Using Statistical Modeling to Predict the Success of College Basketball Transfers

Miles Kee

April 24, 2024

Abstract

The NCAA transfer portal allows college athletes to transfer schools during their college career. Over the last few years, the number of athletes entering the transfer portal has skyrocketed, leading to teams using it as a central resource in building their roster. This paper aims to use statistical learning techniques to efficiently analyze the thousands of players who have entered the transfer portal since 2020. This study compares various statistical methods of classifying whether men's college basketball players will improve or not and predicting their numerical efficiencies. These methods include K-nearest neighbors, logistic and linear regression, neural networks, linear discriminant analysis, random forest, and support vector machines. Similar work has been done to predict NBA player performance, but not for college basketball players. Coaches and teams can use modeling techniques like the ones in this paper to target players to acquire in the transfer portal based off their offensive and defensive statistics from the previous year and avoid recruiting players whose characteristics may not benefit their team.

1 Introduction

In 2021, the NCAA changed the rules regulating student-athletes transferring schools. For years, players who wished to transfer had to sit out a season to become eligible at their new school. While college basketball players would still transfer, impact players transferring schools was a rarity. Under the old rules, players would mainly transfer if their old school was not a great fit, they were unhappy with their playing time, or the coach they had committed to play for had left the school or been fired. Teams rarely had to deal with losing their best players to other schools, or having new players come in that were supposed to be their best player. But, when the NCAA decided to eliminate the sitting-out aspect of transferring schools, the landscape of college basketball, and college sports as a whole, changed completely. The rule change also came at a similar time as college athletes began to be permitted to profit off their name, image, and likeness (NIL). These two key changes to college athletics have created a pseudo-free agency during the off season. Players can choose to capitalize on any success they have had and transfer to a school that will offer them the biggest possible NIL package, and they can do this for as many seasons as they would like. Since these new rules came into play, we have seen players who have gone to four different schools during their four years of eligibility. The new transfer portal has led to an overwhelming amount of players changing schools, a number that projects to be over 1,000 in 2024 off season. This paper uses statistical modeling techniques to predict men's college basketball players' future success in the transfer portal by using their offensive and defensive statistics.

Oliver (2004) was one of the first pieces of literature that outlined a rating system for basketball players and teams. It modernized the way people in basketball use statistics by proposing evaluators normalized to possession counts. Looking at statistics on a rate basis, or per a number of possessions (usually 100) allows us to better understand a team's or player's strengths and weaknesses because some teams have more possessions in their games, allowing the team and its players to accumulate counting statistics that do not tell the full

story of their success or failure, and vice versa with teams that have fewer possessions in their games. Nguyen et al. (2022) uses linear regression, classification, and machine learning to predict NBA player performance, using many of the statistics Oliver (2004) outlined to train their models. However, not much work has been done in the field of predictive college basketball player ratings. Most of the work in predictive analytics for college basketball involves predicting game outcomes or team ratings, not individual players. Zimmermann et al. (2013) uses machine learning techniques to predict the outcomes of college games over the course of a season, including regular and post season games. West (2006) creates a team rating system for predicting success in the college basketball postseason and Ruiz and Perez-Cruz (2015) uses a generative model to predict the outcomes of NCAA tournament games. Sites like KenPom and barttorvik.com have college player rating systems, but do not do much projection of player success. This project is unique in terms of looking at player projection, but also due to the novelty of the transfer portal. Very little work has been done on players who enter the transfer portal. Davis (2023) predicts which players will enter the transfer portal, but does not discuss their chances of success afterwards. This study will build on the past works by projecting the chances of success of players who enter the transfer portal, as well as predicting their numerical output for the next season.

2 Data

The data for this project comes courtesy of barttorvik.com. The variables downloaded from the website include offensive statistics, defensive statistics, and general player information like height and year in college. From these downloaded variables, four additional variables, including "fromp5," "top5," "oimp," and "dimp" are created to indicate conference status and player improvement. See Table 1 for a detailed description about all variables used in this study. Effective field goal percentage (eFG) is chosen instead of traditional field goal percentage because it properly weighs the added impact of a three pointer when compared

Table 1: Variable explanations, divided by general factors, offensive factors, and defensive factors. A check indicates that the variable is used in the type of model it is checked for.

Variable	Explanation	Used in Offensive Model	Used in Defensive Model
General Factors			
player_name	The name of the player		
team	The team the player played for in the season before he transferred		
conf	The conference the player's original team plays in		
min_per	The percentage of a team's minutes a player played in	✓	✓
yr	Year in college of the player		
ht	Height of the player		
new_school	School the player transferred to		
fromp5	Indicates if the school the player transferred from is in a power conference (1) or not (0)	✓	✓
top5	Indicates if the school the player transferred to is in a power conference (1) or not (0)	✓	✓
Offensive Factors			
ORtg	The number of points scored by the player's team per 100 possessions when he was on the court	✓	
usg	The percentage of a team's plays used by the player while on the floor	✓	
eFG	A metric to measure a player's shooting success	✓	
orb_per	The percentage of available offensive rebounds a player got	✓	
ast_per	The percentage of made shots a player assisted while on the court	✓	
to_per	The percentage of turnovers a player committed while on the court	✓	
ftr	The number of free throws a player attempts per field goal attempt	✓	
ORTG	The player's offensive rating for the year after they transferred	✓	
oimp	Indicates whether the player's offensive rating improved the year after they transferred (1) or not (0)	✓	
Defensive Factors			
drb_per	The percentage of available defensive rebounds a player got		✓
blk_per	The percentage of opponent shot attempts blocked by a player while on the court		✓
stl_per	The percentage of opponent possessions ending in a steal by the player		✓
dpbm	The player's defensive contribution in terms of points above league average per 100 possessions		✓
DBPM	The player's defensive box plus minus for the year after they transferred		✓
dimp	Indicates whether the player's defensive box plus minus improved the year after they transferred (1) or not (0)		✓

to a two pointer by multiplying three pointers made by 1.5. The formula is eFG $= (2PM + 1.5 * 3PM)/(FGA)$, where $2PM$ and $3PM$ are two and three pointers made, respectively, and FGA is field goals attempted.

