University of New Hampshire
University of New Hampshire Scholars' Repository

Honors Theses and Capstones | Student Scholarship

Spring 2019

Predictive Analytics for College Basketball: Using Logistic Regression for Determining the Outcome of a Game

Bryce Brown

Follow this and additional works at: https://scholars.unh.edu/honors

Part of the Business Analytics Commons

Recommended Citation

Brown, Bryce, "Predictive Analytics for College Basketball: Using Logistic Regression for Determining the Outcome of a Game" (2019). *Honors Theses and Capstones*. 475.
https://scholars.unh.edu/honors/475

This Senior Honors Thesis is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Honors Theses and Capstones by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.

Undergraduate Honors Thesis

Predictive Analytics for College Basketball: Using Logistic Regression for Determining the Outcome of a Game

By

Bryce Brown

Peter. T Paul College of Business and Economics

University of New Hampshire

Advisor:

Dr. Ali Hojjat

Assistant Professor of Decision Sciences

May 2019

Table of Contents

1. Introduction

College Basketball is one of the most popular sports in the country. A college basketball star, like Zion Williamson, can single-handedly affect the stock price of a company like Nike, by wearing one of their shoes. At the end of every year, a tournament is played called "March Madness". The top college basketball teams around the country play each other and millions of fans create forecasted brackets of the tournament and follow along.

The tournament is called "March Madness" for a reason. It is incredibly hard to predict the outcome of a game. Predictive analytics within college basketball has significantly grown over the years. Everyone wants to create a bracket with the highest accuracy. Fans all over the world are looking for ways to improve their brackets and stay involved as the tournament progresses.

In 2017, ESPN.com had 17.3 million March Madness brackets submitted to their website (Ota, 2018). A March Madness bracket with perfect accuracy has never been created before. In fact, ESPN.com has a free contest every year that awards one million dollars to an individual if they submit a perfect bracket. Last year alone, college basketball topped $1 billion in revenue (Rovell, 2018). It is no secret college basketball is vastly growing in popularity across the country every year.

The two primary research objectives in this thesis are:

1. Create a model that can help predict the winning team of a college basketball game given the historic performance metrics of the two teams.
2. Identify the performance metrics that are statistically significant in predicting the outcome of the game.

We build one separate model for each college basketball team using logistic regression methodology in R. We used historical data of division one basketball teams retrieved from *Kaggle.com*, to fit our model. The accuracy of our model in predicting the outcome of historic games varies from one team to another but ranges from 56% to 84% on training data, and from 21% to 97% on test data.

2. Literature Review

The outcome of a college basketball game is dichotomous: A team either wins or loses. The problem of predicting a categorical (in this case, binary) outcome is called classification, and there are several methodologies available in the literature for this purpose.

Shanahan (1984) built a logistic regression model to predict the probability of a win for a college basketball game. Shanahan used data from the University of Iowa men's and women's basketball teams from 1981-1983 and built a model for each team. Within those seasons, the men's team played 59 games and the women's team played 51. She started her model with 13 independent variables for the Women's model and 15 independent variables for the Men's model. Some of the variables in both models include: Assists, Personal Fouls, Field Goal Percentage, Defensive Rebounds, Total Rebounds, and Blocked Shots. Using backwards elimination, Shanahan then reduced the size of the two models to eight and six variables in the men's and women's team models, respectively. She interestingly found that the significant variables included in the women's model were more offensive-based, while the variables in the men's model was more defensive-based. Overall, her women's model had 90% accuracy in predicting the outcome of a game for that given season, and her men's model had 88% accuracy.

Magel and Unruh (2013) used Logistic Regression and least squares regression models with several explanatory variables such as home court advantage, difference in offensive rebounds, difference in defensive rebounds, difference in assists and difference in blocks to determine different outcomes pertaining to a college basketball game. The logistic regression model was used to determine a binary output (win or lose). The least squares regression model was used to determine the point spread of the final score between two teams of a specific game. The final logistic regression model 68% accuracy and the least squares final model had 64% accuracy.

Among classification methodologies, logistic regression particularly allows us to realize the relative importance of input variables in the prediction outcome and identify the significant variables. For example, Clark *et al.* (2013), from the Massachusetts Institute of Technology, used logistic regression to identify which factors have a significant impact on the success of a made field goal in the National Football League. In their research, they note how traditional analyses assume the main factor is the distance of the field goal, whereas after fitting a regression model, they find that Distance, Cold temperature, Field surface, Altitude, Precipitation, and Wind were all significant in determining the success of a made field goal in the NFL.

Aside from the National Football League, logistic regression has also been used in the Canadian Football League (CFL). Willoughby (2002), used win or lose as his dependent variable, and difference in passing yards, rushing yards, interceptions, fumbles and sacks as his independent variables. Willoughby specifically wanted to know which of these variables were most significant in predicting the outcome of a game for a winning or losing team. Willoughby analyzed three different teams, Calgary (a very good team), Saskatchewan (an average team), and Ottawa (a bad performing team). After fitting his model, Willoughby found that the difference in passing and rushing yards, along with interceptions, were most significant in predicting a win for a good team (Calgary and Saskatchewan), and less significant for bad teams

(Ottawa). Willoughby was able to conclude that a winning team in the CFL should be built around rushing, passing and trying to intercept the ball as much as possible.

Kvam and Sokol (2006) use Logistic Regression to estimate the probability that a team with a given margin of victory at home is better than its opponent. Their model specifically compares pairs of teams. For example, when team A beats team B at home by a certain margin of victory, the authors want to determine the probability that team A will then beat team B when they play at team B's home court. The probabilities of winning at both teams' locations with different margins of victory, helps determine which team will win if the two teams play a neutral game (neither home or away, which most March Madness games are). They used these results to create a ranking system of the teams in the March Madness tournament, then compared their ranking system to the five most commonly used NCAA ranking systems for predicting outcomes of games in the tournament. They found that their ranking system performed well (i.e., predicted a significant number of game outcomes) compared to the others.

Logistic Regression is not the only method for classification problems. Levandoski *et al.* (2017), used random forests methodology specifically for March Madness bracketing. According to Levandoski *et al.* (2017), random forests methodology works by creating a plethora of decision tree classifiers, and the final prediction is based on the mode of the results of those decision trees. Levandoski *et al.* (2017) trained their random forest classifier using 300 decision trees, each with a randomly selected subset of features, equal to the square root of the input dimensionality. Each decision tree in their model used 8 random features from a total of 57 features. They achieved a 68.9% accuracy using this method. They also compared their model against other classification methods such as: Neural Network (79.4%), Logistic Regression (76.2%), Bayes (69.8%), SVM (68.3%), Adaptive Boosting (66.7%), and K-nearest neighbors

(61.9%). The Logistic Regression method, which we use in this paper, outperformed all other methods by a noticeable margin, except the Neural Network technique.

Forsyth and Wilde (2014) used the K-nearest neighbors (kNN) classification method to predict the outcome of a college basketball game. This method compares new data to instances of similar data in the past to determine an outcome. For example, if a quicker team plays a taller team, the method will search through other match-ups where quicker teams played taller teams to determine the likelihood of a win for each team. This method is useful when past data is comprehensive and diverse enough to include a similar match-up (in every respect) to the game we are trying to predict. Forsyth and Wilde (2014) reported a 73% accuracy.

Along with finding the optimal method to use for predicting the outcome of a game, choosing the correct variables (attributes, statistics, metrics) to include in the model is just as important. Shi *et al.* (2013) fit a model using the "four Factors" (variables), that sports analyst Dean Oliver considers the most relevant in determining the outcome of a game. They are: Field Goal Percentage, Turnover Percentage, Offensive Rebound Percentage, and Free throw rate. Shi *et al.* (2013) also tested several different sets of variables as well as various other machine learning techniques such as decision trees, neural networks, and random forests. They received significantly different results when applying feature selection and learned that "the variables used to run the methods are ultimately what makes or breaks success." They experienced poor results when using very complex methods with a lot of variables and received better results when using simpler methods with fewer variables. This shows that a tremendous amount of due diligence is needed when determining which variables should be included in the model.

