# Model 2 Selection Report

Elias Cheleuitte, Micaleb Johnson, Morgan Williams

## Model Selection:

After considering Quasi-Poisson, Zero-Inflated Poisson, and Zero-Inflated Negative Binomial models, after grouping our NBA data, for our second model attempt, we decided to instead use the non-parametric Random Forest regression model. This was due to a number of factors, including some of our predictors being relatively highly correlated, the presence of a few outliers that we determined should be kept in the data set, and the complexity of any potential parametric models.

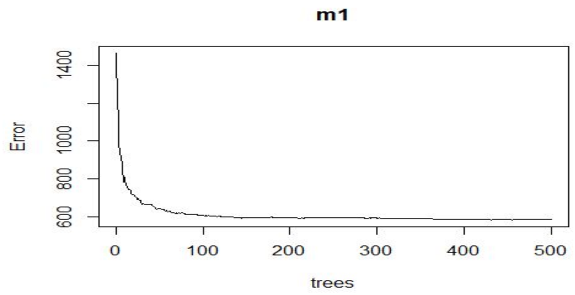
Simply put, random forests create a specified number of decision trees, and then, when given a set of predictors, input those predictors into each tree and either use a majority vote of the trees to classify a categorical response or average the outputs of a continuous response. Random forest algorithms use the bootstrap resampling from the sample data for each decision tree it creates, along with picking which predictor to use to make a split from a random sample of all of the predictors. In order to utilize the observations in the training data that are not sampled in the bootstrap resampling, it is common practice to use this data, known as the out-of-bag (OOB) sample, to approximate the testing error.  So, each decision tree in the random forest  is evaluated using the OOB sample to find it's individual OOB error, this is then averaged for a single approximation of the testing error.

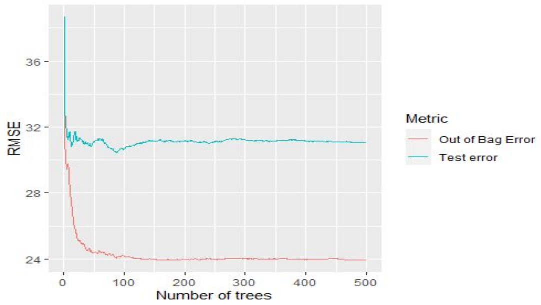
We proceeded to split our modified NBA dataset (as described in the previous short report) with the 15 covariates and the response variable into training and testing sets with 80% of the original data going into the training set and 20% into the testing set. First, a default random forest model to predict win shares was fitted on the training dataset. It created 500 different forests each with a different number of trees within them, ranging from 1 to 500, each picking from a random sample of 5 of the covariates at each tree split, and resampling with replacement from all of the sample data. This resulted in a model with an out-of-bag mean squared error (MSE) of 703.2708 and explained 18.65% of the variance. You can see from Figure 1 that the error rate decreases as the average number of trees increases until around 100 trees where the error rate stabilizes, but continues to slowly trend lower. Then to show the similarity between the OOB and test MSE, we refitted the model while evaluating both the OOB and test MSE which can be seen in Figure 2.

While random forests can provide accurate predictions with little to no tuning, using a standard set of parameters, in order to improve our random forest model to obtain optimal performance and accurate predictions of NBA win shares we proceeded to evaluate different values of our tuning parameters. The tuning parameters of primary focus were the sample size of the predictors which would be considered at each tree split, minimum node size (the lowest number of observations that could be at the end of any one branch of a decision tree, and the size of the bootstrapping sample on which each decision tree would be fitted.

Figure 1:

Figure 2:





## Model Tuning:

We began with an initial tuning process of our model to find which mtry parameter (the number of covariates available for splitting at each tree node) would be optimal. This process was started with the default mtry of the total number of covariates divided by 3, in this case 5, and increased by a step factor of 1.5 until our OOB errors stop improving by 1%. Using this tuning process, figure 3 below shows that the predicted best mtry is 4, as it resulted in the lowest OOB error.

Figure 3:

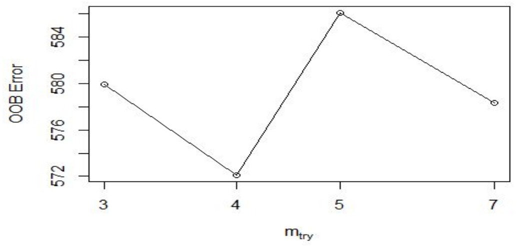
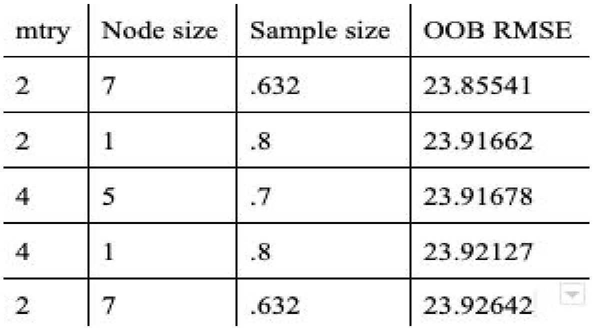