There are many more reliable statistics available to track a player's offensive impact when compared to the statistics available to track a player's defensive impact. Defense is much more nuanced, as not every good defensive play shows up in a box score. This is why dbpm is used, as it reflects a team's full defensive performance when a player is on the court, allowing us to better quantify an inherently non-quantifiable concept.

The data contains statistics for the years before and after a player transferred, and includes all transfers from 2021-2023. The years 2021, 2022, and 2023 are looked at because these are the years the "free agency" aspect of the transfer portal started. The 2024 transfers are excluded from this study because at the time, the new season had not given a significant enough sample size to evaluate the players' performances. This leaves us with 2093 players in the study. The target variables to predict are ORTG and DBPM (or whether or not those metrics improved), as those do the best job of encapsulating a player's impact on offense and defense, respectively, into a single number. For every model made, 70% of the data (1465

Table 2: Offensive classification model performance metrics. The best method is bolded for each metric.

Model	Correct Improvement	Correct Non-improvement	Overall Correct
KNN	70.7%	58.5%	65.6%
Logistic	80.7%	60.0%	72.1%
Neural Network	73.9%	60.0%	68.2%
LDA	**82.6%**	58.1%	72.5%
Random Forest	79.1%	**61.1%**	71.7%
SVM	82.1%	60.1%	**72.8%**

Table 3: Defensive classification model performance metrics. The best method is bolded for each metric.

Model	Correct Improvement	Correct Non-improvement	Overall Correct
KNN	71.9%	68.8%	70.4%
Logistic	76.6%	71.1%	73.9%
Neural Network	75.0%	**72.7%**	73.9%
LDA	75.9%	68.2%	72.1%
Random Forest	74.4%	71.8%	73.1%
SVM	**77.2%**	71.1%	**74.2%**

observations) is used for the training set and 30% of the data (628 observations) is used for the test set. The observations are chosen at random and kept consistent throughout.

3 Classification Models

This section uses six classification modeling techniques: K-nearest neighbors (KNN), logistic regression, neural networks, linear discriminant analysis (LDA), random forest, and support vector machines (SVM). These models are fitted to a training set based on all of the variables available for offense and defense, respectively. The metrics for evaluating classification accuracy for each method are shown in Table 2 and Table 3 for offensive and defensive models, respectively.

We begin by examining the KNN classification method. Using cross validation and the `caret` package in R, $K = 23$ is chosen for the KNN to predict offensive improvement and

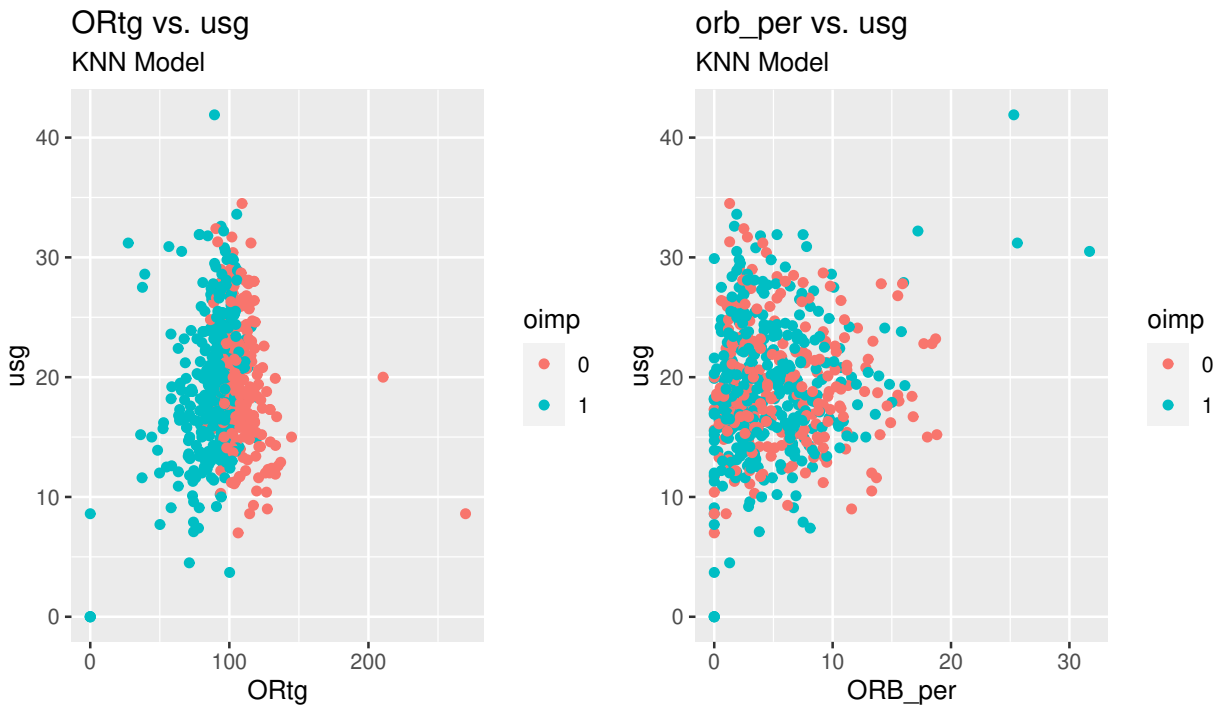

Figure 1: ORtg plotted against usg in offensive model (left panel) and orb_per plotted against usg in offensive model (right panel) for fitted offensive improvement KNN model.

$K = 25$ is chosen for the prediction of defensive improvement. While KNN does not have coefficients to interpret, we can infer variable importance based on plots of the variables along with their classifications. For example, in the left panel of Figure 1, we can see a clear decision boundary emerge as offensive rating changes when plotted against usg. Yet, in the right panel of Figure 1, the distinction between classifications when orb_per is plotted against usg is much less clear. From these two plots, we can infer that ORtg has a much bigger impact on whether a player will improve or not than orb_per in this model. By creating similar plots for each variable in the model, we can start to distinguish the importance of each variable. Offensively, ORtg, eFG, to_per, and fromp5 have the most distinction between classifications, while Min_per, AST_per, TO_per, and top5 have little distinction. A similar process for the defensive model can be seen in the left panel of Figure 2 and the right panel of Figure 2, where dbpm can be seen with a much cleaner boundary than drb_per. Using this process for the defensive model, dbpm and top5 seem to carry the most weight.