3. Dataset

For the purpose of training and testing our logistic regression model, we used the dataset from NCAA 2018 machine learning competition on *Kaggle.com*. The dataset includes historical performance metrics (statistics) observed across 82,041 basketball games from 364 different division one college basketball teams, between 2003 and 2018. For each game, the data includes the performance metrics for both opposing teams.

The specific performance metrics included in this dataset include:

- Season (Year)
- Win (Win:1, Loss:0)
- Score
- Number of Field Goals Made (FGM)
- Number of Field Goals Attempted (FGA)
- Number of Field Goals Made 3 (FGM3)
- Number of Field Goals Attempted 3 (FGA3)
- Number of Free Throws Made (FTM)
- Number of Free Throws Attempted (FTA)
- Number of Offensive Rebound (OR)
- Number of Defensive Rebound (DR)
- Number of Assists (AST)
- Number of Turnovers (TO)
- Number of Steals (STL)
- Number of Blocks (BLK)
- Number of Personal Fouls (PF)

Specifically, we used the data file called RegularSeasonDetailedResults.csv from Stage2UpdatedDataFiles.zip archive posted on the Kaggle competition site. We imported this data into R for the rest of our analysis.

4. Methodology

In this section we will elaborate on our logistic regression model, how we used R to transform raw data into an appropriate format for fitting logistic regression and discuss how we performed feature selection and data partitioning together to simplify our model and alleviate multicollinearity and overfit concerns.

4.1 Logistic Regression

In this research, we want to predict a categorical outcome of a college basketball game (0: lose, 1: win) for a specific college basketball team. This is considered a classification problem because the dependent output variable is binary (0/1) and not continuous (e.g., as demand or sales or market value of a car would be). Logistic regression is one of the powerful methodologies for binary classification problems.

Logistic regression works by using independent variables (also known as predictors, or features) to assess the probability of a dependent binary variable taking the success value (in our case, 1, representing a win). The mathematical formula for calculating the probability is as follows:

$$\text{Prob[Win]} = \frac{1}{1 + e^{-U}}$$

where

$$\mathrm{U} = \beta_0 + \beta_1 \mathrm{x}_1 + \beta_2 \mathrm{x}_2 + \beta_3 \mathrm{x}_3 + \cdots$$

and variables ($\mathrm{x}_{1,}\ \mathrm{x}_{2,}\ \mathrm{x}_3$, ...) are the input (predictor) variables. For our input variables, we use cumulative and moving averages of the historical performance metrics listed in the dataset section, as well as some nonlinear transformations of these metrics which we will further explain in the following sections.

The regression coefficients ($\beta_0, \beta_1, \beta_2, \beta_3, \ldots$) are fitted using Maximum Likelihood Estimation (MLE) on historical data. The regression coefficients should be interpreted as follows: each additional 1 unit increase (decrease) in a predictor variable (performance metric) x_i, multiplies (divides) the odds of winning, meaning Prob[Win]/Prob[Loss], by e^{β_i}.

There are several different ways we could use logistic regression. We could build a separate model for each pair of teams; A separate model for each team (against all others), or one single model to predict all games. Each college basketball team has different and unique historical performance metrics that may be significant in determining that particular team's success, therefore one single model may not perform well for every team. On the other hand, creating a separate model for each pair of teams, even though more customized, is not practically achievable due to the scarcity of data to support estimating the model coefficients and then validating the model. This is because most pairs of teams do not play against each other that often over the course of a decade. Therefore, we create a model for each team to strike a balance between customizing the model to each team, while having enough data to support a proper regression analysis.

A fitted regression model can further give significance values to each predictor variable known as p-values, that show how significant that variable is in the prediction of the dependent variable. The lower the p-value, the stronger the significance of that variable. A reader looking for more information regarding logistic regression may refer to *Best Practices in Logistic Regression* by Osborne (2015).

There are several important steps to consider when fitting a logistic regression model such as: data preparation, feature generation, variable selection, and data partitioning for model validation. We will discuss these steps in the next sections.

4.2 Data Preparation

The classification model should not use the performance result of a game after it has happened to predict the outcome of the same game. The input variables to the model on any game should only be based on the performance of the two teams as observed up to and prior to that game. Therefore, raw data as it appears in the dataset is not useful for fitting the model.

In our work, we calculated a 5-game moving average (MA) and a cumulative average (CA) of each performance metric for each team. For example, if teams 1 and 2 are playing on April 1st, 2018, the 5-game moving average would be the average of each performance metric for each team across the most recent 5 games preceding April 1st, 2018. The cumulative average would be the average of each performance metric for each team across all games played by that team prior to April 1st, 2018. The very first 4 games played by each team in history consequently had to be eliminated from the analysis due to not having a 5-game MA metric yet. We then used these MA and CA variables in place of the raw data to fit our model.

4.3 Feature Generation

Feature generation is a common idea in building strong predictive models where nonlinear transformations of original variables are added as additional variables in the model, hoping that some of these transformed variables would be significant and could improve the overall prediction accuracy.

In our work, we used the following nonlinear transformations of the moving and cumulative average performance statistics: Squared, Square root, Logarithm, Pairwise Ratios, and Pairwise Products. We added these variables to our dataset as new columns and after doing so ended up with a total of 282 input variables. These transformations were not possible on every performance metric, e.g., some leading to frequent division by zeros, and such cases were not generated in this process. Interestingly, and as we will describe in our results section, several of the most significant variables happen to be from these transformed variables that we generated.

4.4 Feature Selection

When fitting a logistic regression model the simpler model is always preferred to a more complex model, if they both yield a similar prediction accuracy. Generally speaking, there are three advantages in performing variable selection: 1) having a simpler model to work with, 2) correcting multicollinearity issues, and 3) alleviating overfit issues.

Having a simpler model to work with if the results are similar is preferred because it makes the model easier to use, explain, and interpret. Furthermore, fewer input metrics, meaning less data, needs to be collected for the purpose of prediction. Multicollinearity exists when independent (predictor) variables are highly correlated to one another. This causes inaccurate model, often with counter-intuitive coefficient signs (see Zainodin *et al.* 2011 for an example). Variable selection resolves multicollinearity issues by dropping one of the variables that are highly correlated. In our model, we particularly observed a strong multicollinearity issue involving variables FTM and FTA. Overfit occurs when the regression model is fitted extremely well to the historical data (e.g., high prediction accuracy) but is unable to predict similarly on brand new data. Overfitting can be caused by an abundance of predictor variables (Babyak, 2004). Within our model, we strive to resolve overfit issues using variable selection.

There are several methods for performing variable selection including: Backward elimination, forward selection, sequential replacement, and best subsets. Backward elimination starts with all predictor variables and then drops variables, one at a time, based on their (lack of) significance. Forward selection starts with no predictor variables and adds them, one a time, based on their significance. Sequential replacement is a method that combines the forward and backward ideas (Grisoni *et al*, 2014). The best subsets method works by exploring all possible subsets of predictor variables given a set number (constraint). This method is impractical to models with too many variables, since the number of subsets to try becomes prohibitively large (Hastie *et al*, 2008).

The best subset method is optimal but impractical for models beyond 15-20 variables. Among forward and backward, we found that backward leads to a model with higher accuracy in our application. We also found that having about 15 variables in the model is the sweet spot for simplicity of the model, yet giving a high accuracy, and having resolved most overfit concerns.