Figure 4:



| mtry | Node size | Sample size | OOB RMSE |
|---|---|---|---|
| 2 | 7 | .632 | 23.85541 |
| 2 | 1 | .8 | 23.91662 |
| 4 | 5 | .7 | 23.91678 |
| 4 | 1 | .8 | 23.92127 |
| 2 | 7 | .632 | 23.92642 |

The next step was to perform a large grid search to predict the best hyperparameters for our model. Specifically, we created a grid that would evaluate 96 different models using an mtry that ranged from 2 to 12 by a sequence of 2, minimum node size with a sequence of 1 to 7 by 2, and a sample size of 0.55, 0.623, 0.70, or 0.80. The grid search generated candidates for each hyperparameter from the sequence constraints and generated combinations of the hyperparameter values. With the generated combinations of parameter values we ran a random forest model and used the OOB RSME to evaluate each model. The 5 best models and their hyperparameter values were recorded above in figure 4. From this, our highest performing model can be seen to have an OOB RSME of 23.85541 along with the following hyperparameter values, an mtry of 2, node size of 7, and a sample size of .632. Contrary to figure 3, the top models had mtry values of 2, not 4.

## Model Performance:

The resulting optimal model with our training data produced an R-Squared value of 0.1227324. When the model was run using our test data it yielded a higher R-Squared value of 0.2407223. While both are not great R-squared values our data still follows the win share trend fairly accurately outside of the top 5 picks. In the top 5 picks, the pattern that the data previously had from the previous 25 picks abruptly increases in variability. This is due to the superstar players, who are almost always selected with a top draft pick, and have huge impacts on the game and thus have win share totals that are far greater than any other player. Our model fails to take into account the upwards variability of win share values within the top 5 picks created by “superstar” players. Thus, it underestimates the win shares for the highest values of total win shares as seen in both figures 6 and 7. In figure 6, you can see that the win shares for the top 5 picks are much more variable than the other 25 picks and the model fails to take this into account. This is likely due to the fact that random forest models are robust to outliers. In both plots, when the actual value of win shares is below 50, the model does a very good job predicting, however, for the superstar players whose win shares are above 50, the model thinks that these players follow the same pattern as the other players so it drastically undercalculates the win shares for these players.

Figure 6:

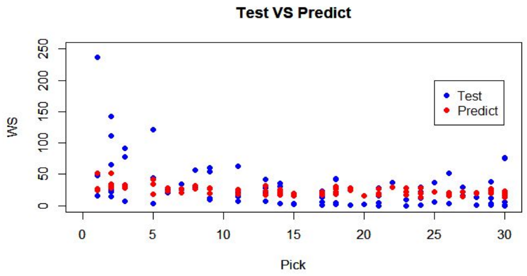
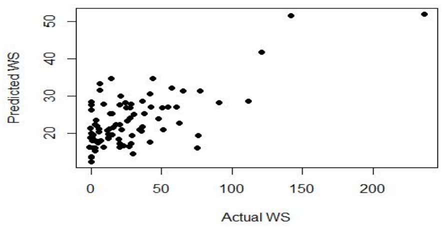
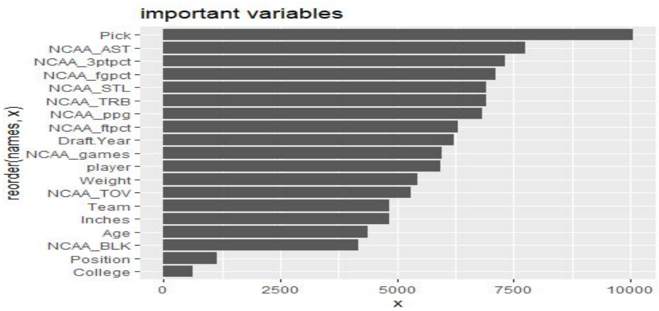


Figure 7:



The model also provides the importance of each variable, which is measured by looking at how much the MSE decreases when a variable is used at a node to split a decision tree. You can see from figure 8 that our model pick and college type were deemed the most and least important variables respectively. The importance of each variable is similar to our correlation from the variables in our linear regression model.

Figure 8:



## References:

https://uc-r.github.io/random_forests (random forest guide), randomForest (default model), ranger (optimal models), tidyverse (Plotting and data manipulation), rsample (resampling), caret (modelling)