2021 NBA Outcomes

Predicting Spread, Total Points, and Total Offensive Rebounds

Authors: Kelsey Johnson, Julianna Cybrynski, Crystal Baker, Vasu Gupta

I. Data Information

A. Data Cleaning and Joining

The five given datasets all contained valuable information, so we wanted to combine data of interest together into one dataset. Some data was on the team level while other data was on the player level, so we had to spend some time properly joining the data. An example of this is the offensive rebound (OREB) data, which was provided at the player level, however, we wanted it at the team level for each game. To do this, we grouped players by team and game and added together their OREBs, giving us a total number of OREB for both teams in each game. Some missing values were noticed, but upon investigation, they appeared to be from players who did not play in that game, so the missing values were replaced with 0. We then joined this with the team level data for both the home team and the away team in every game. This gave us two variables, OREB_home and OREB_away, which we summed together to create OREB, the total number of offensive rebounds in each game. We followed a very similar procedure for assists (AST), steals (STL), blocks (BLK), and turnovers (TO) to use the player level data to create home and away totals for each game as well as a total sum. We also combined existing variables to make the total and spread variables we will be creating models to predict. For total, we added together the columns for home points and away points, creating a new "total" column using the mutate function in R. We followed a similar process to create the variable "spread", this time subtracting away points from home points.

Some games had values that fit with the rest of the data for most categories, but one category in which they did not fit at all. This is explored in the methodology section for the offensive rebounding model, where some games had extremely high or low values for OREB. The maximum was 84, much higher than the rest of the data, and the minimum was 5, much lower than the rest of the data. To avoid the influence large outliers can have on the total model, less than 1% of data from each end was removed to exclude those rare events, ending up with a range for OREB between 10 and 36.

B. Engineered Variables

One variable we created that was not in the original dataset was the turnover differential. We believed this would be a good measure of the relative ability of the teams to maintain possession. We calculated this by subtracting the number of turnovers made by the away team from the number of turnovers made by the home team. We also created differentials for steals in the same way for similar reasons, showing the relative amount of aggressive actions led to turnovers as a measure of physicality of defensive play. We also created a differential for blocks as a measure for the relative defensive skill and ability to block shots during the games. These differentials were significant predictors in our models for offensive rebounds. Another differential variable that was created was for defensive rebounds as a way to measure the relative success of rebounding on the defensive end, which was not significant in our final OREB model.

Another variable we added to our dataset was Win percentage for both teams. Win Percentage is the ratio of wins a particular team has at the time of data collection. In order to correctly include this variable, we had to consider the date of the game and the date that the variable was recorded. Once those were mapped out, we calculated Pace Factor and Possessions for both teams. Pace Factor is an advanced statistic used by the NBA to calculate the number of possessions per game a team uses per game. We calculated this using the variables from our datasets and in the process created Possessions, which estimates the number of times a team is in possession of the ball. The complete equation for Pace Factor is below.

$$48 \times ((Team\ Possessions + Opponent\ Possessions) \div (2 * (Team\ Minutes\ Played \div 5)))$$

C. Outside Data

In order to gather more updated information on how teams are performing in the 2021 season, we collected data from outside resources. From NBA.com, we gathered more information on offensive rebounds for all 30 NBA teams this season, including their average OREBs per game, contested OREBs, contested OREB percentage, OREB chances, percentage of OREB chances that were successful, deferred OREB chances, and adjusted OREB chance percentage (https://www.nba.com/stats/teams/offensive-rebounding/). These statistics will likely be helpful in the model predicting total OREB by providing more specific information about each team's current offensive rebounding abilities and averages than we currently can gather

from the original data. We were able to join this information with relative ease by using the team name to connect the datasets and renaming the variables with “home” and “away” identifiers to avoid having multiple variables of the same name.

We entitled this updated dataset containing all the new variables “data1”. From data1, we split the data into a training set and test set, which contained 70% and 30% of the updated data respectively. We entitled these new datasets “train” and “test”.

