NBA 2023 Predictions:

Spread, Total Score, and Offensive Rebounds

Group 15 Authors: David Bruss, Eliza Hancock, Yulim Kim, Jacob Schick, and Rohan Venkatraman

April 03, 2023

STOR 538

Dr. Mario Giacomazzo

1. Data Information

1.1 Cleaning Summary

We acquired our data from the nbastatR package in R from Alex Bresler; while we considered using the Kaggle datasets provided, they were outdated as they had not been updated since December. We quickly concluded that recent data would be the most helpful to predict the games on the last five days of the NBA regular season. Therefore using the nbastatR package, we could use its built-in functions, such as game_logs, to collect each team's basic stats from the beginning of the 2023 season on October 18th until March 28th. We only used data from the 2022-23 regular season because the NBA constantly changes. Using historical data when players often change teams would not benefit our models. Realizing that basic stats would not be sufficient to provide good predictions for spread, total, and offensive rebounds, we decided to also use the box_scores function in the nbastatR package. At first, we could not use the box_scores function, so we web-scraped nba.com advanced stats. However, these stats had been aggregated to season averages and could not be used alongside the game logs data as it would not tell a story for each game. If we could not get the box_scores function to work, we planned to create linear weights on the opponent's defensive rating to ensure we would not have the same prediction for every game by a team.

However, we eventually got the box_scores function to work to pull in more comprehensive tables such as Four Factors, Miscellaneous, and regular Advanced statistics on a game-by-game level for each team. Each of these tables was created separately and merged by *idGame* and *idTeam*; this produced one large table with 2,270 observations and 72 variables. To incorporate opponent statistics into each row, we included the opponent's *countDaysRestTeam*, *drtg*, *netrtg*, *possessions*, and *ratioPIE*. To achieve this, we split the large table into the home and away teams tables, then merged them with both merge orders (home games merged with away games, and away games merged with home games.) After merging, we only kept the necessary opponent variables and renamed the merging suffixes for each newly merged table. Lastly, we appended both merged tables into a new table sorted by *idGame* to ensure there were two rows per game. This table contained the same 2,270 observations but with 59 variables—incorporating the opponent variables while also removing the unnecessary team variables such as *isWin* because this is what our model will figure out, *minutesTeam* because every game has the same number of minutes unless of overtime which also cannot be accounted for in our prediction, and *pctTOV_Opp* because it was a duplicated

value coming from the Four Factors table (note that there were also other variables removed for similar reasons). This table produced a master table in which each row represented a team's statistics for a game, along with a few of their opponent's statistics—this would allow our model to have more information to make predictions. We chose to split each game by team rather than combining them into a single observation because we created our model to predict each team's total points and offensive rebounds. Then we would combine the individual team predictions with their opponent's to produce the spread, total, and total offensive rebounds.

1.2 Variables

Our master table had the following variables. Their definitions can be found in Appendix A:

idGame, dateGame, idTeam, slugTeam, numberGameTeamSeason, nameTeam, isB2B, isB2BFirst, isB2BSecond, locationGame, countDaysRestTeam, fgmTeam, fgaTeam, pctFGTeam, fg3mTeam, fg3aTeam, pctFG3Team, fg2mTeam, fg2aTeam, pctFG2Team, ftmTeam, ftaTeam, pctFTTeam, orebTeam, drebTeam, trebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, ptsTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pfd, pctAST, pctOREB, pctDREB, pctTREB, pctTOVTeam, pctEFG, pctTS, pctUSGE, ortg, netrtg, ratioASTtoTOV, ratioAST, pace, pacePer40PACE_PER40, possessions, ratioPIE, countDaysRestTeam_Opp, drtg_Opp, netrtg_Opp, possessions_Opp, ratioPIE_Opp

1.2.1 Handling Variable Anomalies

Once we established our variables for the master table, there were still some variable anomalies that needed to be handled. At the beginning of the season, *countDaysRestTeam* and *countDaysRestTeam_Opp* had outlier values of 120, but instead of deleting those games entirely, we decided to replace them with 4, which was approximately the number of days each team had since their last preseason game. This was because preseason games still allowed players to acclimate to an actual game atmosphere after a long off-season break. Additionally, we noticed that the *nameTeam* "LA Clippers" was used instead of "Los Angeles Clippers", which led to errors when merging with the predictions table. Therefore, we decided to standardize all instances of "LA Clippers" to "Los Angeles Clippers".