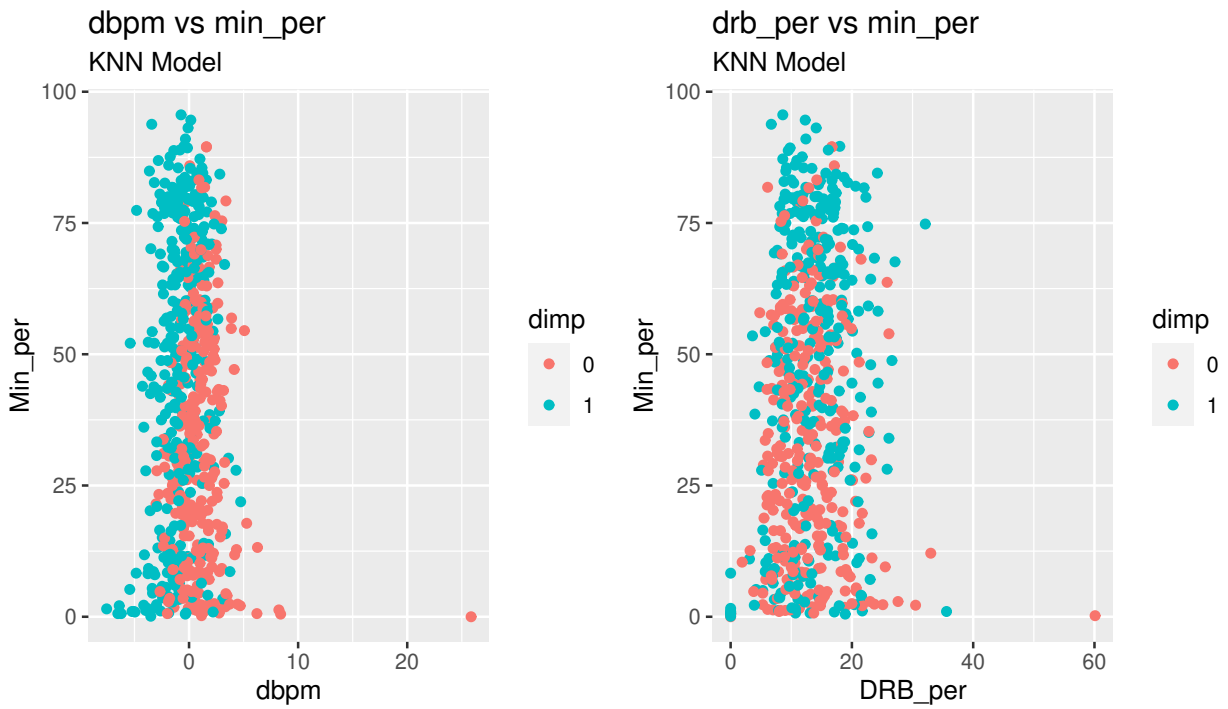

Figure 2: dbpm plotted against Min_per in defensive model (left panel) and drb_per plotted against Min_per in defensive model (right panel) for fitted defensive improvement KNN model.

The second method used is logistic regression. The coefficients of the logistic regression model to classify offensive and defensive improvement are shown in the left panel of Figure 3 and the right panel of Figure 3, respectively. On the x-axis of the graphs, each bar is labeled with its variable name. The y-axis shows the coefficient estimate for each variable. The color of each bar corresponds to whether it is statistically significant or not, and the number printed on each bar is the standard error estimate for each variable.

Next, a single-layer neural network is fitted to the data. An epoch length of 100 is chosen, as this is where the loss and accuracy of the model begin to plateau. A batch size of 32 is chosen, which is the standard number to start with when modeling a dataset of this size, and also showed the best results in Thakur (2023). RMSprop, which is an improved version of gradient descent, is used as the optimizer when compiling the model. More information about the RMSprop optimizer can be found from Hinton et al. (2012). The loss function used in this case is binary cross entropy, which is considered the best loss function to use for