4.5 Data partitioning

Data partitioning is a standard practice for model validation. We specifically want to resolve any overfit issues. “Overfitting a model is a condition where a statistical model begins to describe the random error in the data rather than the relationship between variables” (Frost, 2019). An overfit model is so precisely fit to the original data that it is unable to replicate results on new data (Babyak, 2004). It is important to check for overfit issues to be sure that the model will work well when exposed to new data. Data partitioning allows us to check for overfit issues by splitting our data into a training set, which we use to fit our model, and a test set, which we use to confirm that the model (fitted on training data) gives a similar rate of correct predictions on a new but similar data which was not a part of fitting the model.

In our work, we used the games played by a team during 2003-2017 for training/fitting the model and held the data for games played during the 2018 season for validation. This left us with an average of 430 observations per team for the training set and 30 observations per team for the test set. As we will show in the following section, we found that our feature selection step and reducing the number of variables down to 15 resolved the overfit issue for most teams.

5. Results

In this section we show the results of our model and answer the two research objectives stated in the introduction. We fitted the logistic regression model and created an accuracy table presented in the table below. Along with the accuracy table, we identified the top 20 variables that were most often (that is, for many teams) deemed statistically significant in predicting the outcome of a game.

5.1 Prediction Accuracy on Twenty Well-Known Teams

The table below shows the prediction accuracy of our model, i.e., the percentage of times our model could correctly predict the winner of a game, for 20 of the most popular basketball teams. The complete table for 351 teams appears in the Appendix A, along with coefficients and p-value information. Teams that did not play in the 2018 season (which we considered to be our test data period) were not considered in the analysis.

The "Full model" is our logistic regression fitted with all variables. It is evident that with all variables the model is overfit. For example, the model build for Michigan State shows a 92% accuracy on the 2003-2017 data on which it was fitted, while showing only 18% accuracy when used to predict new games from the 2018 season. This shows that the model is unable to predict accurately when applied to brand new data.

The “Sub Model” is our model after performing a backward elimination of variables down to fifteen variables. It is evident that the overfit issues across most teams are resolved when variable selection is applied. Looking back at Michigan State, the training set accuracy is now 72%, which is lower than before. However, the test data accuracy of 76% gives us confidence that the model will deliver consistent accuracy when applied to brand new data.

	Full Model		Sub Model	
Team	Train	Test	Train	Test
Virginia	94%	85%	69%	91%
Gonzaga	95%	82%	84%	88%
Villanova	94%	38%	74%	88%
Purdue	92%	53%	69%	82%
Arizona	94%	74%	74%	79%
Kansas	94%	53%	84%	79%
Duke	94%	64%	82%	79%
Michigan St	92%	18%	72%	76%
Miami FL	93%	71%	66%	74%
Nevada	92%	74%	67%	74%
North Carolina	96%	66%	76%	71%
Kentucky	93%	71%	79%	71%
Houston	94%	45%	68%	70%
Texas Tech	94%	47%	69%	66%
Louisville	94%	58%	76%	61%
Florida	93%	41%	72%	59%
Tennessee	91%	45%	66%	58%
Marquette	93%	50%	68%	53%
Michigan	92%	50%	64%	47%
Auburn	92%	19%	62%	38%

Along with analyzing the best sub-model to use. We were able to identify which teams were most predictable (win or lose) against any given team. From our results, Virginia is the team with the highest accuracy. It can be inferred that Virginia plays more consistently than the 19 other college basketball teams in the table. Auburn, on the other hand, appears to be an unpredictable team.

To find the winner of one specific basketball game. For example, Virginia vs. Michigan. We would first use the Virginia model and fill in the opposing team's variables with Michigan's (MA, CA, and their transformed) statistics to assess the likelihood of Virginia winning. We could also use the Michigan model and fill in the opposing team's variables with Virginia's statistics to assess the likelihood of Michigan winning. If both models predict the same outcome, we could be fairly confident in the winner of the game. If the two models give different predictions, then we would probably trust the model that has shown higher accuracy on historical train and test data. If both models have low accuracy, then we would not be too confident in either one of the predictions.

5.2 Statistically Significant Performance Metrics

We identified which variables were most often deemed significant in predicting the outcome of a game by sorting the variables by the number of times they showed up as a significant variable across all the sub-models that we developed for the 351 different teams. The top 5 variables include: Square root of the cumulative average of field goals made, square root of the moving average of steals, cumulative average of score, moving average of personal fouls, and square root of the moving average of turnovers, all measured for the team of interest (and not the opposing team). The complete list of common variables appears below:

1. Square Root of the Cumulative Average of FMG1
2. Square Root of the Moving Average of STL1
3. Cumulative Average of the Score1
4. Moving Average of PF1
5. Square Root of the Moving Average of TO1
6. Square Root of the Cumulative Average of AST1
7. Cumulative Average of STL1
8. Square Root of the Moving Average of TO2
9. Cumulative Average of FGA1
10. Cumulative Average of DR1
11. Square Root of the Moving Average of PF1

12. Square Root of the Cumulative Average of FGA1
13. Square Root of the Cumulative Average of FTA1
14. Square Root of the Moving Average of BLK1
15. Square Root of the Cumulative Average of Score1
16. Cumulative Average of FGA31
17. Cumulative Average of FGM1
18. Square Root of the Moving Average of STL1
19. Square Root of the Cumulative Average of FTM1
20. Cumulative Average of FTA1

Variable acronyms were introduced before in our Dataset section 3. The numbers 1 & 2 after each variable are denoting the team of interest (for which the model is built) and the opposing team, respectively. An interesting discovery is that several of the top variables include the "square root" function. This proves that using feature generation in our research benefitted our model considering it provided most of the common significant variables. Furthermore, we observe that all top variables (except the 8th item in the list) pertain to the team of interest (for which the model is built) and not the opposing team. Appendix B provides a detail list of variables and coefficients for the 20 well-known basketball teams. Even though a few performance metrics from the opposing team do show up in most models, none of them is consistently a significant across multiple model to make our top-20 variable list, except for the square root of the moving average of turnovers.

6. Future Work

In our work, we built a logistic regression model to predict the winner of a college basketball game for 351 different teams. We transformed raw data into moving and cumulative averages, and created nonlinear transformations of these metrics to create even more features. We used backward elimination down to 15 variables to create a sub-model for each team to alleviate multicollinearity and overfit issues. We partitioned our data into a training and test for model validation. Our training set included data from 2003-2017, and we used data from 2018 as our test data. We were able to produce accuracy results for each team. We specifically found success in creating accurate models for some prominent teams. We were also able to identify which historical performance metrics were most commonly significant in the prediction of the outcome of a game.

There were a few key limitations that future research in this area may explore. First, we have accuracy results for each team against all teams. It would be interesting to see how accurate the predictions can be if we create a model for each pair of teams. For example, fitting a model specifically for Virginia vs. Michigan. The problem we faced was that most teams did not play each other enough times to have sufficient data for us to successfully fit and validate a model. This specific approach would be practical only for popular teams who play each other often. For example, the rivalry of Duke vs. North Carolina. This method could provide fans with a more customized tool to use when predicting the winner of a game.

Secondly, our model does not account for player injuries. Often times, a star player on a team can be the main producer for some of the performance metrics. If that player does not play in a certain game, the performance metrics could be completely different. One heuristic approach to bypass this limitation of the model could be to assess how much, on average, each individual player contributes to each of the team's overall performance metric (such as field goals made), to be able to determine how those metrics should be adjusted/scaled, if that player does not play, before inputting them in the logistic regression model.