1.2.2 Engineering New Variables

To improve our model, we wanted a way to measure how much of an advantage a team had when playing at home—historically, teams often play better in front of their home crowd. Home-court advantage is more prevalent in the playoffs when stadiums are packed to support their home crowd; however, we wanted to measure its importance during the regular season.

$$Home\ Court\ Advantage\ (HCA)\ = \\ mean(Spread\ at\ home\ games)\ -\ mean(Spread\ at\ away\ games)$$

Using this metric we can measure how much better teams are at out-scoring their opponents when playing at home versus playing away. Ideally, a good team will have a negative mean spread at away games because the spread is measured as the home minus away score. Therefore the best home teams in the NBA will have positive *HCA* scores, and the worst home teams with negative *HCA* scores, while teams equally good home and away will have scores close to zero. The goal for this variable was to adjust our final score predictions based on this value.

1.2.2.1 Creating Table for Predictions

To create our predictions, we needed to engineer a new table as input to our model. We concluded that a team's last five games often indicate their current form. Additionally, to ensure that our predictions for each team would not be the same, we also incorporated their season matchup history stats into this new table for predictions. Factors such as playing style, player matchups, and coaching strategies can all be influenced by previous matchups, and therefore it is vital to measure the opponent's impact on the game. To create our data for predictions, we first counted the number of previous matchups between the two teams playing and added it to five (representing the last five games.) For example, if the Lakers had played the Warriors twice in the season already, we would add two to five for a total of seven. This total determines how much weight to give to each statistic for the team. We would then multiply each statistic in each of the last five games by 1/7 and also multiply each statistic in the last two matchups by 1/7. This ensures that the statistics from the most recent games and previous matchups are given equal weight. Finally, we would sum the weighted values by statistic to create a weighted average statistic incorporating a team's recent trend and matchup history.

To gather data for the last five games, we used our master dataset. We grouped by team name to create a rolling average of a team's statistics during a five-game span at any point during the season. For our purposes, we only selected the maximum date for each team to get their most recent five-game history.

Gathering the data for each matchup's history was a little more complicated. First, we had to get a list of matchups, and this was achieved by splitting the master table into home teams and away teams tables and then merging them with a new column *matchup*, an alphabetical concatenation of the two team names. Then each statistic was summarized by *matchup*, and a total count of each matchup was collected. The table now consists of two rows per matchup, where each row had the totals (no need for means because of the weighting with the last five matchups) for each statistic for teams A and B, respectively.

A different predictions table was created for points and offensive rebounds because we tailored the table to the variables necessary after choosing the best model.

1.3 Handling Outliers

When observing the points and offensive rebounds by teams in each game, we wanted to ensure no outliers would affect our model. Therefore using the IQR method of detecting outliers, rows from the full data set were removed before creating predictive models. Focusing on outlier-removed data would eliminate any variance caused by anomaly games, making our predictions more concrete.

For *ptsTeam*, it was decided that any game scoring under 84 or above 145 points was classified as an outlier score. IQR was calculated by subtracting the first quartile of the spread of points from the third quartile and was a value of 15. We removed nineteen rows from our original data set based on these bounds.

$$IQR = 122 - 107 = 15$$
$$Lower\ Outlier\ Bound = 107 - 1.5(15) = 84.5$$
$$Upper\ Outlier\ Bound = 122 + 1.5(15) = 144.5$$

For the *orebTeam* variable, we were able to remove any rows of data with *orebTeam* values of 0 and any greater than 21. The IQR value calculated was 5. We discarded seventeen rows of data as outliers.

$$IQR = 13 - 8 = 5$$
$$Lower\ Outlier\ Bound = 8 - 1.5(5) = 0.5$$
$$Upper\ Outlier\ Bound = 13 + 1.5(5) = 20.5$$

We opted to remove outlier rows instead of entire games because our focus was on identifying the most significant variables for predicting points and offensive rebounds. By removing only the rows where a team's statistics were unusually high or low, we could eliminate the effects of these outliers on our analysis while retaining valuable information from the other rows in the game. This approach allowed us to improve the accuracy of our predictions without discarding entire games.