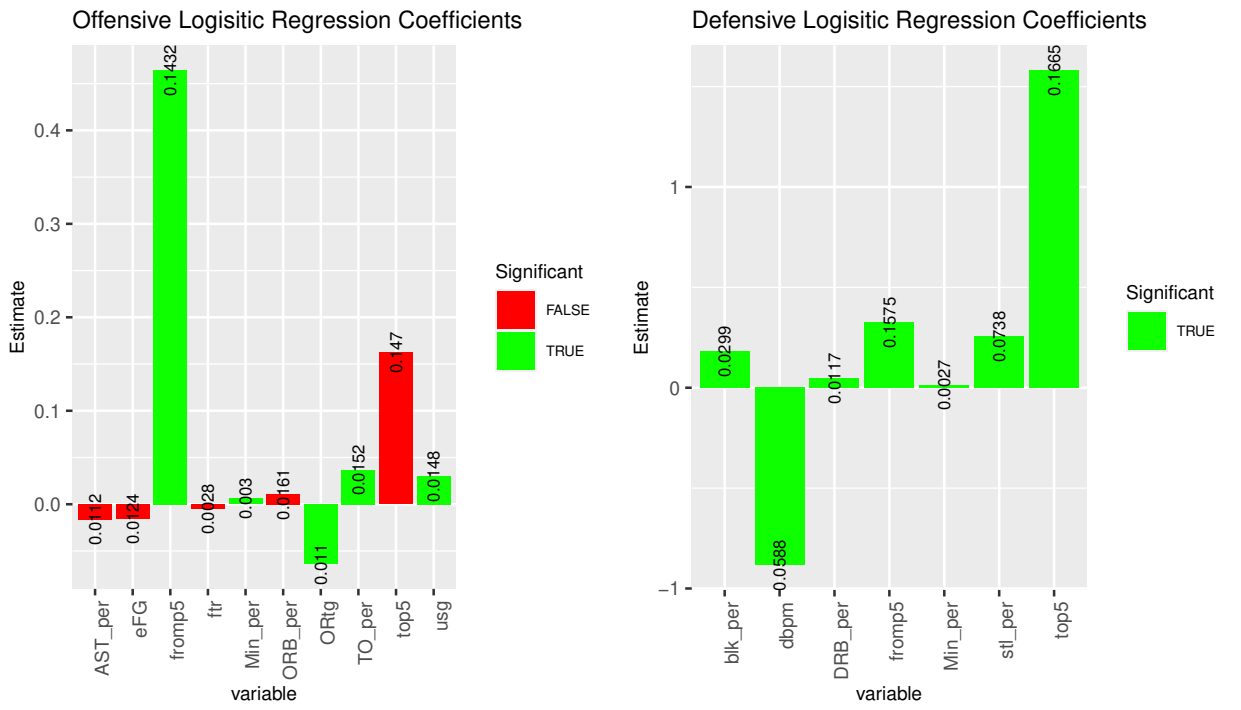

Figure 3: Coefficients, standard errors, and p-values of the offensive (left panel) and defensive (right panel) logistic regression models.

classification tasks (Ruby and Yendapalli, 2020). Following Hunter et al. (2012), we use the same number of neurons in the hidden layer as the number of inputs, resulting in 11 neurons and 7 neurons for the hidden layers in the offensive and defensive models, respectively.

As the fourth method, LDA is chosen to use over PCA because LDA is much better for separating classes and understanding the distribution of the data (**?**). Since LDA predictions are created from a linear combination of the coefficients found, their relative importance can be seen in a bar plot of their values. The left panel of Figure 4 and right panel of Figure 4 display the LD1 coefficients for the fitted offensive and defensive models, respectively. The model only has LD1 coefficients because there are only two classification outcomes possible, improvement or non-improvement. In the offensive model, fromp5, top5, ORtg, and usg seem carry the most importance. In the defensive model, top5, dbpm, and stl_per appear to be the most important. The fifth method used is random forest. For the number of variables to randomly sample as the candidates split, the generally used method for classification

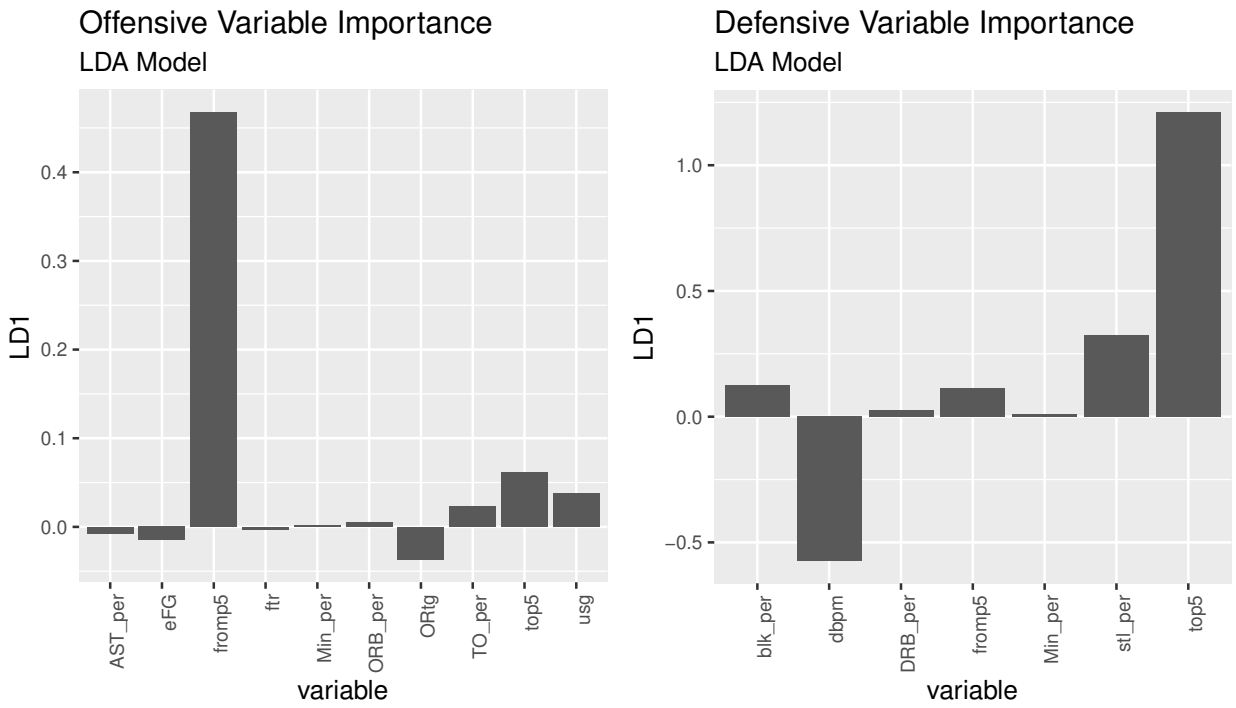

Figure 4: LD1 coefficients of the offensive (left panel) and defensive (right panel) LDA models for classifying improvement.

is to find $floor(\sqrt{X})$, where X is the number of features in the data (Andy Liaw and Matthew Wiener, 2002). So, with 10 and 7 features for the offensive and defensive models, respectively, the models will both use 2 as $mtry$. The number of trees for each model is chosen by examining the plot of out of bag error rate for and selecting the number of trees where the error begins to converge. For both models, 300 trees are used based on the plots. Variable importance can be seen in the left panel of Figure 5 and the right panel of Figure 5 for offense and defense, respectively, using the `plot_multi_way_importance` function from `randomForestExplainer` (Aleksandra Paluszyńska et al., 2020). The x-axis is a measure of how much the accuracy of predictions decreases when the variable is removed, and the y-axis is a measure of how well the variable can split the data. Variables closer to the top and right parts of the graph are more important, so therefore ORtg, eFG, Min_per, and TO_per are the most important variables for offense. Defensively, dbpm, Min_per, and top5 are the most important variables. For the sixth method used, SVM, a linear kernel is chosen.

