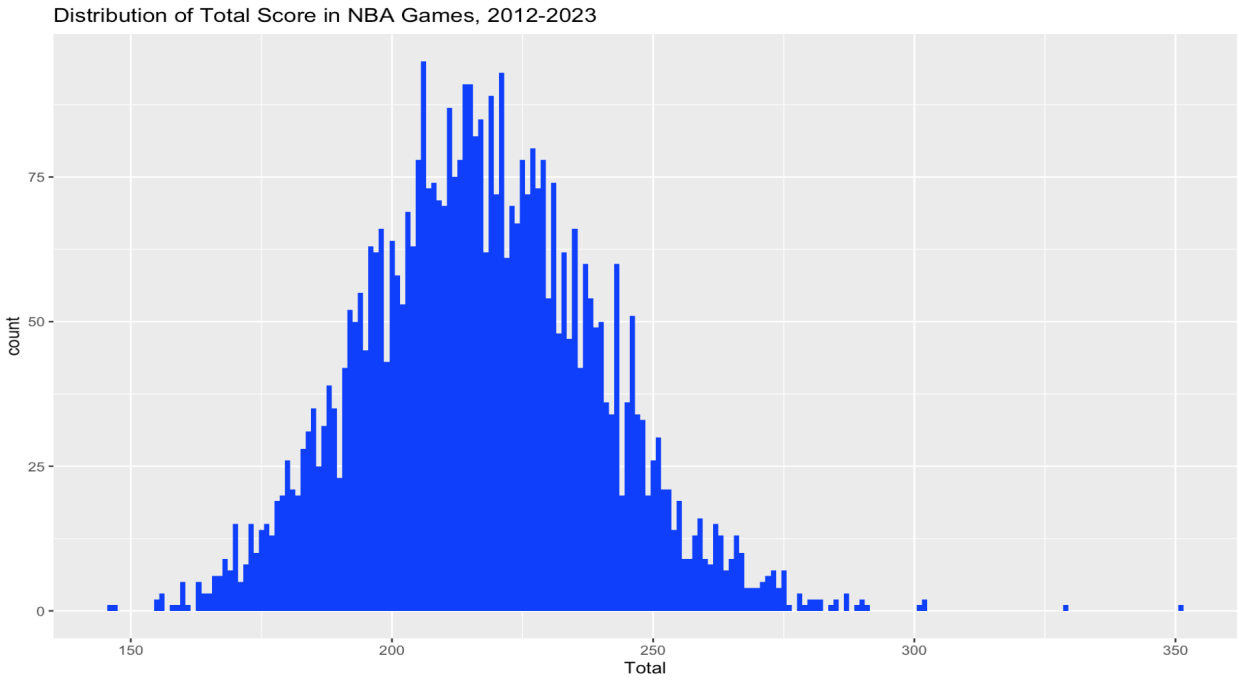
In our case, lambda would be 0.1 and the weights and biases are determined by the optimization function and are listed in R. Outside of the fact that ridge regression had the lowest

MAE, we felt that ridge regression on all predictors was our best model because it allowed us to leverage many predictors - both interconnected and not. We were not losing any predictive capability by removing any confounding variables. Instead they were being penalized by the model, but still allowed to have some weight. For such a complex statistic like Spread, allowing each predictor to carry some predictive weight, no matter how minimized, was important.

Methodology for Total

Distribution of Total:

Figure 3.1: Distribution of Total



The distribution of Total points for NBA Games between 2012 and 2023 is, surprisingly, approximately normal. Because of the similarities between distributions for total and spread, we

chose to explore similar models to the ones we did for spread. Given this distribution, we made the decision not to utilize a Poisson model for our predictions of Total.

We started off by trying simple linear regression with all of our predictors, just like how we did for our spread model. We got an MAE of 25.12. Total values being approximately normally distributed indicate that an error of around 25 points is not terrible, but definitely could be better, as can be seen in **Figure 3.1** by basic inspection.

From here, for the same reasons as total, primarily being high multicollinearity, we tried ridge regression on our model with all predictors. Our ridge regression model for total had a MAE of 0.13, which, just like the MAE for spread, was very low (maybe even due to an overfit). However, we were satisfied with this model because of its thoroughness.

Additionally, we wanted to perform feature selection. We looked at the correlation coefficients against total points. The results for nearly all home variables are shown in **Table 3.1**. The predictors that proved to be significant included home and away values for 3-point field goals made, overall field goal percentage, and overall field goal attempts among other variables. Notably, FG2A had a very low correlation, showing that the number of 2-pointers attempted had little correlation with the total score of the game. Instead, we know FG2_PCT—the percentage of 2-point FG made—to tell us a lot more about how many points are being scored rather than 2-point FG attempted.