Thirdly, our training and test data all pertained to regular season games only, and not from the end of the year tournament (i.e., the March Madness). Games played in the March Madness tournament are normally much more intense than a given regular season game and so the performance metrics for each team in the tournament could be drastically and characteristically different from those collected during the regular season. It would be interesting to see how performance metrics increase/decrease for each team during a game of higher intensity. Of course, only a few teams play in the tournament consistently and often (e.g., Duke, Virginia, or North Carolina); therefore, such analysis would not be an option for most NCAA basketball teams.

Finally, our work was limited to exploring the logistic regression methodology. It would be interesting to see how other classification methods such as: k-Nearest Neighbors, Support Vector Machine, Neural Networks, etc. would perform on the same data and using the same variables. The performance of different classification techniques is highly dependent on the data, therefore one of these alternative methods may very well lead to much more accurate predictions.

Moving forward, the same methods in this paper could be applied to other sports. Football and baseball are historically very analytical. There is a large amount of data available for both sports. It would be captivating to see if following the same steps we followed to develop and refine our logistic regression models could produce similar, or even better results, for some prominent football or baseball teams.

References

Babyak, Michael A. "What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models." *Psychosomatic medicine* 66, no. 3 (2004): 411-421.

Grisoni, F., M. Cassotti, and R. Todeschini. "Reshaped sequential replacement for variable selection in QSPR: Comparison with other reference methods." *Journal of Chemometrics* 28, no. 4 (2014): 249-259.

Clark, Torin, Aaron Johnson, and Alexander Stimpson. "Going for three: Predicting the likelihood of field goal success with logistic regression." In *The 7th Annual MIT Sloan Sports Analytics Conference*. 2013.

Forsyth, Jared, and Andrew Wilde. "A Machine Learning Approach to March Madness." Retrieved from: http://axon.cs.byu.edu/~martinez/classes/478/stuff/Sample_Group_Project3.pdf

Frost, Jim. "Overfitting Regression Models: Problems, Detection, and Avoidance" (2019). Retrieved from: https://statisticsbyjim.com/regression/overfitting-regression-models/

Hastie, Trevor., Tibshirani, Robert, and Friedman, Jerome. "The Elements of Statistical Learning" *Data Mining, Inference, and Prediction. Springer Series in Statistics. Second Edition. (2008):57-59.*

Kaggle.com "Google Cloud & NCAA Machine Learning Competition 2018 Dataset." Available online at: https://www.kaggle.com/c/mens-machine-learning-competition-2018/data

Kvam, Paul, and Joel S. Sokol. "A logistic regression/Markov chain model for NCAA basketball." *Naval Research Logistics (NrL)* 53, no. 8 (2006): 788-803.

Levandoski, Andrew, and Jonathan Lobo. "Predicting the NCAA Men's Basketball Tournament with Machine Learning." (2017). Retrieved from: http://jonathanlobo.com/docs/predicting_mm.pdf

Magel, Rhonda, and Samuel Unruh. "Determining factors influencing the outcome of college basketball games." *Open Journal of Statistics* 3, no. 04 (2013): 225.

Ota, K. "21st ESPN Tournament Challenge Collects 17.3 Million Brackets" (2018, March 15). Retrieved from: https://espnmediazone.com/us/press-releases/2018/03/21st-espn-tournament-challenge-collects-17-3-million-brackets/

Rovell, D. "NCAA tops $1 Billion in revenue during 2016-2017 school year" (2018, March 7). Retrieved from: http://www.espn.com/college-sports/story/_/id/22678988/ncaa-tops-1-billion-revenue-first

Shanahan, Kathleen Jean. "A model for predicting the probability of a win in basketball." (1984). Retrieved from: https://ir.uiowa.edu/etd/5082/

Shi, Z. Moorthy, S. Zimmermann, A. "Predicting NCAAB match outcomes using ML techniques – some results and lessons learned." (2013). Retrieved from: https://arxiv.org/pdf/1310.3607.pdf

Osborne, Jason. *Best Practices in Logistic Regression.* USA: SAGE publications, Inc. (2015).

Willoughby, Keith A. "Winning games in Canadian football: A logistic regression analysis." *The College Mathematics Journal*33, no. 3 (2002): 215-220.

Zainodin, H. J., A. Noraini, and S. J. Yap. "An alternative multicollinearity approach in solving multiple regression problem." *Trends in Applied Sciences Research* 6, no. 11 (2011): 1241-1255.

Appendix

A. Prediction Accuracy for All Teams

The table below is the accuracy table for all 351 teams. For each team, the full model contains all 282 variables, whereas the sub-model contains only 15 variables, identified using backward elimination. Each model is trained on 2003-2017 data and tested on 2018 data. The table is sorted by the accuracy of sub-model on test data, from highest to lowest.