II. Methodology for Spread

The first outcome we sought to predict is Spread, which is the number of points the home team scores minus the number of points from the opponent. In preparing the data for prediction we first used backward and forward stepwise regression to find the best predictors of Spread. In an attempt to create variables that would help in predicting spread, we developed the variables, Pace Factor, Possessions, and Most Recent Win Percentage for both teams. This process was done by running linear regression on all of the predictors or one predictor and removing or adding one variable at a time to determine if it is statistically significant. Both techniques yielded similar results, but we decided to use the results of the forward regression because it provided variables with p-values close to zero. Forward regression resulted in the following variables: Field Goal Percentage, Field Goals Attempted, Free Throw Percentage, Free Throw Differential, 3-Point Field Goal Percentage, Assist Differential, Block Differential, Defensive Rebound Differential, Turnover Differential, and Home Team Outcome. Surprisingly, the Win Percentage, Pace Factor, or Number of Possessions were statistically significant in predicting Spread.

```
Call:
lm(formula = Spread ~ FG_PCT_home + FG_PCT_away + FG3_PCT_home +
    FG3_PCT_away + FT_PCT_home + FT_PCT_away + FTA_diff + TO_diff +
    BLK_diff + AST_diff + HOME_TEAM_WINS + FGA_home + FGA_away +
    DREB_diff, data = spread)

Residuals:
     Min       1Q   Median       3Q      Max
-25.2311  -2.1494  -0.0365   2.1191  17.5312

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)
(Intercept)    -3.891e+00  4.265e-01   -9.123   <2e-16 ***
FG_PCT_home     1.214e+02  7.688e-01  157.870   <2e-16 ***
FG_PCT_away    -1.213e+02  7.696e-01 -157.617   <2e-16 ***
FG3_PCT_home    1.748e+01  2.222e-01   78.663   <2e-16 ***
FG3_PCT_away   -1.692e+01  2.236e-01  -75.666   <2e-16 ***
FT_PCT_home     1.802e+01  2.354e-01   76.561   <2e-16 ***
FT_PCT_away    -1.722e+01  2.276e-01  -75.634   <2e-16 ***
FTA_diff        5.600e-01  3.942e-03  142.054   <2e-16 ***
TO_diff        -1.632e-01  8.179e-03  -19.954   <2e-16 ***
BLK_diff        9.983e-02  6.848e-03   14.579   <2e-16 ***
AST_diff        1.632e-01  4.369e-03   37.354   <2e-16 ***
HOME_TEAM_WINS  3.079e+00  7.366e-02   41.805   <2e-16 ***
FGA_home        7.302e-01  5.206e-03  140.282   <2e-16 ***
FGA_away       -7.124e-01  5.145e-03 -138.458   <2e-16 ***
DREB_diff       1.589e-01  6.729e-03   23.619   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.397 on 24174 degrees of freedom
Multiple R-squared:  0.9354,    Adjusted R-squared:  0.9353
F-statistic: 2.499e+04 on 14 and 24174 DF,  p-value: < 2.2e-16
```

Once the features were solidified, we began the model selection process.We used the results of the linear regression model as a baseline. After splitting the data into 70 % train and 30% test datasets, our linear regression model had a root mean squared error (RMSE) of 3.39. From there, we began considering which models would work best for the data. First, we noticed a pattern in the coefficients from the linear regression model. They displayed a pattern in which the error of the home team was positive, while many of the coefficients of variables representing the away team were negative. This suggested that a polynomial transformation may be a better fit. In response, we developed a second degree polynomial regression model that fit the data better as it has a RMSE of 3.02. This is a difference of 0.30, but we remained cautious with this model due to the possibility of overfitting.We also developed a Stochastic Gradient Descent linear model as it works to fit linear models using loss functions and works well with datasets as large as the one we used for predicting Spread. Initially the model performed very poorly, with a RMSE of 3.48. However, after using a polynomial transformation with a degree of 2, and scaling the data, it produced a RMSE of 3.14.

This RMSE was still relatively high so we developed two Decision Tree models, but both of these models resulted in very high RMSEs as high as 5.66. In effort to improve the results of the Decision Tree model we developed a Random Forest Model. As expected, the results improved, but not enough to compare to the baseline as the random forest produced a RMSE of 4.84.