Table 3.1: Home Correlations for Variables and Total

Variable Name	Correlation with Total
HOME_TEAM_WINS	0.00
FGM_home	0.69
FGA_home	0.41
FG_PCT_home	0.46
FG3M_home	0.50
FG3A_home	0.38
FG3_PCT_home	0.30
FTT_PCT_home	0.14
FG2M_home	0.31
FG2A_home	-0.03
FG2_PCT_home	0.43
MIN_home	0.28
FTM_home	0.30
FTA_home	0.26
OREB_home	0.02
DREB_home	-0.06
AST_home	0.45
STL_home	-0.03
BLK_home	-0.05
TOV_home	-0.04

We evaluated linear regression on our new correlation set. We calculated a MAE of 22.16, which was better than our model with all variables, but nowhere near close to the model for ridge regression. We then chose to look at ridge regression for the correlation set. We had a MAE of 10.82. We started to see a pattern. When isolating our model to only use the variables in the correlation set, the ridge regression was performing worse when compared to sets of other variables. This is most likely also due to the overpenalization that the ridge penalty adds.

We then used stepwise selection as our method of feature selection. We created a linear regression model and a ridge regression model. The MAEs were 25.12 and 0.193 respectively. In the same vein as spread, the linear regression on a stepwise set performed more or less the same as all variables and ridge regression resulted in very good results (but still not our lowest error).

When trying the Naive Bayes model, we received, by far, our worst results again. We found a MAE of 156.0, which is a near impossible level of error. Naive Bayes does not seem suitable for this type of prediction, at least for the structure of data that we have.

On the other hand, the Random Forest model performed considerably well on the test data. With a low MAE of 0.39, this was nearly our best model. This is likely due to the nature of the predictions being conducive to many factors, a sentiment that random forest handles well.

We compiled MAE values in **Table 3.2** below.

Table 3.2: Total Model Performances

Model	MAE
Linear Regression (All Variables)	25.12
Ridge Regression (All Variables)	**0.13**
Linear Regression (Correlation Set)	22.16
Ridge Regression (Correlation Set)	10.82
Linear Regression (Stepwise Selection)	25.12
Ridge Regression (Stepwise Selection)	0.193
Naive Bayes	156.09
Random Forest (nforest=11, ntree=500)	0.39

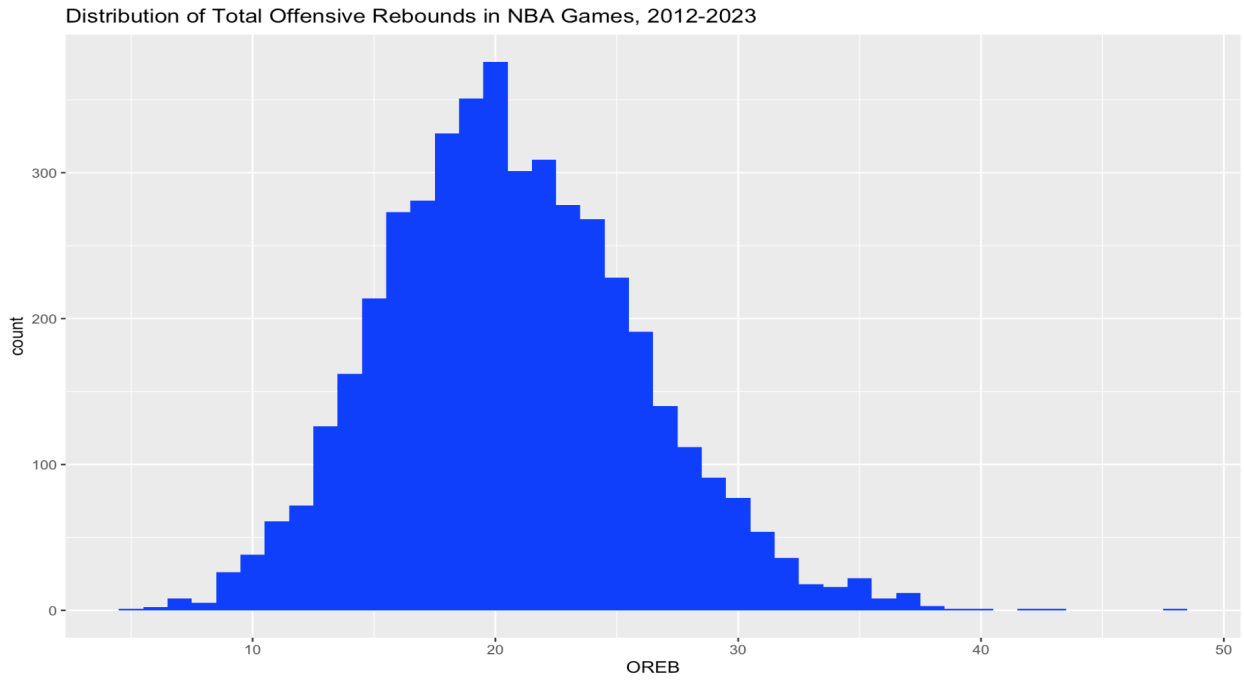
Best Model for Total:

Our logic for the best model for total is nearly the same as it was for spread. In predicting these values, both of which follow a normal distribution, ridge regression was successful because it let us keep all predictors while minimizing the harmful correlations they had amongst each other. The formula for ridge regression is given by **Figure 2.2**, just like it was for spread. All our hyperparameters (namely lambda=0.1) remain the same.