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Chicago St	97%	77%	75%	97%
Delaware St	92%	67%	74%	93%
Virginia	94%	85%	69%	91%
Bryant	100%	6%	72%	90%
Maine	96%	73%	69%	90%
Houston Bap	100%	14%	73%	89%
Cincinnati	93%	68%	68%	88%
Gonzaga	95%	82%	84%	88%
Villanova	94%	38%	74%	88%
Pittsburgh	91%	59%	76%	88%
Alabama A&M	97%	19%	70%	87%
Longwood	98%	63%	78%	87%
Alcorn St	95%	54%	77%	86%
CS Northridge	94%	25%	64%	86%
San Jose St	96%	89%	74%	86%
Xavier	91%	42%	71%	85%
MS Valley St	94%	81%	67%	84%
St Mary's CA	95%	75%	73%	84%
Coppin St	95%	40%	71%	83%
MD E Shore	96%	33%	76%	83%
Northern Arizona	93%	17%	63%	83%
Savannah St	97%	43%	73%	83%
Purdue	92%	53%	69%	82%
SC Upstate	100%	50%	69%	82%
Buffalo	92%	50%	63%	81%
Marist	93%	61%	64%	81%
Detroit	93%	50%	68%	80%
MTSU	93%	37%	71%	80%
SF Austin	95%	40%	69%	80%
Arizona	94%	74%	74%	79%
Kansas	94%	53%	84%	79%
Missouri KC	96%	48%	68%	79%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
UC Riverside	96%	72%	70%	79%
BYU	92%	48%	72%	79%
Duke	94%	64%	82%	79%
Prairie View	93%	52%	71%	79%
Vermont	95%	67%	72%	79%
Charlotte	94%	71%	60%	79%
Wichita St	94%	50%	73%	78%
Fordham	96%	61%	68%	77%
Old Dominion	93%	32%	67%	77%
Dartmouth	95%	81%	73%	77%
Albany NY	91%	63%	66%	77%
Norfolk St	93%	57%	67%	77%
Arkansas	90%	59%	70%	76%
Bucknell	92%	30%	67%	76%
Florida A&M	95%	67%	72%	76%
Michigan St	92%	18%	72%	76%
Air Force	95%	32%	73%	75%
Belmont	93%	38%	71%	75%
IPFW	95%	61%	67%	75%
Rice	93%	25%	66%	75%
St Bonaventure	95%	69%	65%	75%
VMI	95%	57%	68%	75%
Howard	95%	81%	75%	74%
James Madison	92%	52%	61%	74%
Miami FL	93%	71%	66%	74%
ULL	92%	77%	61%	74%
UMBC	96%	61%	73%	74%
Portland	93%	63%	65%	74%
Stetson	94%	52%	68%	74%
Nevada	92%	74%	67%	74%
TX Southern	96%	68%	73%	74%
Georgetown	93%	40%	73%	73%
New Mexico St	92%	70%	70%	73%
G Washington	93%	45%	69%	73%
Murray St	95%	72%	70%	72%
Presbyterian	100%	55%	76%	72%
ETSU	90%	69%	65%	72%
Rhode Island	93%	66%	67%	72%
Citadel	97%	32%	78%	71%
New Hampshire	93%	36%	66%	71%
North Carolina	96%	66%	76%	71%
Oral Roberts	91%	57%	65%	71%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Illinois	90%	42%	71%	71%
Mississippi St	92%	52%	66%	71%
Pacific	94%	58%	66%	71%
Brown	94%	63%	64%	71%
Kentucky	93%	71%	79%	71%
Kennesaw	99%	63%	77%	70%
Creighton	91%	53%	71%	70%
Edwardsville	100%	60%	76%	70%
Fresno St	90%	43%	65%	70%
S Carolina St	93%	67%	68%	70%
Seattle	100%	43%	67%	70%
Houston	94%	45%	68%	70%
Stanford	95%	61%	67%	70%
Jackson St	93%	38%	56%	69%
Morgan St	93%	55%	68%	69%
S Dakota St	97%	21%	70%	69%
TN Martin	98%	48%	70%	69%
Utah Valley	98%	31%	62%	69%
Youngstown St	93%	31%	73%	69%
La Salle	94%	63%	67%	69%
South Florida	94%	59%	66%	69%
Syracuse	94%	66%	75%	69%
UC Irvine	90%	69%	59%	69%
Ark Pine Bluff	94%	54%	76%	69%
McNeese St	96%	68%	62%	68%
Denver	90%	61%	57%	68%
E Kentucky	93%	57%	63%	68%
FL Atlantic	94%	68%	67%	68%
IUPUI	94%	57%	64%	68%
Santa Barbara	92%	46%	62%	68%
Colorado	92%	58%	69%	68%
DePaul	93%	61%	69%	68%
Holy Cross	91%	42%	63%	68%
UNC Greensboro	96%	77%	67%	68%
Wright St	90%	65%	63%	68%
West Virginia	92%	65%	67%	68%
CS Bakersfield	100%	63%	69%	67%
Hofstra	96%	57%	66%	67%
Kansas St	93%	58%	67%	67%
Montana St	94%	47%	63%	67%
Quinnipiac	91%	33%	61%	67%
Samford	93%	26%	67%	67%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
UNLV	93%	52%	66%	67%
Grand Canyon	100%	50%	81%	66%
Rutgers	95%	59%	69%	66%
Texas Tech	94%	47%	69%	66%
UCLA	91%	50%	73%	66%
Army	97%	69%	66%	66%
Bethune-Cookman	95%	34%	68%	66%
Binghamton	94%	41%	70%	66%
Bowling Green	91%	59%	64%	66%
CS Sacramento	95%	62%	67%	66%
E Illinois	94%	62%	65%	66%
Idaho	90%	52%	63%	66%
St Francis NY	94%	62%	61%	66%
Winthrop	93%	62%	69%	65%
Oregon	93%	62%	70%	65%
Providence	94%	47%	68%	65%
Alabama St	95%	77%	69%	65%
Arizona St	90%	52%	66%	65%
Delaware	91%	58%	65%	65%
Florida St	92%	45%	68%	65%
Ga Southern	93%	55%	62%	65%
St John's	93%	68%	67%	65%
Washington St	94%	58%	66%	65%
ULM	96%	43%	71%	64%
Weber St	94%	36%	67%	64%
Oakland	92%	70%	69%	64%
Lafayette	95%	60%	66%	63%
N Illinois	94%	63%	70%	63%
North Florida	98%	63%	73%	63%
South Dakota	100%	60%	68%	63%
Southern Utah	95%	43%	71%	63%
UTRGV	97%	57%	75%	63%
Gardner Webb	89%	56%	64%	63%
Lamar	92%	59%	64%	63%
New Orleans	97%	48%	64%	63%
NJIT	100%	44%	73%	63%
Connecticut	92%	72%	70%	63%
Duquesne	94%	47%	70%	63%
George Mason	92%	53%	65%	63%
Iona	90%	47%	65%	63%
Minnesota	92%	78%	66%	63%
Seton Hall	94%	50%	70%	63%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Siena	90%	56%	63%	63%
F Dickinson	96%	69%	67%	62%
Florida Intl	93%	45%	70%	62%
N Kentucky	100%	69%	69%	62%
Sam Houston St	95%	55%	68%	62%
Memphis	94%	44%	77%	62%
Penn St	95%	47%	69%	62%
Cornell	94%	73%	71%	62%
Baylor	95%	55%	67%	61%
Davidson	95%	39%	72%	61%
Liberty	96%	42%	66%	61%
Maryland	93%	65%	70%	61%
Montana	93%	23%	66%	61%
Oklahoma	95%	55%	69%	61%
San Diego St	93%	71%	71%	61%
Tulane	94%	45%	69%	61%
UNC Asheville	93%	39%	63%	61%
Arkansas St	94%	39%	64%	61%
Campbell	92%	64%	69%	61%
Charleston So	96%	50%	67%	61%
Georgia	91%	42%	68%	61%
Louisville	94%	58%	76%	61%
Notre Dame	93%	67%	69%	61%
UT Arlington	92%	52%	59%	61%
Boise St	91%	60%	66%	60%
East Carolina	95%	53%	68%	60%
Elon	94%	43%	69%	60%
Hampton	90%	67%	68%	60%
Harvard	96%	47%	66%	60%
Sacred Heart	91%	33%	63%	60%
South Alabama	93%	60%	65%	60%
W Carolina	92%	63%	66%	60%
Wagner	89%	47%	60%	60%
Butler	95%	53%	71%	59%
Clemson	90%	41%	65%	59%
Florida	93%	41%	72%	59%
Georgia St	92%	69%	65%	59%
Georgia Tech	89%	72%	63%	59%
Marshall	93%	66%	67%	59%
Massachusetts	91%	56%	61%	59%
S Illinois	93%	56%	62%	59%
Texas A&M	93%	53%	69%	59%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Toledo	91%	69%	66%	59%
UCF	91%	53%	67%	59%
Abilene Chr	100%	52%	79%	59%
Cleveland St	95%	68%	68%	59%
Cal Poly SLO	92%	66%	66%	59%
Lehigh	91%	38%	65%	59%
SE Missouri St	94%	41%	69%	59%
Southern Miss	96%	52%	68%	59%
W Illinois	98%	71%	74%	58%
California	89%	81%	65%	58%
Drexel	95%	35%	63%	58%
FL Gulf Coast	100%	45%	66%	58%
Louisiana Tech	94%	45%	72%	58%
Manhattan	91%	58%	64%	58%
NC Central	100%	58%	78%	58%
Tulsa	91%	39%	66%	58%
Long Island	93%	52%	62%	58%
Robert Morris	93%	39%	65%	58%
Tennessee	91%	45%	66%	58%
Boston Univ	94%	50%	64%	57%
Hawaii	91%	43%	65%	57%
Idaho St	95%	54%	71%	57%
Santa Clara	93%	46%	62%	57%
St Francis PA	94%	68%	69%	57%
Evansville	94%	47%	66%	57%
Loyola MD	89%	67%	67%	57%
LSU	92%	47%	65%	57%
Mercer	94%	47%	70%	57%
Mt St Mary's	90%	37%	66%	57%
North Texas	93%	70%	64%	57%
Portland St	96%	53%	66%	57%
Towson	94%	57%	69%	57%
UAB	94%	50%	70%	57%
E Washington	93%	50%	64%	56%
NC State	91%	38%	70%	56%
Ohio St	94%	38%	75%	56%
San Francisco	94%	50%	65%	56%
Temple	92%	66%	67%	56%
Vanderbilt	92%	50%	67%	56%
USC	93%	56%	60%	56%
Morehead St	96%	33%	66%	56%
Coastal Car	94%	69%	60%	55%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Indiana St	94%	59%	61%	55%
Ohio	89%	59%	63%	55%
W Michigan	91%	48%	60%	55%
Central Conn	92%	48%	67%	55%
Iowa St	93%	39%	68%	55%
Pepperdine	93%	58%	62%	55%
Radford	92%	35%	68%	55%
Tennessee Tech	91%	58%	63%	55%
Troy	91%	52%	62%	55%
WI Milwaukee	91%	42%	63%	55%
Iowa	92%	61%	65%	55%
Kent	92%	58%	69%	55%
New Mexico	93%	61%	68%	55%
VA Commonwealth	93%	58%	73%	55%
Incarnate Word	100%	21%	77%	54%
High Point	93%	42%	66%	54%
MA Lowell	100%	50%	71%	54%
Ball St	95%	67%	64%	53%
Miami OH	92%	60%	67%	53%
Yale	93%	67%	68%	53%
Illinois St	94%	59%	62%	53%
Marquette	93%	50%	68%	53%
Richmond	92%	44%	61%	53%
St Joseph's PA	92%	47%	65%	53%
St Louis	96%	56%	65%	53%
Nicholls St	98%	32%	74%	52%
E Michigan	92%	48%	66%	52%
Jacksonville	95%	52%	68%	52%
Southern Univ	94%	45%	65%	52%
Drake	94%	55%	63%	52%
Indiana	91%	58%	65%	52%
Long Beach St	94%	65%	61%	52%
Rider	91%	55%	68%	52%
Fairfield	91%	58%	66%	52%
Texas	91%	55%	71%	52%
Alabama	91%	44%	67%	50%
American Univ	94%	77%	63%	50%
Appalachian St	92%	53%	63%	50%
Cent Arkansas	100%	50%	77%	50%
IL Chicago	95%	50%	67%	50%
Loy Marymount	94%	63%	64%	50%
Missouri St	89%	50%	62%	50%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Northwestern	92%	57%	63%	50%
Oregon St	91%	56%	61%	50%
SMU	94%	59%	68%	50%
South Carolina	90%	72%	65%	50%
St Peter's	94%	53%	66%	50%
TCU	95%	63%	72%	50%
Washington	94%	38%	67%	50%
Oklahoma St	92%	58%	71%	48%
Furman	96%	74%	67%	48%
Missouri	93%	55%	71%	48%
Texas St	96%	42%	64%	48%
Utah St	94%	52%	71%	48%
UC Davis	96%	38%	63%	48%
UT San Antonio	94%	62%	63%	48%
Wofford	93%	41%	69%	48%
Michigan	92%	50%	64%	47%
Nebraska	93%	56%	71%	47%
Virginia Tech	92%	53%	59%	47%
Austin Peay	92%	53%	65%	47%
C Michigan	90%	57%	63%	47%
Grambling	97%	40%	75%	47%
SE Louisiana	96%	60%	68%	47%
WI Green Bay	91%	43%	68%	47%
Princeton	94%	57%	63%	46%
Wisconsin	94%	45%	76%	45%
Akron	91%	39%	68%	45%
Dayton	92%	52%	67%	45%
Hartford	93%	77%	66%	45%
Jacksonville St	94%	52%	66%	45%
Penn	95%	65%	68%	45%
Stony Brook	93%	52%	70%	45%
NE Omaha	100%	45%	74%	45%
William & Mary	95%	69%	64%	45%
Boston College	93%	47%	68%	44%
Northwestern LA	95%	32%	65%	44%
Canisius	93%	50%	64%	44%
Niagara	90%	44%	70%	44%
Wyoming	90%	44%	62%	44%
Northern Iowa	90%	57%	65%	43%
San Diego	92%	53%	67%	43%
Chattanooga	93%	71%	64%	43%
CS Fullerton	92%	43%	64%	43%