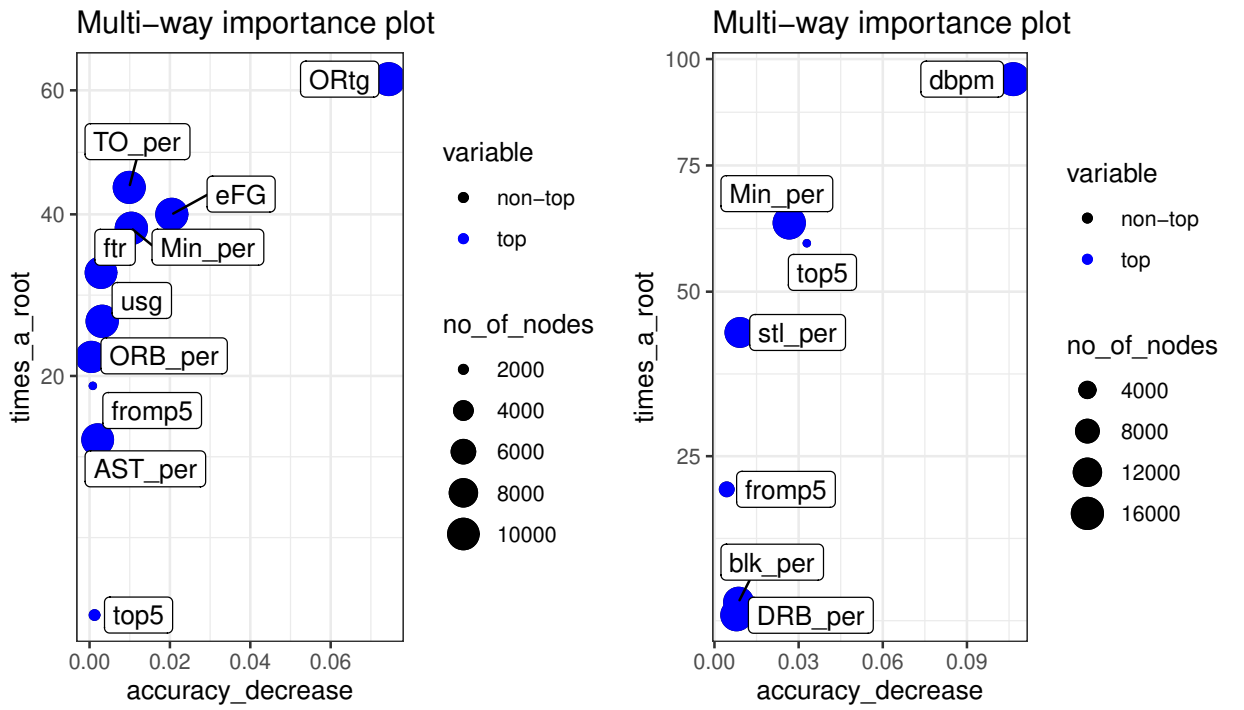

Figure 5: Measures of variable importance in the offensive (left panel) and defensive (right panel) random forest models.

This is chosen due to the idea that the data is linearly separable. A cost of 5 is chosen after using cross validation for each model. A similar technique used to look at KNN variable importance can be used for SVM. This technique can be seen offensively in Figure 6, and defensively in Figure 7. Offensively, the most important variables are ORtg, eFG, and to_per, while defensively dbpm seems to be the only variable with clear importance. ORtg is agreed upon by all the offensive models as one of the most important predictors of the following season's success. This is intuitive, as a player's offensive rating one year shows a lot about their overall skill level and can be used very well to predict their overall offensive output for the following season. Interestingly, ast_per is not found to be very useful by many models, but to_per is found to be useful. This is interesting because assists and turnovers often go hand in hand when referencing a player's ability to make plays for others, but it seems in this case that it is more important that a player does not make mistakes (turnovers), than making good plays (assists). fromp5 is important in a few of the models, but top5 is