The last model we developed was a Multi-Layer Perceptron (MLP) Regressor. This model is a supervised learning neural network that is composed of an input layer, an output layer, the hidden layers in between where much of the computational work is done. The model takes in data when training and adjusts the weights and parameters as it learns. We created 3 MLP models in total, each with varying parameters, specifically, learning rate, solver, and number of neurons. The learning rate manages the way the model responds to errors while it's training.The solver adjusts the weights between layers. MLP model 1 has an adaptive learning rate, a stochastic gradient descent solver, and 100 neurons. MLP Model 2 has a constant learning rate, uses a stochastic gradient-based optimization solver, and has 100 neurons. Lastly, MLP Model 3 uses an adaptive learning rate, a stochastic gradient-based optimization solver, and 89 neurons. When testing these models, they had RMSEs of 2.88, 2.87, and 2.87 respectively.

Before choosing a model, we wanted to see how the model would perform on new data. To do this, we performed K-Fold Cross validation with 10 folds on each model using recent 2021 data. The results of the cross validation can be seen on Table 1.1. The results of cross validation are not surprising as it reflects the results seen while training the models. Ultimately, the third Multi-Layer Perceptron (MLP) Regressor model proved to be the best model for predicting Spread as it performed the best during cross validation and it scored well with the more recent 2021 data that was not used in our training set. We believe that the learning rate and the decrease in neurons contributed to the lower score.

Model	K-Fold Cross Validation Score (RMSE)
Random Forest	4.825
Polynomial Regression	3.349
Stochastic Gradient Descent	3.081
Linear Regression	3.31
Multi-Layer Perceptor 2	2.897
Multi-Layer Perceptor 1	2.882
Multi-Layer Perceptor 3	2.848

Table 1.1

III. Methodology for Total

In order to predict the Total, the total number of field goals per game by both the home and away teams, we first sought out to determine which variables would form the best set of predictors. As Total is a numeric and count variable, it has the capability of being highly skewed. For instance, a shot can be worth 1,2, or 3 points, depending on the location the shot was made on the court, so a game with many 3-pointers would skew the total to be much larger than average. Another example would be if the teams are playing far greater defense than offense, the total number of points for that game would be skewed far lower than average.

To address this significant variance in predicting the total number of field goals, we created a Poisson regression model and compared the predictive value of each variable in the games dataset. Poisson regression is a very useful way to measure count data, such as ours with discrete values by providing us with the proper statistics to determine which explanatory

variables have an effect on our response variable, which is our count rate of total points in a game. In addition, by using the Poisson distribution, we avoid the error that could occur by using the normal distribution with a linear regression. Because the Poisson distribution accounts for the probability of events occurring over a certain amount of time, the results must be greater than or equal to zero (cannot have a game where both teams score 200 points a negative number of times), whereas the normal distribution and a linear regression model would work best for a continuous variable or set of variables, like we had for Spread.

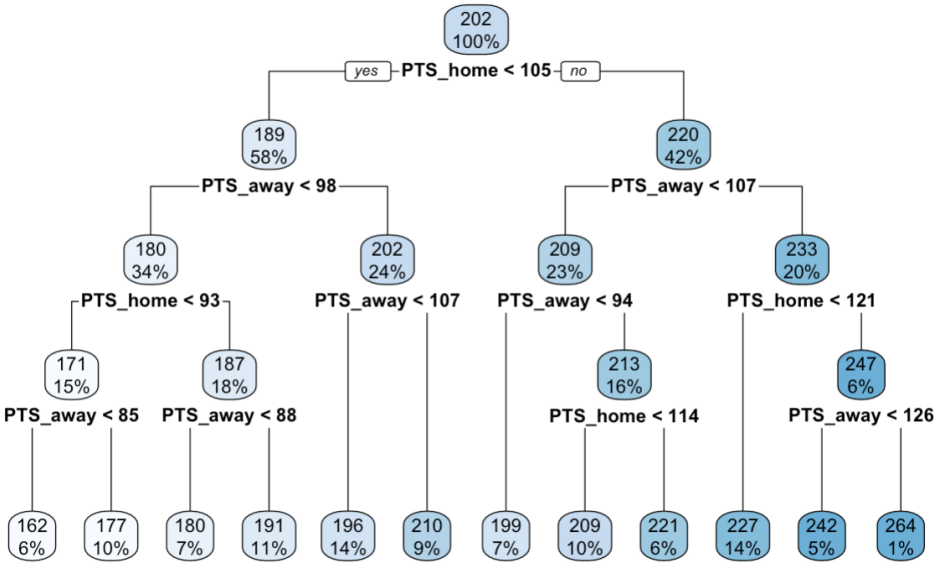
To begin, we used the dataset "data1" to model our Poisson regression. We used the glm function in R, as it accounts for response variables that do not follow the normal distribution, and specified it as a Poisson distribution by initializing the parameter, family, to be Poisson. All variables were added to the model, except those that clearly wouldn't have provided any useful information, such game status text, game ID, game date, home team ID, or visitor team ID.