Team Name	Full Model Train	Full Model Test	Sub Model Train	Sub Model Test
Tennessee St	94%	46%	68%	43%
WKU	92%	30%	66%	42%
Mississippi	90%	65%	69%	42%
NC A&T	95%	45%	72%	42%
N Dakota St	99%	41%	67%	41%
Bradley	92%	50%	63%	41%
Col Charleston	92%	77%	65%	40%
Monmouth NJ	94%	43%	66%	40%
N Colorado	96%	61%	69%	39%
Navy	95%	61%	64%	39%
Wake Forest	93%	48%	68%	39%
Valparaiso	95%	34%	64%	38%
Auburn	92%	19%	62%	38%
Colgate	93%	50%	65%	37%
Utah	93%	47%	63%	37%
Columbia	91%	72%	63%	36%
Colorado St	92%	32%	64%	35%
UNC Wilmington	93%	31%	68%	34%
Loyola-Chicago	94%	34%	64%	34%
Northeastern	94%	66%	63%	34%
UTEP	92%	57%	65%	33%
North Dakota	100%	47%	66%	30%
TAM C. Christi	95%	42%	71%	27%
Lipscomb	93%	76%	66%	24%
Ark Little Rock	90%	72%	60%	21%

B. Coefficient and P-Value Information for Individual Teams

In this section, we provide the exact list of 15 variables that remained in the sub-model of the 20 well-known teams listed in the results section, along with the corresponding standard errors, t-scores, and p-value information. The variable names are composed of the performance metric acronyms defined in the Data section 3, followed by either "MA" meaning Moving Average or "CA" meaning Cumulative Average. A number "1" in the variable name means the variable pertains to the performance of the team for which the model is constructed, whereas "2" refers to a performance metric of the opposing team. A prefix denotes a nonlinear transformation performed on the variable before it is used in the model, as introduced in the Feature Generation section 4.3. "Sq" means Squared, "sqrt" means Square Root, "log" means logarithm, "rat" means Ratio of that metric for team 1 over team 2, and "mult" means Product of that metric for teams 1 and 2.

Arizona:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.661e+02  5.542e+02   0.480   0.6313
sqrtTO1cs    2.319e-02  5.250e-01   0.044   0.9648
OR1cs       -2.812e-02  4.808e-02  -0.585   0.5590
sqFTM1cs    -1.405e-03  4.154e-03  -0.338   0.7353
sqBlk1ma     7.258e-04  2.520e-03   0.288   0.7735
ratFTA1ma   -1.676e-04  2.021e-04  -0.829   0.4074
sqrtFTM1ma  -8.330e-01  9.959e-01  -0.836   0.4034
DR1ma        1.709e-02  9.564e-03   1.787   0.0745 .
logFTM1ma    1.640e+00  1.974e+00   0.831   0.4063
sqrtFTA2cs  -6.992e+02  1.328e+03  -0.527   0.5987
logFTA2cs    6.057e+02  1.123e+03   0.539   0.5899
FTA2cs       5.636e+01  1.101e+02   0.512   0.6089
sqFTA2cs    -2.119e-01  4.459e-01  -0.475   0.6349
ratFTA1cs   -7.744e-04  1.078e-02  -0.072   0.9427
sqrtFTA1cs   9.130e-01  2.395e+00   0.381   0.7032
Score1cs    -4.430e-04  1.713e-02  -0.026   0.9794
```

Auburn:

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.0716546  3.6367623   1.395  0.16385
FTM2cs       0.0354486  0.0575406   0.616  0.53817
FTM2ma      -0.0229371  0.0075792  -3.026  0.00262 **
sqBlk1ma    -0.0065885  0.0055998  -1.177  0.24000
multFTM1cs   0.9489960  0.7470078   1.270  0.20461
logTO1ma    -5.7249536  3.3401213  -1.714  0.08723 .
sqrtTO1ma    2.9456455  1.7834339   1.652  0.09931 .
FGM1cs       0.1362618  0.1341509   1.016  0.31031
sqrtBlk1ma   0.3391414  0.1850435   1.833  0.06751 .
sqrtStl1ma  -0.3833006  2.7673917  -0.139  0.88990
logStl1ma    0.6930677  2.6728551   0.259  0.79553
Score1cs    -0.0630628  0.0746461  -0.845  0.39866
PF1ma        0.0242419  0.0121925   1.988  0.04740 *
logFTA1ma   -1.7072218  1.8406528  -0.928  0.35417
sqrtFTA1ma   0.6883877  0.8154833   0.844  0.39904
sqStl1ma     0.0008908  0.0093372   0.095  0.92403
```