2. METHODOLOGY

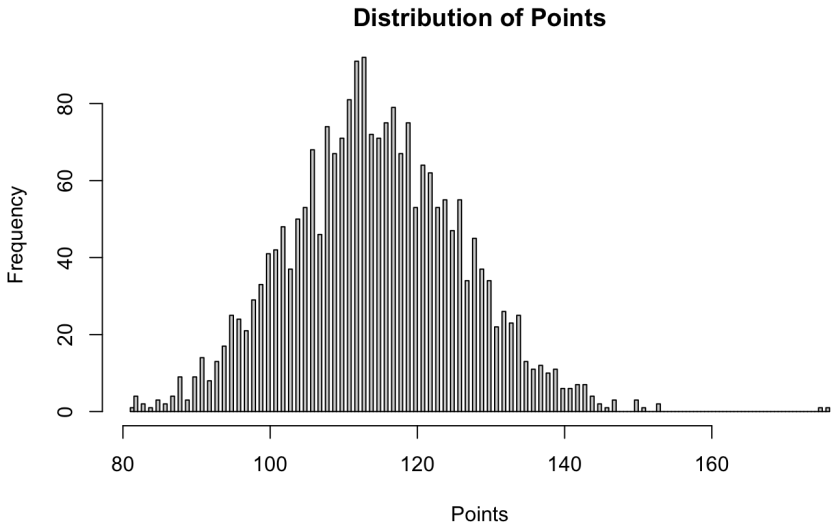
2.1 Points

To predict the spread and total points, we created one general model for both. This model predicted points per team, and we used these outputs to calculate the two variables we are looking for, as they are direct computations from points themselves.

$$Spread = Home\ Team\ Points - Away\ Team\ Points$$
$$Total\ Points = Home\ Team\ Points + Away\ Team\ Points$$

Before deciding which models to test in predicting points, we looked at the distribution of points for NBA teams in the 2023 season. As seen below, the distribution of this variable appears to be normal. Therefore, we chose modeling techniques that highlight the distribution and attributes of the variable itself. With the fairly normal distribution of the *ptsTeam* variable, we decided it would be beneficial to investigate a linear model. We also considered a Poisson model since the value we are trying to predict is a count variable.

Figure 1: Distribution of the Points by Team



Some possible outliers can be noticed within the graph. These were removed during the data cleaning steps taken prior to modeling using IQR outlier detection methods.

2.1.1 Points Variable Selection

Although unnecessary variables were removed in the data cleaning process, we conducted a second round of assessing a variable's importance within the data set on which our model would be produced. Initially, a correlation matrix of all variables was made to dive into the relationships between many of the variables. We found that variables were highly correlated if they explained similar things—for example, *orebTeam* and *derebTeam* were highly correlated with *trebTeam* because *trebTeam* = *orebTeam* + *derebTeam*. In this case, *trebTeam* was removed because it was not as specific as having the two offensive and defensive rebound statistics. The same reasoning was used for *pctFGTeam*, since *pctFGTeam* = *pctFG2T* + *pctFG3Team*.

This process of looking into how variables were calculated was used for free throw and field goal statistics, where the attempts total, shot made total, and percentage of each were within the data set. For these instances, the percentages were kept, as we found that knowing the efficiency of these shots was more important than the count of attempts and made baskets.

Looking at the correlation between variables, we were able to look at groupings of variables with high correlations with each other. After researching how certain advanced statistics were calculated, we determined which variables would be beneficial to keep. Narrowing down on variables that had high correlations with each other ensured that there would be a lack of multicollinearity and overfitting within our models.

The final data set for our models for Points contains 22 variables: *isB2B, isB2BFirst, isB2BSecond, locationGame, countDaysRestTeam, pctFTTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pctUSGE, pace.*

2.1.2 Points Training and Testing Sets

Prior to creating models, we split our data into Test and Train models, where 70% of the data was used to train the models we created, and the rest was used to test the accuracy. These test and train data sets were derived from the 2023 game-by-game log data set, and we utilized random seed generators to create unbiased selections for the two.