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.1278140  0.0234558  133.35   <2e-16 ***
FG_PCT_home  1.2632498  0.0212122   59.55   <2e-16 ***
FT_PCT_home  0.1831317  0.0084285   21.73   <2e-16 ***
AST_home     0.0018963  0.0002017    9.40   <2e-16 ***
REB_home     0.0074305  0.0001538   48.30   <2e-16 ***
FG_PCT_away  1.2589801  0.0218469   57.63   <2e-16 ***
FT_PCT_away  0.1862646  0.0081268   22.92   <2e-16 ***
FG3_PCT_away 0.0914459  0.0084902   10.77   <2e-16 ***
AST_away     0.0018523  0.0002011    9.21   <2e-16 ***
REB_away     0.0075304  0.0001564   48.16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From this Poisson regression, we concluded that all of the other variables were significant predictors of the total - points away, points home, home assists, away assists, home rebounds, away rebounds, free throw percentage home, free throw percentage away, 3-pointers percentage away, and 3-pointers percentage home. Each predictor obtained a p-value of less than 2e-16, meaning they were very significant. In order to double check these findings, we performed a stepwise algorithm on our Poisson model, using the "step" function in R. This further proved the significance of our model by also concluding that each variable was a statistically significant predictor. From this stepwise analysis, the median residual was -0.0397, which is very close to

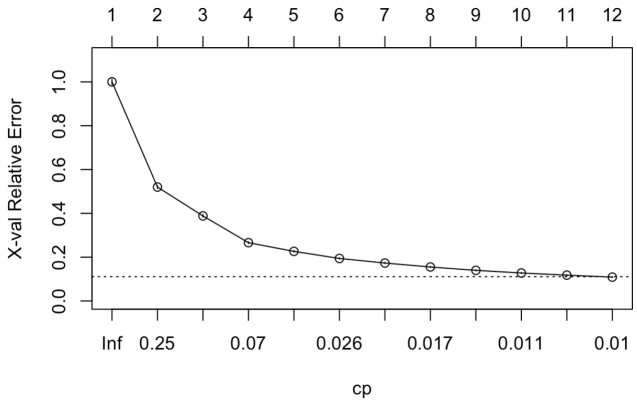
zero, signalling that our model was pretty good at predicting the total points the majority of the time. However, the minimum and maximum values for the residuals were -3.7768 and 0.1699 respectively, indicating that in certain scenarios, our model is not as great at predicting total, since it tends to skew to the left.

To see if we could form a more accurate model, we created regression trees in R. We used the classification and regression tree method. From our updated dataset that includes total, we split the data into a training set and test set, which contained 70% and 30% of the updated data respectively and used these models, “train” and “test” to perform the rest of our analysis for total. We used the packages Rsample for data splitting, dplyr for analyzing and manipulating the data, rpart and rpart.plot to build the regression trees, and ipred and caret for bagging. Below is the regression tree that we created.



As we can see from the model, the number of home points is the most significant predictor of the total number of points per game. This can also be concluded by the fact that a home court advantage truly does exist in sports, and it accounted for in several statistics. The second most significant predictor is the number of points by the away team, which also makes sense, as these two variables together make up the total. An advantage of this model and regression trees in general, are that they are very easy to interpret. The most significant variables are at the top, partitioned by a certain number of points, in our case. In addition, it provides a nonlinear response, which enables us to analyze the results, since we are not dealing with a

smooth, linear regression. On the other hand, since we have one single tree, there is a lot of variance, which can cause our model to work very well on our training set, but poorly on our test set. The plot below, created with the rpart function in R, displays the cross validated error of 0.1087, which is good. However, we want to see if we can get closer to an error of 0.5.