Duke:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	2.413e+00	2.640e+00	0.914	0.3611	
logOR2cs	-2.424e-01	1.664e-01	-1.457	0.1457	
sqTO1cs	-1.346e-03	2.379e-03	-0.566	0.5718	
multDR1cs	4.303e-01	3.600e-01	1.195	0.2326	
Blk1ma	9.537e-03	1.508e-02	0.632	0.5275	
multTO1cs	3.564e-01	6.512e-01	0.547	0.5845	
ratDR1ma	-2.846e-04	3.694e-04	-0.770	0.4414	
sqrtAst2ma	-1.439e-02	6.254e-02	-0.230	0.8181	
logDR2ma	-1.217e-02	2.974e-01	-0.041	0.9674	
FTA1cs	1.464e-02	2.275e-02	0.643	0.5204	
FGM31cs	6.748e-02	1.391e-01	0.485	0.6279	
sqFGA2ma	3.326e-05	4.123e-05	0.807	0.4202	
sqTO2cs	1.861e-03	1.776e-03	1.048	0.2954	
sqPF1cs	-1.283e-03	1.395e-03	-0.920	0.3582	
multFGM1ma	3.565e-01	1.521e-01	2.344	0.0195	*
DR1cs	-1.096e-01	9.125e-02	-1.201	0.2305	

Florida:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-3.943e+05	3.471e+05	-1.136	0.2566	
DR2cs	-4.342e-02	1.790e-02	-2.426	0.0156	*
Score2cs	-3.217e-02	2.067e-01	-0.156	0.8764	
sqrtScore2cs	1.422e-01	3.453e+00	0.041	0.9672	
FTM2cs	-1.489e+00	1.603e+00	-0.929	0.3534	
sqFTM2cs	2.012e-02	2.311e-02	0.871	0.3844	
sqrtFTM2cs	7.070e+00	7.151e+00	0.989	0.3233	
sqrtAst2ma	-1.321e-01	5.934e-02	-2.226	0.0265	*
sqScore1cs	-1.174e+01	1.046e+01	-1.122	0.2624	
sqrtScore1cs	-2.679e+05	2.374e+05	-1.128	0.2597	
Score1cs	1.125e+04	9.987e+03	1.126	0.2606	
sqrtStl1ma	-2.275e-01	2.540e-01	-0.896	0.3708	
sqStl1ma	3.781e-03	3.273e-03	1.155	0.2487	
sqrtAst1cs	7.068e-01	5.972e-01	1.183	0.2373	
logScore1cs	4.486e+05	3.968e+05	1.131	0.2588	
FGM1cs	-1.391e-02	1.147e-01	-0.121	0.9035	

Gonzaga:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-2.744e+01	3.101e+01	-0.885	0.3767	
FTM2cs	7.626e-02	4.551e-02	1.675	0.0945	.
FTA2cs	-4.594e-02	2.869e-02	-1.601	0.1100	
sqrtFGM1cs	8.079e+00	8.560e+00	0.944	0.3458	
sqrtOR1ma	1.008e-01	7.820e-02	1.289	0.1982	
ratFTA1ma	-3.183e-05	1.516e-04	-0.210	0.8339	
sqrtOR2ma	-5.172e-02	5.452e-02	-0.949	0.3433	
sqFGM2cs	6.538e-04	9.574e-04	0.683	0.4950	
logStl1ma	7.606e-02	7.655e-02	0.994	0.3210	
sqrtBlk1ma	1.292e-02	6.127e-02	0.211	0.8331	
ratScore1cs	-3.943e-04	2.669e-04	-1.477	0.1403	
OR1cs	2.353e-02	9.936e-02	0.237	0.8129	
sqrtFTM1cs	3.282e+00	3.988e+00	0.823	0.4109	
logFGM31cs	3.279e+00	3.221e+00	1.018	0.3091	
sqrtFGA31cs	-2.003e-01	5.662e-01	-0.354	0.7237	
Score1cs	-4.061e-01	4.485e-01	-0.905	0.3657	

Houston:

```
     Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.670e+00  8.989e+00  -0.297 0.766559
Stl2ma      -5.325e-02  1.428e-02  -3.728 0.000218 ***
sqStl2cs    -2.726e-03  6.729e-03  -0.405 0.685580
sqPF2cs      4.589e-03  4.635e-03   0.990 0.322731
TO2cs        4.416e-02  1.256e-01   0.352 0.725380
sqrtFGM1ma  -1.246e+00  2.351e+00  -0.530 0.596379
sqrtPF2cs   -1.388e+00  1.542e+00  -0.900 0.368603
sqTO2cs      1.023e-03  3.989e-03   0.256 0.797759
logFGM1ma    3.755e+00  5.909e+00   0.635 0.525448
sqrtStl2cs   8.015e-02  6.087e-01   0.132 0.895291
ratFTA1ma   -2.144e-04  2.068e-04  -1.037 0.300383
Ast1ma      -7.208e-03  1.315e-02  -0.548 0.584033
DR2ma       -1.742e-02  8.172e-03  -2.132 0.033586 *
ratFGA31ma   7.832e-05  1.872e-04   0.418 0.675879
sqBlk1cs     5.188e-03  9.746e-03   0.532 0.594758
PF1cs        8.534e-02  4.047e-02   2.109 0.035546 *
```

Kansas:

```
       Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.278e+00  1.158e+02   0.046   0.96365
sqTO2ma       6.822e-04  2.765e-04   2.467   0.01398 *
sqrtFGM1cs   -9.913e+01  7.626e+01  -1.300   0.19431
sqrtScore1cs  6.746e+01  3.749e+01   1.799   0.07259 .
Score1cs      8.324e-01  1.847e+00   0.451   0.65236
logTO1ma      6.045e-03  1.540e-01   0.039   0.96871
sqrtFGM31cs  -6.782e+01  5.585e+01  -1.214   0.22523
FTM1cs       -4.592e+00  3.532e+00  -1.300   0.19416
logFGM31cs    4.687e+01  4.226e+01   1.109   0.26797
sqFGM31cs     1.409e-01  2.849e-01   0.495   0.62109
sqrtTO1cs     2.393e+00  8.709e-01   2.748   0.00623 **
Blk1cs        1.814e-01  1.170e-01   1.551   0.12161
logAst1cs     2.397e-04  1.901e+00   0.000   0.99990
sqrtFGA31ma  -1.399e-01  3.870e-01  -0.361   0.71792
OR1ma         1.179e-02  1.051e-02   1.122   0.26235
sqFGA31ma     6.783e-04  1.357e-03   0.500   0.61748
```