The data sets contained the same 22 variables: *isB2B, isB2BFirst, isB2BSecond, locationGame, countDaysRestTeam, pctFTTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pctUSGE, pace.*

2.1.3 Points Models

The first model created was a simple linear regression model. To find our best linear model, we started with an initial model that utilized all variables in the dataset. The variables used were as follows: *isB2B, isB2BFirst, isB2BSecond, locationGame, countDaysRestTeam, pctFTTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pctUSGE, pace*. A multiple R-squared value of 0.9022 and an adjusted R-squared value of 0.9008 were found, which is not awful. However, we believed that the model as a whole would be improved with variable selection.

Using stepwise, backwards, and forwards variable selection methods, the models produced utilized the same variables. These variables included the following: *pctTS, orebTeam, pace, tovTeam, pfTeam, pctUSGE, ptsPaint, drebTeam, ptsSecondChance,*

astTeam, pctFG2Team, rateFTA, stlTeam, ptsOffTOV, ptsFastBreak. From the 22 original variables, 15 were chosen to be significant to the model. A new multiple R-squared value of 0.9021 and an adjusted R-squared value of 0.9011 were found. Although this was not a large difference from the original full model, we were more confident that the significance of the predictors benefited us in predicting points. Using our testing data set, we produced an RMSE value of 3.436834.

The next two models we created were built upon the previous linear regression techniques. The lambda values were hyper-tuned for each to produce the most effective model. Linear ridge regression models are more sensitive to variance within data—such as outliers. Although we went through the process of removing outliers based on the IQR method previously, we believed the model could tune in more to anomalies we failed to catch. The optimal lambda value for this model was 0.007943, and we utilized all 22 variables. When testing the model, we calculated an RMSE value of 3.446409. A linear LASSO regression model utilizes shrinkage to determine a smaller set of parameters while improving the model's accuracy. With a lambda value of 0.0316227, the variables selected through this method included: *countDaysRestTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pctUSGE, pace* - just 17 of the 22 variables from the original data set. When testing the predictions on our test data set, we produced an RMSE value of 3.433345.

We used a similar thought process for the next set of models. This time, we focused on Poisson models to account for the fact that the points variable is a count that occurs within a set amount of time. Starting with the original full data set of 22 variables, we used ANOVA Chi-Sq measurements to determine the significance of the variables within the Poisson model. From the 22 variables, the model selected 16 to be significant: *locationGame, pctFTTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsPaint, pctTS, pace.* After testing this model on our testing set, we produced an RMSE of 3.570859.

Ridge and LASSO techniques were utilized alongside the Poisson model as well. The Poisson ridge regression model utilized all 22 variables from the data set with a lambda value of 0.1. The RMSE value, when tested on the testing data set, was 3.474666. The Poisson LASSO model utilized a lambda value of 0.03981 and 18 of the 22 original variables. The significant predictors within the Poisson LASSO model included: *locationGame, countDaysRestTeam, pctFG2Team, orebTeam, drebTeam, astTeam, stlTeam,*

blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pctUSGE, pace. An RMSE value of 3.455833 was found.

We found that the linear LASSO regression model was the best of these previous models to produce the lowest RMSE value. With this model, we wanted to incorporate adding the home court advantage (*HCA*) variable we created by finding the best linear weight to apply to the variable. We believed that *HCA* would play a role in points scored by a team—as it is historically shown that teams often play better in front of their home crowd. To try and account for *HCA*, a linear regression model was used to predict points, with our two predictors being *HCA* and predictions from the linear LASSO model. By using a linear regression model, we determined the correct coefficients for each variable. We used a subset of our test data that included only home games, as *HCA* would only be applied to teams that were the home team. After creating our model, *HCA* showed to be insignificant to the model, with a p-value of 0.436. We still calculated the RMSE values for the predictions made with *HCA* linear weights but found a value of 3.727575, showing it did not improve our predictions.