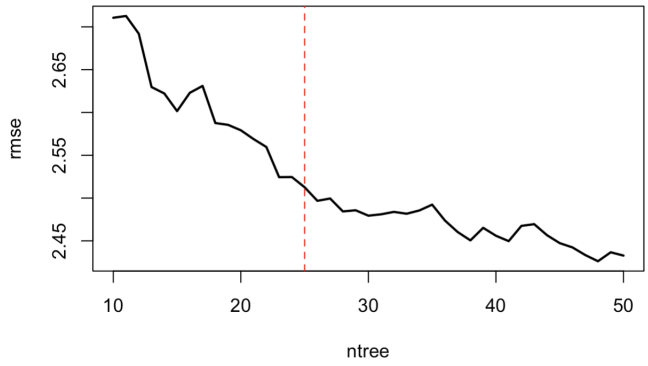


In order to see if we can improve our model, we used tuning algorithms, such as minsplit and maxdepth in R. Minsplit is the minimum number of data points necessary for a split, before the tree must create a terminal node. Our default was 20, so we decreased this by 10 to have terminal nodes with a much smaller number of observations for the predicted total. Maxdepth is the number of nodes in between the start and terminal nodes. The default is 30, so we also decreased this to 12, so that our trees would be smaller, and therefore, reduce the variance. From tuning, our cross validated error decreases to 0.1063, which shows that tuning made a significant difference in our models' predictive ability.

In our final attempt of creating the best model, we used bagging algorithms to utilize the variability of our tree to actually improve its performance by making several trees. In this process, we created, combined, and averaged 25 trees, which in turn reduces overfitting, and therefore, improves performance. The root mean squared error for the bagged trees was 7.6356. Although this value is not great, the range of the total number of points in a game varies greatly, so it makes sense that the standard deviation would be rather high.

To predict the total scores for future games, we made two new datasets containing the game data from the last 3 games and the last 5 games respectively, and used this new data on our previous models. The Poisson regression model did not predict the total scores well, as only

Points Away and Points Home were the only significant predictors now, with the new data. For the bagged tree, we got a new RMSE of 2.5126, which is a vast improvement from the RMSE we got from our original model. This shows how well the bagged trees really worked in predicting future values for total points in a game. Below the figure shows the RMSE for our bagged tree model.



IV. Methodology for OREB

Our first attempt at a model for OREB was to perform backward, forwards, and stepwise regression on the dataset to find which variables best predicted the total number of offensive rebounds. However, this method simply chose home offensive rebounds plus away offensive rebounds, because this sum is the exact number of total OREB. In order to create a more interesting model that includes other variables than home and away OREB counts, we decided to investigate different model building methods.

In order to still use backward, forwards, and stepwise regression, we split the dataset three ways: first, we combined the Home and Away predictors into a single dataset, then we split the dataset further into Home predictors and Away predictors separately. The datasets both contained some differential variables as well. This allowed us to build three linear models for the number of OREB by the home team, away team, and cumulatively. We first created the models separately, then summed the predictions from each to find a total.

For the Home predictors' model, we created an empty model with no predictors and performed forward regression. We also used the full model to perform stepwise regression. All three methods of model selection came back with the similar initial results, the stepwise and