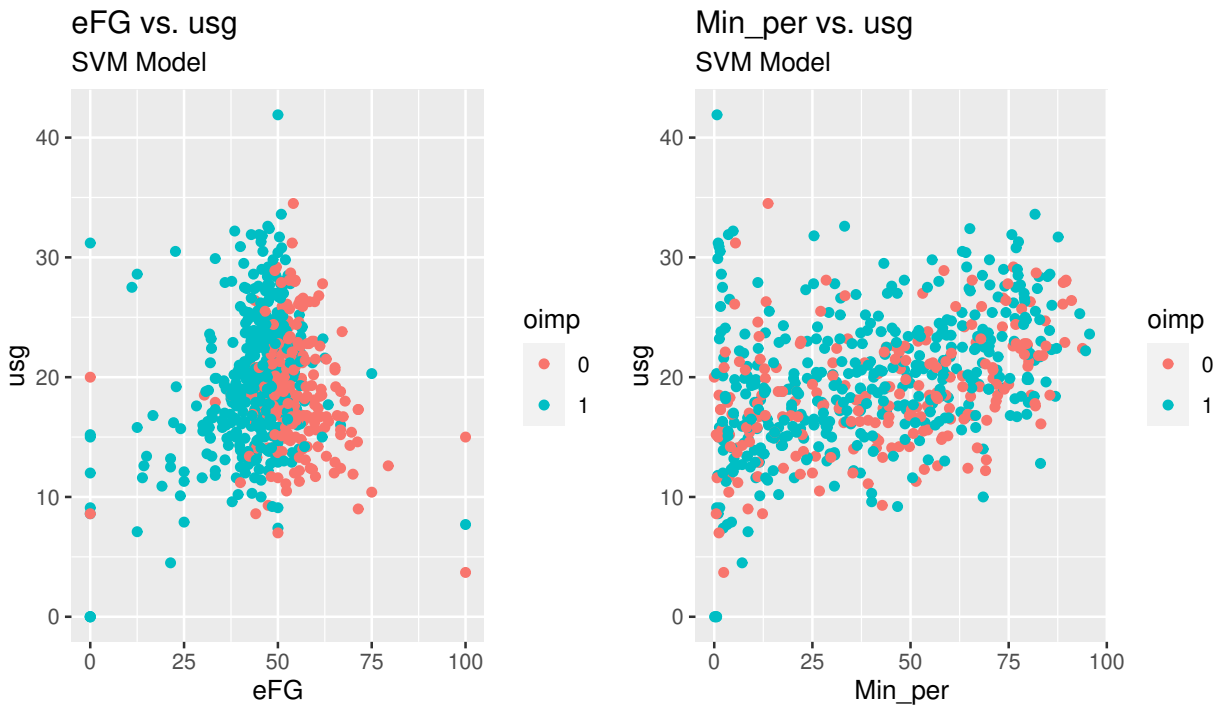

Figure 6: eFG plotted against usg in offensive model (left panel) and Min_per plotted against usg in offensive model (right panel) for fitted offensive improvement SVM model.

important in none of the models, meaning that the school a player previously went to is much more indicative of future success than the school they transferred to. Statistics that showed a player's volume of work, like Min_per and usg, do not seem to carry much predictive value, and ftr and orb_per do not either.

Similar to ORtg, dbpm is important across the board, for the same reason. Opposite of the offensive models, top5 has much more importance than fromp5, indicating that defensive success is more dependent on what school you are going to than what school you came from. Min_per has much more value in the defensive models than in the offensive ones, meaning that playing time is more predictive of defense than offense. drb_per is also not considered very important, which means that rebounding on both sides of the ball do not carry much predictive value. stl_per is important in more models than blk_per, indicating that steals are a more effective predictor of future defensive success than blocks.

Performance-wise, the LDA model is the best for classifying offensive improvements, and

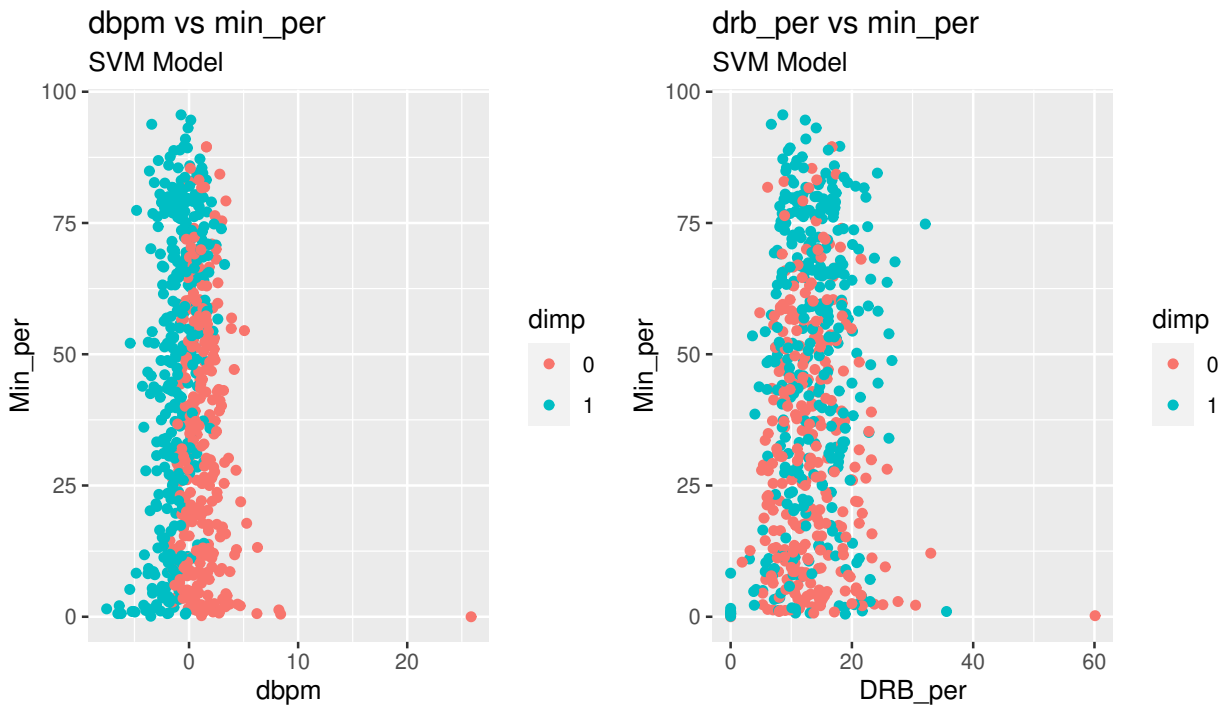

Figure 7: dbpm plotted against Min_per in defensive model (left panel) and drb_per plotted against Min_per in defensive model (right panel) for fitted defensive improvement SVM model.

the random forest model is best for predicting non-improvements. Overall, however, the SVM model has the best prediction rate. Defensively, the SVM model is best for predicting improvements, the neural network is best for non-improvements, and the SVM model has the best overall prediction rate. It seems that the SVM models overall performed the best, as they are at the top or close to it in all of the metrics shown in the results tables.