2.1.4 Points Best Model

Spread and total used the linear LASSO regression model to predict points, as it gave the best RMSE of all the modeling techniques tested. A lower RMSE value signifies better modeling performance and accuracy of the predictions made. The simple linear regression models using stepwise and backwards variable selection methods had an RMSE value close to the one linear LASSO produced. We cannot compare R-squared metrics between the linear and LASSO regression models, as R-squared can not be accurately applied to LASSO regressions. So RMSE became our standard in choosing the best modeling method (see next page).

Table 1: RMSE Analysis of Points Prediction Models

Points	Model	RMSE
	Linear Regression (Stepwise, Backwards Selection)	3.436834
	Linear Ridge Regression (λ = 0.007943)	3.446409
	Linear LASSO Regression (λ = 0.0316227)	**3.431431**
	Poisson Regression	3.570859
	Poisson Ridge Regression (λ = 0.1)	3.474666
	Poisson LASSO Regression (λ = 0.03981)	3.455833
	Linear Modelling on *HCA* and Linear LASSO Predictions	3.727575

2.1.5 Results from Linear LASSO Regression Model on Points

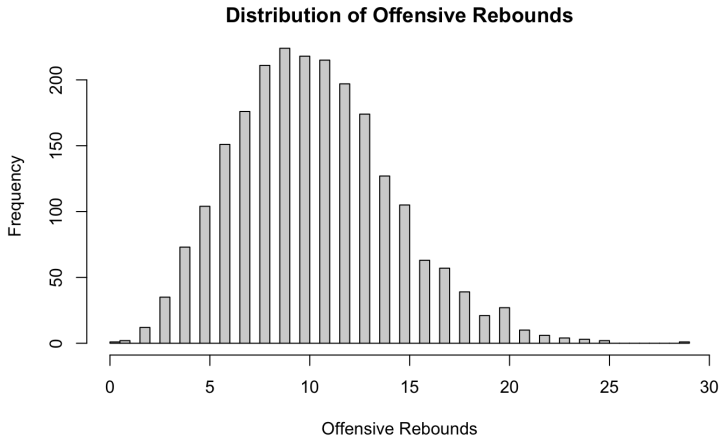
(Intercept)	-31.28641888	blkTeam	0.02791892
isB2BTRUE	.	tovTeam	-0.96836503
isB2BFirstTRUE	.	pfTeam	0.22550042
isB2BSecondTRUE	.	rateFTA	5.39575946
locationGameH	.	ptsOffTOV	0.04497216
countDaysRestTeam	0.02973829	ptsSecondChance	0.13631411
pctFTTeam	.	ptsFastBreak	-0.02688079
pctFG2Team	-9.01736615	ptsPaint	0.09044171
orebTeam	0.94690652	pctTS	174.19998743
drebTeam	0.14203891	pctUSGE	-278.24717619
astTeam	0.17077591	pace	0.85691475
stlTeam	0.10468047		

The variables with the largest impact on predicting points include *pctTS* and *pctUSGE.* These variables make sense to have the biggest impact on the model, as true shooting percentage measures shooting efficiency—a larger percentage leads to larger points. The coefficient on *pctUSGE* shows that higher usage percentages are penalized—which makes sense as a high percentage may indicate that one player is responsible for the majority of points scored rather than a variety of players scoring points. Having more players scoring points, rather than one key player, often leads to higher scores.

2.2 Offensive Rebounds

To predict offensive rebounds, we took a similar approach to predicting points when creating models. Before forming models, we again looked at the distribution of the variable *orebTeam*, which represents offensive rebounds, from our original data set. As seen below, there is a normal distribution. At first glance, we hypothesized a Poisson model would be the best fit for predicting offensive rebounds, as it is a count variable within a set time frame, and the rate at which they happen is considerably lower than scoring points. Because of the similar distribution of the variable, it allowed us to follow the procedures we took in modeling points—mainly looking at linear regression models and Poisson models.

Figure 2: Distribution of Offensive Rebounds by Team



Although the distribution of offensive rebounds is relatively normal, there are some outliers. As stated in the previous section, these were removed using the IQR outlier detection method.

2.2.1 Offensive Rebounds Variable Selection

Prior to creating training and testing data sets, variables were once again combed through to ensure there would be a lack of overfitting and multicollinearity within our models.