forward regression resulted in the same model, but backward regression included two extra predictors that were not significant at the 5% level. These two predictors were OREB_CHANCES_ home and OREB_CHANCE%_home. The increase in the adjusted R^2 value seen when including these two predictors was very minimal, so chose to move forward with the forward and stepwise model selection results, which included 16 predictors, all with p-values very close to 0. The summary of this model is shown below.

```
Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                539.164506   8.421487  64.022  < 2e-16 ***
REB_home                     0.376530   0.007706  48.862  < 2e-16 ***
DREB_diff                   -0.156202   0.007366 -21.204  < 2e-16 ***
SEASON                      -0.275555   0.004211 -65.441  < 2e-16 ***
FGA_home                     0.106769   0.004828  22.116  < 2e-16 ***
AST_home                    -0.085197   0.004982 -17.100  < 2e-16 ***
BLK_diff                    -0.061633   0.005633 -10.940  < 2e-16 ***
OREB_CHANCES_home            0.151061   0.012096  12.489  < 2e-16 ***
STL_home                     0.100983   0.008662  11.658  < 2e-16 ***
TO_diff                      0.036376   0.006323   5.753 8.90e-09 ***
PTS_home                     0.026518   0.003065   8.653  < 2e-16 ***
Deferred_OREB_CHANCES_home  -0.916430   0.138852  -6.600 4.21e-11 ***
STL_diff                    -0.037644   0.008274  -4.550 5.40e-06 ***
`Contested_OREB%_home`      -0.021619   0.006332  -3.414 0.000641 ***
FT_PCT_home                 -0.924333   0.215194  -4.295 1.75e-05 ***
FG3_PCT_home                -0.569582   0.194160  -2.934 0.003355 **
FG_PCT_home                 -2.334763   0.865776  -2.697 0.007008 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.705 on 20948 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.5313,    Adjusted R-squared:  0.5309
F-statistic:  1484 on 16 and 20948 DF,  p-value: < 2.2e-16
```

We built the model for offensive rebounds by the away team in a very similar way, using the other half of the dataset containing variables for the away team to create a full model and empty model in order to perform backward, forwards, and stepwise regression. This time, each method came back with slightly different results, but with similar adjusted R^2 values. We chose to move forward with the stepwise regression model because it had the highest R^2 value and the fewest insignificant predictors. However, this model still had one insignificant predictor, BLK_away. When we removed this variable, the adjusted R^2 did not change at all, so excluding it did not lower our predictive ability. This left us with a model containing 18 significant predictors at the 95% confidence level. The summary of the model for offensive rebounds by the away team is shown below.

```
Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                  490.889152   8.893250  55.198  < 2e-16 ***
REB_away                       0.374778   0.005840  64.177  < 2e-16 ***
DREB_diff                      0.152765   0.005564  27.455  < 2e-16 ***
SEASON                        -0.255088   0.004210 -60.585  < 2e-16 ***
FGA_away                       0.111803   0.004874  22.940  < 2e-16 ***
BLK_diff                       0.083857   0.005598  14.979  < 2e-16 ***
AST_away                      -0.083552   0.004814 -17.356  < 2e-16 ***
TO_diff                       -0.040129   0.006306  -6.364 2.01e-10 ***
STL_away                       0.103060   0.008648  11.917  < 2e-16 ***
STL_diff                       0.060127   0.008227   7.309 2.80e-13 ***
PTS_away                       0.020288   0.002411   8.416  < 2e-16 ***
Deferred_OREB_CHANCES_away    -2.570926   0.534540  -4.810 1.52e-06 ***
`Contested_OREB%_away`         0.145630   0.047170   3.087  0.00202 **
FT_PCT_away                   -0.478981   0.189342  -2.530  0.01142 *
FG3_PCT_away                  -0.435862   0.195116  -2.234  0.02550 *
`Adjusted_OREB_CHANCE%_away`   0.815976   0.276317   2.953  0.00315 **
Contested_OREB_away           -1.998882   0.468946  -4.263 2.03e-05 ***
OREB_away2021                  1.730232   0.282524   6.124 9.28e-10 ***
`OREB_CHANCE%_away`           -0.958098   0.293737  -3.262  0.00111 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.696 on 20946 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.5148,    Adjusted R-squared:  0.5144
F-statistic:  1235 on 18 and 20946 DF,  p-value: < 2.2e-16
```