4 Numerical Prediction Models

This section will use numerical prediction modeling techniques like linear regression, linear discriminant analysis, neural networks, random forest, and support vector machines. The training and testing split used will be the same as the one used for the classification models, which is discussed in section 2. The metric we chose to evaluate the models is mean absolute

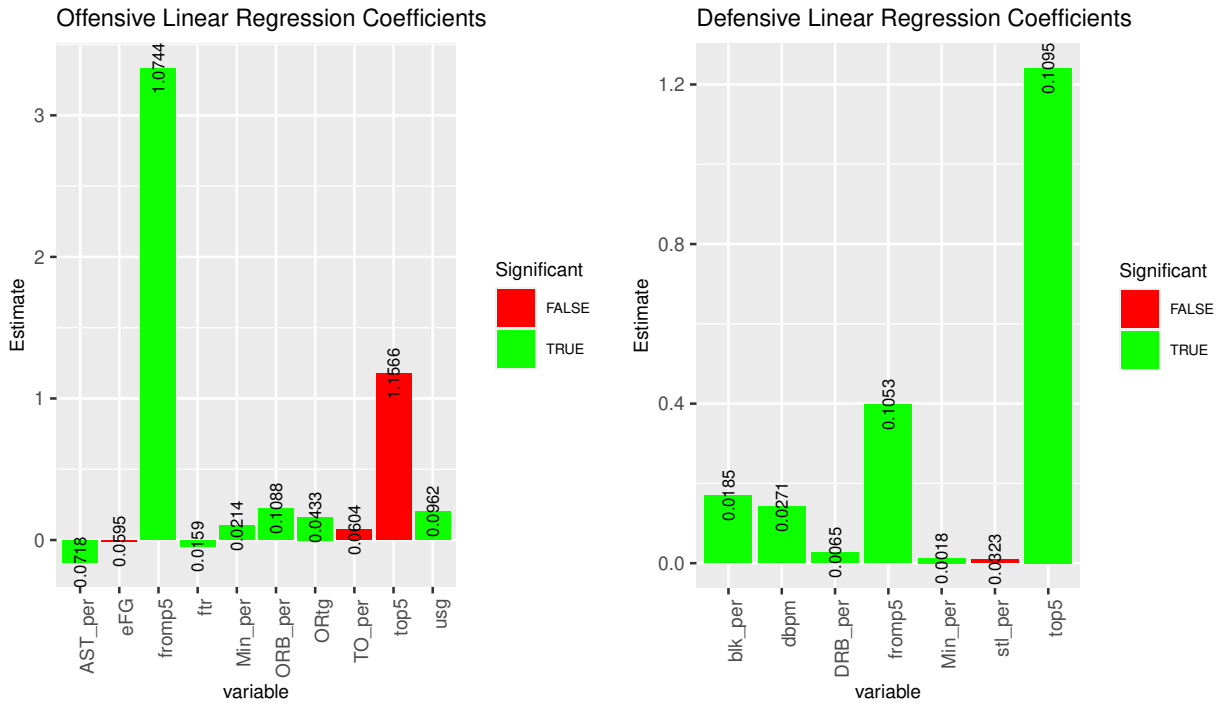

Figure 8: Coefficients, standard errors, and p-values of the offensive (left panel) and defensive (right panel) linear regression models.

error, given by

$$MAE = \frac{\sum_{i=1}^{n} |x_i - p_i|}{n}, \tag{1}$$

where x_i is the actual value, p_i is the predicted value, and n is the sample size. This metric is chosen because we believed using root mean squared error would give an unfair weight in the error term to observations that have very high errors due to the player simply not playing enough minutes to be able to draw a logical conclusion on. The results of each model are shown in Table 4.

The coefficients of the linear regression model to predict offensive and defensive efficiencies are shown in the left and right panels of Figure 8 , respectively. On the x-axis of the graphs, each bar is labeled with its variable name. The y-axis shows the coefficient estimate for each variable. The color of each bar corresponds to whether it is statistically significant or not, and the number printed on each bar is the standard error estimate for each variable.

The same tuning parameters are used in the neural networks created for numerical pre-

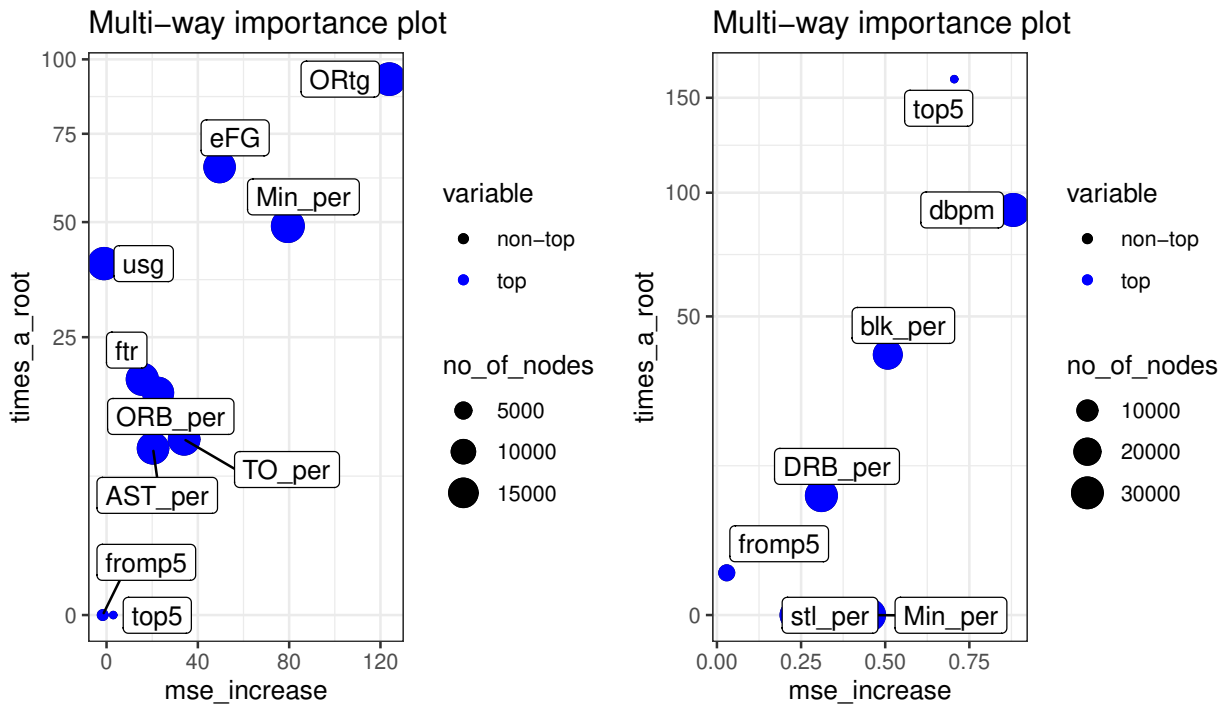

Figure 9: Measures of variable importance in the offensive (left panel) and defensive (right panel) random forest models.

diction as in section 3. This includes a batch size of 32, epoch length of 100, the RMSprop optimizer being used, and the number of hidden layers in each model (11 for offense, 7 for defense). The only difference in these models is that the ReLU activation function is used since we are no longer predicting probabilities, but instead predicting efficiencies themselves.