Specifically, *pctTS* and *ortg* were concerning because of their similarities in what they measured. To detect if either or both of these variables needed to be removed, we created a linear regression model utilizing all variables and looked at variance inflation factors (VIF) values to detect multicollinearity. Normally, VIF values above five are considered to be concerning, and the values of *pctTS* and *ortg* were 13.328506 and 12.631023, respectively.

To determine which of the two was the main concerning variable regarding multicollinearity, two data sets were created—each containing all the other variables and only one of the two variables we wanted to investigate. Linear regression models were made with each of the two data sets, and the respective VIF values were looked into. When only *pctTS* was removed from the data set, *pctFG3Team* had a VIF value of 5.385740, and *ortg* still had a large VIF of 9.725378. Looking at the linear regression model where only *ortg* was removed, all VIF values were reasonably under five. From this investigation, prior to our modeling process, we determined that removing *ortg* from the data set would help with multicollinearity issues when predicting offensive rebounds.

Using a correlation matrix, other variables were removed based on relationships with other variables. For example, *trebTeam* was removed because *orebTeam* (the variable we are predicting) is directly used within the computation of the variable. The percentages of statistics, such as *pctFG3Team, pctFTTeam,* and *pctFG2Team,* were kept over the counts on made and attempted shots as we did previously. The variable *pctFGTeam* was removed once again as well, since *pctFGTeam = pctFG2Team + pctFG3Team*, and having all three variables would be unnecessary.

The data sets contain 26 variables: *isB2B, isB2BFirst, isB2BSecond, locationGame, countDaysRestTeam, pctFG3Team, pctFTTeam, pctFG2Team, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pfd, pctTS, pctUSGE, pace, ratioPIE, countDaysRestTeam_Opp, ratioPIE_Opp.*

2.2.2 Offensive Rebounds Training and Testing Sets

Utilizing random seed generators, we created two unbiased samples from the outlier-removed data to serve as our test and train data sets. Similar to the training and testing sets for the points models, 70% of the data served to train our prediction models, and 30% served to test the performance of the created models.

The data sets contain 26 variables: *isB2B, isB2BFirst, isB2BSecond, locationGame, countDaysRestTeam, pctFG3Team, pctFTTeam, pctFG2T, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pfd, pctTS, pctUSGE, pace, ratioPIE, countDaysRestTeam_Opp, ratioPIE_Opp*.

2.2.3 Offensive Rebounds Models

Beginning with a basic linear model, our first model contained all variables from the dataset. The variables are as follows: *isB2B, isB2BFrst, isB2BSecond, locationGame, countDaysRestTeam, pctFG3Team, pctFTTeam, pctFG2T, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, pfTeam, rateFTA, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pfd, pctTS, pctUSGE, pace, ratioPIE, countDaysRestTeam_Opp, ratioPIE_Opp.* This model has a multiple R-squared value of 0.6318 and an adjusted R-squared value of 0.6257. Although these values are lower than the R-squared values from the points linear regression models, we intentionally removed variables during the data cleaning step we believed would lead to overfitting of the model. The lower R-squared value is indicative of the absence of overfitting.

We created our next linear regression model using stepwise, forwards, and backwards variable selection methods. The models produced using these methods contained the same variables: *ptsSecondChance, pctTS, ratioPIE, ptsPaint, pctFG2Team, tovTeam, drebTeam, stlTeam, astTeam, ptsOffTOV, ptsFastBreak, blkTeam, ratioPIE_Opp, pctUSGE.* Only 14 of the original 26 variables were found to be statistically significant. A multiple R-squared value of 0.6311 and an adjusted R-squared value of 0.6278 were noted—a slight improvement from the full linear regression model. We recorded an RMSE value of 2.436942 when predicting the test data set.

The next model we produced was a linear ridge regression model. As previously mentioned, ridge regression models are often useful for dealing with data sensitive to outliers—and even though we removed anomalies earlier in our modeling process, we predicted that any variance in the data not picked up by the IQR method would be dealt with within this model. An optimal lambda of 0.031622 was used, and all 26 variables

were included in the model. After using testing data to create predictions, we found an RMSE value of 2.438655.