We then conducted cross-validation by splitting the datasets into training and testing data. 70% of the data was designated to the training data and 30% was withheld in the testing dataset. We then calculated the Root Mean Square Error (RMSE) for each model, resulting in a value of 2.68 for the home model and 2.95 for the away model. We also calculated the Mean Absolute Error (MAE) for each and got a result of 2.12 for the home model and 2.35 for the away model. However, the more important cross-validation is for total offensive rebounds. To do this, we split the entire dataset into training and testing data, again with 70% designated for training and the other 30% for testing. We then ran both models on the training dataset then found predictions for the testing data. We then calculated the RMSE and got a value of 4.70. This is a relatively high RMSE, so we decided to investigate the variability in offensive rebounds in games to see if this was unreasonable. The summary statistics for OREB is shown below.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   18.00   21.00   21.69   25.00   84.00
```

In our dataset, we had a large amount of variability, with OREB counts as low as 5 and some as high as 84, which would explain the relatively high RMSE. These extremely high and

extremely low numbers seemed like rare events, so we decided to investigate further. We first decided to count the number of OREB observations above 50. There were only 2 of these observations, so we decided these could be removed from our dataset for this model to avoid the high amount of influence they could have. We then looked at the number of observations of OREB above 40. There were 43 of these observations, again, a very low number considering the large number of total observations. To handle extremely low observations, we checked how many were below 10 and found that to be 143. This is less than 1% of observations, so we decided these could also be removed for our dataset for this model. In the interest in balancing the amount of data we removed from both extreme ends, we ended up removing OREB observations above 36, which contained a more similar number to what we removed from the lower end, again with fewer than 1% of the observations. Running our models again without these extreme outliers, we were able to get an RMSE of 4.45, meaning our model will typically predict within +/- 4.45 of the actual number of offensive rebounds. This RMSE is still relatively high, so we decided to try a cumulative model, instead of predicting home and away offensive rebounds separately.

To predict the total number of offensive rebounds, we used linear regression on the Home and Away cumulative model. First, we cleaned the given dataset and created separate rows for Home teams and Away teams. This gave us 48,122 observations in total to perform backward regression on. We then created an empty model with no predictors for forward regression and elimination of outlying variables. Combined with further stepwise regression, we were able to identify one predictor which was not significant at the 5% confidence level and eliminated it moving forward, which is the DREB_diff. The remaining eight predictors are shown to be statistically significant. The summary of the stepwise regression model for the cumulative data is shown below.

```
Call:
lm(formula = OREB ~ REB + DREB_diff + AST + FGA + STL + PTS +
    BLK + FT_PCT + FG_PCT, data = totaldata)

Residuals:
    Min      1Q  Median      3Q     Max
-11.361  -2.042  -0.084   1.958  31.876

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.758397   0.278454  -6.315 2.73e-10 ***
REB          0.253982   0.002463 103.127  < 2e-16 ***
DREB_diff   -0.001748   0.001957  -0.893    0.372
AST         -0.094060   0.003523 -26.700  < 2e-16 ***
FGA          0.129951   0.002635  49.319  < 2e-16 ***
STL          0.065390   0.004807  13.603  < 2e-16 ***
PTS         -0.010925   0.002095  -5.215 1.85e-07 ***
BLK         -0.073713   0.005480 -13.452  < 2e-16 ***
FT_PCT      -2.218667   0.142455 -15.575  < 2e-16 ***
FG_PCT      -9.344391   0.471049 -19.837  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.997 on 48112 degrees of freedom
Multiple R-squared:  0.4154,    Adjusted R-squared:  0.4153
F-statistic:  3799 on 9 and 48112 DF,  p-value: < 2.2e-16
```

We then conducted Cross-Validation by splitting the datasets into training and testing data. 70% of the data was designated to the training data and 30% was withheld in the testing dataset. We then calculated the Root Mean Square Error (RMSE) for the Home and Away cumulative model, resulting in a value of 2.98. We also calculated the Mean Absolute Error (MAE) of 2.38. This cumulative model has a RMSE lower than that of previous attempts to sum separate home and away models. This model will likely predict within +/- 2.7 offensive rebounds, as opposed to the previous +/- 4.45. Because we successfully lowered the RMSE, this is the model we will move forward with to make predictions on upcoming games. To predict the OREB for future games, we made a new dataset containing the average game data statistics from the most recent games for each matchup and used this new data on our previous model. The highest OREB quotient predicted is 18.94 whereas the lowest OREB quotient predicted is 4.80.

Throughout the model selection process, multiple nonlinear transformations were also attempted to see if we could increase our R^2 or decrease of root mean square error. However, most of these nonlinear transformations, such as squared and cubic terms, ended up resulting in worse predictions or very slight improvements. Therefore, in the interest of simplicity and comprehensibility, we decided these transformations should not be included in the final models.