Methodology for OREB

Distribution of OREB:

Figure 4.1: Distribution of OREB



The distribution of Offensive Rebounds for NBA Games between 2012 and 2023 is also normal. Given this distribution, we made the decision not to utilize a Poisson model for our predictions of OREB, and instead elected to work with methods that assume normality.

Methodology for OREB:

Before we created models, we hypothesized that Bully Ball would be especially significant. On paper, we believed that bigger (both in height and in weight) players would have an advantage in rebounding, and thus that teams with a higher value in the Bully Ball stat we engineered would have a high number of OREBs.

We began with a linear regression model with all predictors. With a MAE of 5.94, this model in itself was not terrible. We were surprised by the results for ridge regression. Unlike the other two models, ridge regression on all variables performed worse than linear regression did. We wanted to see if this trend held true for different sets of features as well. We looked at the correlation coefficients of each variable against OREB. Very few variables had a correlation above 0.3, as can be seen in **Table 4.1**, so we chose every predictor with a correlation above 0.2. We ran linear regression on our correlation set for OREB and we got a slightly better result of 4.50. So far, this model performed better. This is likely because we removed many irrelevant variables. Our ridge model, however, performed worse, with an MAE of 29.

Table 4.1: Home Correlations for Variables and OREB

Variable Name	Correlation with OREB
HOME_TEAM_WINS	-0.01
FGM_home	0.02
FGA_home	0.33
FG_PCT_home	-0.21
FG3M_home	-0.08
FG3A_home	0.02
FG3_PCT_home	-0.14
FTT_PCT_home	-0.08
FG2M_home	0.07
FG2A_home	0.25
FG2_PCT_home	-0.18
MIN_home	0.12

FTM_home	0.04
FTA_home	0.08
OREB_home	0.68
DREB_home	-0.03
AST_home	-0.08
STL_home	0.03
BLK_home	0.22
TOV_home	0.03

We wanted to check if this pattern held true for stepwise selection. After the feature selection was performed, we were left with only a few variables: PTS_home and PTS_away. This is because of the penalty terms applied by the stepwise algorithm, limiting the number of predictors used in the model. Our stepwise selection process applied to linear regression gave us a MAE of 4.20, which we felt were pretty good results. When we tried ridge regression, we got much better results. A stepwise selection-based ridge regression model gave us a MAE of .08. Our primary idea for why this model performed so well was that stepwise selection procedurally got rid of any irrelevant variables, such as steals and assists. Although these variables are important to the point-based outcomes of the game, they do not necessarily impact the number of rebounds as well. Once these variables were eliminated, the ridge penalty was able to reduce multicollinearity well in the important variables, nothing more, nothing less.

When we tried Naive Bayes, we got a MAE of 4.44. This was, by far, our best Bayesian model. This was likely because offensive rebounds are a simpler statistic than spread or total, meaning that comparatively less goes into determining who gets an offensive rebound and when. Random forest, as expected, also performed extremely well, with an MAE of 0.20.

Again, MAE values are listed in **Table 4.2** below.

Table 4.2: OREB Model Performances

Model	MAE
Linear Regression (All Variables)	5.94
Ridge Regression (All Variables)	18.28
Linear Regression (Correlation Set)	4.50
Ridge Regression (Correlation Set)	29.00
Linear Regression (Stepwise Selection)	4.20
Ridge Regression (Stepwise Selection)	**0.08**
Naive Bayes	4.44
Random Forest (nforest=11, ntree=500)	0.20

Best Model for OREB:

At first, we expected that ridge regression on all predictors would be our best model for OREB, as that was the case for spread and total. However, ridge regression among the predictors learned from stepwise selection was our best model. Following the same formula as **Figure 2.2** and the same hyperparameters, ridge regression turned out to be our best model once again. This makes sense because multicollinearity is an ever-present problem with the data we chose to characterize a basketball game. However, we feel it necessary to choose the stepwise selection model because it strategically eliminates variables irrelevant to rebounding, such as steals, assists, etc. In our final model, bully ball was not significant, which goes to show that although physicality is important, player positioning, circumstance, and luck are all important measures

that we could not quantify well. Nevertheless, ridge regression proved to be our most powerful model in predicting OREB as well as spread and total.