The last of the linear regression methods we used to predict offensive rebounds was the linear LASSO regression model. Using this method, we used an optimal lambda of 0.015849 for the model, and 17 variables were chosen to be the most significant. The variables are as follows: *locationGame, countDaysRestTeam, pctFTTeam, pctFG2Team, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pctUSGE, pace, ratioPIE.* After creating predictions from the test data set, we produced an RMSE value of 2.435226.

Poisson regression models were focused on in our next round of prediction models. Starting with a Poisson model with all 26 variables in the data set, we utilized ANOVA Chi-squared test methods to find the most significant variables within the training data set. 11 variables were chosen: *pctFG3Team, pctFTTeam, pctFG2Team, stlTeam, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pctTS, pace, ratioPIE.* This model resulted in an RMSE value of 2.56371 after predictions were made using the test data set.

Ridge and LASSO regression methods were once again utilized alongside Poisson modeling. The Poisson ridge regression model used a lambda value of 0.079432 and once again used all 26 variables of the data set. We calculated an RMSE value of 2.496905. Using Poisson LASSO regression modeling, we used a lambda value of 0.0158489, and 21 variables were selected to be in the model. The variables are: *isB2B, isB2BFirst, locationGame, countDaysRestTeam, pctFTTeam, pctFG2Team, drebTeam, astTeam, stlTeam, blkTeam, tovTeam, ptsOffTOV, ptsSecondChance, ptsFastBreak, ptsPaint, pfd, pctTS, pctUSGE, pace, ratioPIE, countDaysRestTeam_Opp*. The RMSE value calculated off of predicted values was 2.489072.

2.2.4 Offensive Rebounds Best Model

As discussed previously, when determining the best model for the points model, we utilized the RMSE scores to choose the best model for predicting offensive rebounds. The model with the lowest RMSE was once again the linear LASSO regression model. We hypothesized before model creation that Poisson models would best predict offensive rebounds. Although they had RMSE values relatively close to the linear LASSO's, we ultimately went with the model with the smallest RMSE.

Table 2: RMSE Analysis of Offensive Rebound Prediction Models

Offensive Rebounds	Model	RMSE
	Linear Regression (Stepwise, Backwards Selection)	2.436942
	Linear Ridge Regression (λ = 0.031622)	2.438852
	Linear LASSO Regression (λ = 0.015849)	**2.435226**
	Poisson Regression	2.56371
	Poisson Ridge Regression (λ = 0.079432)	2.496905
	Poisson LASSO Regression (λ = 0.0158489)	2.489072

2.2.5 Results from Linear LASSO Regression Model

(Intercept)	13.74936078	pfTeam	.
isB2BTRUE	.	rateFTA	.
isB2BFirstTRUE	.	ptsOffTOV	-0.01788325
isB2BSecondTRUE	.	ptsSecondChance	0.36437907
locationGameH	0.06371244	ptsFastBreak	0.01860112
countDaysRestTeam	0.03196025	ptsPaint	0.07848160
pctFG3Team	.	pfd	.
pctFTTeam	0.05717461	pctTS	-23.55440690
pctFG2Team	-13.87009741	pctUSGE	28.34648155
drebTeam	-0.10213827	pace	-0.01351053
astTeam	0.03615726	ratioPIE	12.79692226
stlTeam	-0.09722087	countDaysRestTeam_Opp	.
blkTeam	-0.03502325	ratioPIE_Opp	.

The variables with the largest weight in the model are *pctTS* and *pctUSGE*. These variables also had the largest coefficients in the points model but with opposite signs. Higher true shooting percentages are penalized—as making more shots will lead to fewer opportunities for offensive rebounds. A high usage percentage is now positively weighted—higher percentages mean more shots are taken overall.

3. CONCLUSIONS

Through our methodology of creating predictive models for Spread, Total Points, and Offensive Rebounds for NBA games, the linear LASSO regression model proved to be superior. However, we recognized critiques and limitations of our predictions that can be delved into in the future. First, there were various aspects of NBA games other than game statistics that could have been added to improve the accuracy of our prediction models. Player and injury data would play a crucial role in a team's scoring output, especially with teams with injured star players or ones that rest often. The importance of games could have been considered as well. At this point in the season, teams have either clinched the playoffs, been eliminated, or continue to fight for a playoff spot. Because of this, each game has a different meaning for each team. Games could have possibly been weighted to account for this. Additionally, an analysis of referees would show bias between teams, as the referees' tendencies (pace of play, foul calls, team records, etc.) are found to be unfair. Unfortunately, we are not given the list of who is refereeing the game until the day of the game.