In the random forest model, again ntree is set to 300 after examining the graph of out of bag error. mtry is set to 4, as the standard for random forest regression is $floor(ncol(x)/3)$. A similar plot is shown as is used in section 3 for the random forest models, but this time the x-axis measures "mse_increase," which measures how much the mean square error of the model increases when the variable is removed. Again, variables with a higher importance will be higher and farther to the right on the graph. These metrics can be seen offensively in the left panel of Figure 9 and defensively in the right panel of Figure 9.

ORtg is again important in all the models examined. Min_per and usg are much more important in the numerical prediction models than they are in the classification models,

Table 4: Numerical prediction model performance metrics. The best method is bolded for each metric.

Model	Offensive MAE	Defensive MAE
Linear Regression	12.53	**1.22**
Neural Network	15.80	1.33
Random Forest	12.57	1.38
SVM	**12.29**	1.33

presumably because it is more important how much a player played when predicting their exact efficiency instead of simply improvement or lack thereof. fromp5 is important in the linear model but not in others, yet eFG is important in many models but not the linear one. Defensively, dbpm and top5 are very important. Min_per is not important, which is interesting because it is important in the classification models. blk_per is much more important than stl_per, which is a reverse of the findings of the classification model.

The SVM model performed best in terms of MAE for offensive projection, and the linear model performed best for defensive projection. Overall, the SVM and linear regression models clearly performed better than the other two models used for numerical prediction.

5 Conclusion and Discussion

This paper uses both classification models and numerical prediction models on the dataset. After KNN, logistic regression, LDA, neural network, random forest, and SVM models are created, the SVM model performed the best overall based on classification accuracy and MAE on the test set, offensively and defensively. Offensively, ORtg and eFG are the most important statistics across the board, and other factors like top5, fromp5, usg, min_per, and to_per have importance in some models, but not others. Factors like orb_per, ast_per, and ftr have little to no importance in most of the models. fromp5 has much more impact in the offensive models than top5, which means that the school a player is transferring from is more indicative of future offensive success than the school they are transferring to. Of the main areas of offensive basketball, shooting efficiency is the most indicative of future success, as

effective field goal percentage carried much more weight than statistics that measure the other areas of the game. Defensively, dbpm is the only stat that carried consistent importance. fromp5 and top5 have varying levels of importance, with top5 being more important than fromp5. So, oppositely from offense, it can be seen that future defensive impact is based more on where you are transferring to than where you are transferring from. stl_per and blk_per are important in some models but not others, and drb_per is important in very few of the models. Since orb_per and drb_per are both very unimportant factors, rebounding has very little impact on the study overall. Defensively, there is no aspect of defense that carried much more weight than other aspects, unlike in offense where shooting is clearly most important.

Future studies can build on this one as more data in college basketball becomes available.. Unlike with the NBA, there are very few sources of advanced college basketball player statistics. Another data issue arises from the unquantifiable nature of playing defense. There is simply no statistic that measures a player being able to shuffle his feet well or being able to get through screens better than most. In the future, statistics may be created that make "good" defense more quantifiable and paint a much fuller picture of defensive ability. Similar studies could also be conducted for women's college basketball as more data becomes available. The popularity of women's basketball has grown exponentially in the past few years, but finding advanced statistics for the women's game remains difficult.

References

Aleksandra Paluszyńska, Przemyslaw Biecek, and Yue Jiang (2020). randomForestExplainer: Explaining and Visualizing Random Forests in Terms of Variable Importance. https://cran.r-project.org/web/packages/randomForestExplainer.

Andy Liaw and Matthew Wiener (2002). Classification and Regression by randomForest. *R News*, 2(3):18–22.

Davis, S. A. (2023). An Exploratory Analysis of ACC Men's Basketball Players and the Transfer Portal in the Past 5 Years. *University of North Carolina Digital Repository*.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural Networks for Machine Learning Lecture 6a: Overview of Mini-Batch Gradient Descent. *Coursera, video lectures*.

Hunter, D., Yu, H., Pukish III, M. S., Kolbusz, J., and Wilamowski, B. M. (2012). Selection of Proper Neural Network Sizes and Architectures—A Comparative Study. *IEEE Transactions on Industrial Informatics*, 8(2):228–240.

Nguyen, N. H., Nguyen, D. T. A., Ma, B., and Hu, J. (2022). The Application of Machine Learning and Deep Learning in Sport: Predicting NBA Players' Performance and Popularity. *Journal of Information and Telecommunication*, 6(2):217–235.

Oliver, D. (2004). *Basketball on Paper: Rules and Tools for Performance Analysis*. Potomac, Washington, DC.

Ruby, U. and Yendapalli, V. (2020). Binary Cross Entropy with Deep Learning Technique for Image Classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(10).

Ruiz, F. J. and Perez-Cruz, F. (2015). A Generative Model for Predicting Outcomes in College Basketball. *Journal of Quantitative Analysis in Sports*, 11(1):39–52.

Thakur, A. (2023). What's the Optimal Batch Size to Train a Neural Network? https://wandb.ai/ayush-thakur/dl-question-bank/reports/What-s-the-Optimal-Batch-Size-to-Train-a-Neural-Network—VmlldzoyMDkyNDU.

West, B. T. (2006). A Simple and Flexible Rating Method for Predicting Success in the NCAA Basketball Tournament. *Journal of Quantitative Analysis in Sports*, 2(3).

Zimmermann, A., Moorthy, S., and Shi, Z. (2013). Predicting College Basketball Match

Outcomes Using Machine Learning Techniques: Some Results and Lessons Learned. *arXiv preprint arXiv:1310.3607*.