Looking at our modeling processes as a whole, finding a way to utilize the variable we created, *HCA*, would have been ideal if given more time. Home court advantage is known to often be an influential component of a team's performance. Although we found it to be statistically insignificant within our modeling techniques, different methods could be tested to incorporate them into the predictions. Our model performances were also based solely on RMSE scores when determining which could be best for predictions. Looking into other statistical measures, such as mean absolute error (MAE), could have been beneficial. Different performance limitations could have been highlighted with various measures of predictive performance. With more time, more advanced explorations of the data may have resulted in different outcomes with our models.

Appendix A (Variable Definitions):

Variable Name	Description
idGame	Unique identifier for each game
dateGame	Date of the game
idTeam	Unique identifier for each team
slugTeam	Abbreviated team name
numberGameTeamSeason	The number of games the team has played in the current season
nameTeam	Team name
isB2B	Boolean indicating whether the game is part of a back-to-back series for the team
isB2BFirst	Boolean indicating whether the game is the first game of a back-to-back series for the team
isB2BSecond	Boolean indicating whether the game is the second game of a back-to-back series for the team
locationGame	Home or away game for the team
countDaysRestTeam	Number of days of rest for the team prior to the game
fgmTeam	Field goals made by the team
fgaTeam	Field goals attempted by the team
pctFGTeam	Field goal percentage for the team
fg3mTeam	3-point field goals made by the team
fg3aTeam	3-point field goals attempted by the team
pctFG3Team	3-point field goal percentage for the team
fg2mTeam	2-point field goals made by the team
fg2aTeam	2-point field goals attempted by the team
pctFG2Team	2-point field goal percentage for the team
ftmTeam	Free throws made by the team
ftaTeam	Free throws attempted by the team
pctFTTeam	Free throw percentage for the team
orebTeam	Offensive rebounds by the team
drebTeam	Defensive rebounds by the team

trebTeam	Team rebounds by the team
astTeam	Assists by the team
stlTeam	Steals by the team
blkTeam	Blocks by the team
tovTeam	Turnovers by the team
pfTeam	Personal fouls by the team
ptsTeam	Total points scored by the team
rateFTA	Rate of free throw attempts per field goal attempt
ptsOffTOV	Points off turnovers for the team
ptsSecondChance	Second-chance points for the team
ptsFastBreak	Fast break points for the team
ptsPaint	Points scored in the paint by the team
pfd	Personal fouls drawn by the team
pctAST	Percentage of field goals that were assisted by the team
pctOREB	Percentage of available offensive rebounds obtained by the team
pctDREB	Percentage of available defensive rebounds obtained by the team
pctTREB	Percentage of available team rebounds obtained by the team
pctTOVTeam	Percentage of possessions resulting in a turnover by the team
pctEFG	Effective field goal percentage for the team
pctTS	True shooting percentage for the team
pctUSGE	Percentage of team plays used by the player when they were on the floor (averaged the players USGE)
ortg	Offensive rating for the team
netrtg	Net rating for the team
ratioASTtoTOV	Ratio of assists to turnovers for the team
ratioAST	Ratio of field goals made that were assisted by the team
pace	Pace of the game for the team
pacePer40PACE_PER40	Pace of the game per 40 minutes for the team
possessions	Number of possessions for the team
ratioPIE	Ratio of a team's player impact estimate (PIE) to the total PIE of both

	teams in a game. Measures player's overall statistical contribution (averaged players PIE)
countDaysRestTeam_Opp	Number of days of rest for the opposing team prior to the game
drtg_Opp	Defensive rating of the opposing team
netrtg_Opp	Net rating for the opposing team
possessions_Opp	Number of possessions for the opposing team
ratioPIE_Opp	Ratio of the opposing team's player impact estimate (PIE) to the total PIE of both teams in a game. Measures player's overall statistical contribution (averaged players PIE)