Kentucky:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.7945680 34.8603512  -0.051 0.958966
sqTO2ma       0.0001764  0.0003395   0.520 0.603589
sqPF1ma      -0.0002319  0.0002459  -0.943 0.346099
Stl1ma        0.0060502  0.0124325   0.487 0.626738
multBlk1cs   -0.0361203  0.0376203  -0.960 0.337483
sqTO2cs      -0.0035723  0.0034686  -1.030 0.303599
logBlk2cs    -0.2585461  0.1194601  -2.164 0.030944 *
sqFGM1ma     -0.0047595  0.0147192  -0.323 0.746573
FGM1ma        0.6890265  2.4913900   0.277 0.782237
sqrtTO2cs     1.0651404  0.7865275   1.354 0.176312
sqrtFGM1ma   -4.4205071 17.5738686  -0.252 0.801507
logBlk1cs     0.8341037  0.1970961   4.232 2.79e-05 ***
sqrtPF1cs     1.7313175  0.6025184   2.873 0.004243 **
sqrtScore2ma -0.1643967  0.0493351  -3.332 0.000929 ***
FGA2ma       -0.0002112  0.0052496  -0.040 0.967923
logPF2cs      0.1368484  0.2866445   0.477 0.633288
```

Louisville:

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.159e+02  3.905e+02  -1.577    0.115
sqrtStl2ma   6.869e-02  5.454e-02   1.260    0.208
sqrtFTA2ma  -6.273e-02  4.209e-02  -1.490    0.137
sqrtBlk1ma   8.192e-02  6.747e-02   1.214    0.225
FGM1cs       7.635e+01  4.960e+01   1.539    0.124
sqrtTO1ma   -1.336e+00  1.401e+00  -0.954    0.341
logTO1ma     2.085e+00  2.455e+00   0.849    0.396
sqrtFTM1ma  -4.362e-02  5.438e-02  -0.802    0.423
logBlk2cs   -2.872e-01  6.285e-02  -4.570 6.26e-06 ***
sqrtAst1cs   4.816e-01  7.576e-01   0.636    0.525
sqrtOR1cs   -5.755e-01  6.949e-01  -0.828    0.408
sqrtFGM1cs  -1.589e+03  1.029e+03  -1.544    0.123
logFGM1cs    2.067e+03  1.334e+03   1.549    0.122
sqrtPF1ma   -6.917e-01  1.808e+00  -0.383    0.702
Score1cs    -1.951e-02  2.623e-02  -0.744    0.457
logPF1ma     1.431e+00  3.941e+00   0.363    0.717
```

Marquette:

```
           Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.120e+02  6.566e+01   1.706 0.088638 .
Blk1cs        6.507e-02  1.305e-01   0.498 0.618386
FGA32cs      -2.970e-02  1.027e-01  -0.289 0.772606
sqFGA32cs     7.884e-05  2.716e-03   0.029 0.976854
FGM2cs       -5.315e-02  1.376e-02  -3.862 0.000129 ***
PF2ma         1.111e-02  9.374e-03   1.185 0.236608
multTO1cs    -8.777e-01  2.400e-01  -3.657 0.000286 ***
sqrtScore1ma  1.488e+00  6.637e-01   2.243 0.025411 *
FGA31ma       1.548e-02  1.186e-02   1.304 0.192778
sqScore1ma   -5.455e-04  2.578e-04  -2.116 0.034865 *
PF2cs         1.193e+01  6.891e+00   1.731 0.084066 .
FGM31ma      -2.519e-02  2.435e-02  -1.034 0.301485
sqPF2cs      -9.873e-02  5.975e-02  -1.652 0.099172 .
sqrtPF2cs    -7.113e+01  4.019e+01  -1.770 0.077428 .
sqFTA1ma     -8.333e-06  1.461e-04  -0.057 0.954550
sqBlk2ma     -2.681e-03  1.559e-03  -1.720 0.086088 .
```

Miami FL

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.460e+02  1.748e+02   0.835   0.4039
sqTO2ma       1.231e-03  2.977e-04   4.136 4.22e-05 ***
sqrtPF1ma     6.322e+00  6.337e+01   0.100   0.9206
logFTA2cs    -9.040e-01  2.298e-01  -3.933 9.70e-05 ***
logPF1ma     -8.120e+00  6.531e+01  -0.124   0.9011
Ast2ma       -1.653e-02  8.647e-03  -1.912   0.0566 .
sqrtDR1cs    -6.223e+00  4.278e+00  -1.454   0.1465
TO1cs        -3.564e-02  3.913e-02  -0.911   0.3630
sqDR1cs       1.273e-02  8.540e-03   1.490   0.1369
sqrtScore1cs  1.077e+01  1.790e+01   0.602   0.5478
logScore1cs  -4.889e+01  7.804e+01  -0.626   0.5313
sqrtTO1ma     6.625e-01  1.070e+00   0.619   0.5362
PF1ma        -2.975e-01  3.831e+00  -0.078   0.9381
FGM31cs       2.595e-01  1.399e-01   1.855   0.0642 .
FGA1cs       -1.834e-02  3.863e-02  -0.475   0.6352
logTO1ma     -9.630e-01  1.866e+00  -0.516   0.6062
```

Michigan:

```
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.021e+01  2.009e+01   1.006  0.31486    
Stl2ma       5.992e-03  1.323e-02   0.453  0.65081    
sqFTA2ma    -3.773e-04  1.354e-04  -2.786  0.00556 ** 
logFGM31cs   1.628e+01  1.066e+01   1.528  0.12726    
sqrtFGM31cs -1.319e+01  8.733e+00  -1.511  0.13152    
FTM1cs      -2.525e-01  2.461e-01  -1.026  0.30541    
sqrtFTA1cs   8.232e+00  8.040e+00   1.024  0.30644    
logFTA1cs   -1.787e+01  1.837e+01  -0.973  0.33116    
DR1cs       -2.418e-02  1.211e-01  -0.200  0.84185    
PF2ma        1.594e-02  8.981e-03   1.774  0.07665 .  
Blk1cs      -4.918e-01  1.604e-01  -3.066  0.00230 ** 
multOR1cs   -5.595e-02  1.280e+00  -0.044  0.96517    
sqrtOR1cs    2.402e+00  1.402e+00   1.713  0.08739 .  
ratTO1ma     3.081e-03  5.827e-04   5.288 1.92e-07 ***
logOR2cs    -5.344e+00  4.851e+00  -1.102  0.27124    
sqrtOR2cs    3.140e+00  2.440e+00   1.287  0.19865    
```

Michigan State:

```
        Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.120e+00  4.103e+01  -0.076   0.9394    
sqrtStl2ma  -1.742e-02  6.129e-02  -0.284   0.7763    
multAst1cs   7.575e-01  1.699e-01   4.460 1.03e-05 ***
sqrtFGA31cs  2.004e+01  1.955e+02   0.103   0.9184    
logBlk1ma   -4.880e-02  4.352e-01  -0.112   0.9108    
sqBlk1cs    -5.042e-03  1.979e-02  -0.255   0.7990    
logOR1cs    -6.305e-01  9.343e-01  -0.675   0.5001    
FTA1cs      -4.524e-02  4.982e-02  -0.908   0.3644    
sqFGA31cs   -3.414e-02  2.275e-01  -0.150   0.8808    
logFGA31cs  -2.423e+01  2.754e+02  -0.088   0.9299    
sqrtBlk1ma  -2.036e-03  4.676e-01  -0.004   0.9965    
logFTA2ma    1.536e-01  2.578e-01   0.596   0.5515    
multFTA1ma   1.811e-01  2.123e-01   0.853   0.3941    
sqFTM1ma    -1.763e-04  5.757e-04  -0.306   0.7596    
DR1ma        3.425e-03  1.109e-02   0.309   0.7576    
FGA1ma       1.131e-02  6.151e-03   1.839   0.0665 .  
```

Nevada:

```
        Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.105e+01  1.771e+01   1.188  0.23536    
ratDR1ma    -5.034e-05  2.363e-04  -0.213  0.83141    
sqStl2cs     5.158e-02  6.857e-02   0.752  0.45232    
logScore1cs -1.333e+01  5.194e+00  -2.566  0.01060 *  
sqrtFGA1cs   7.324e-01  4.513e-01   1.623  0.10531    
FGM1ma       2.631e-02  9.050e-03   2.907  0.00383 ** 
sqrtStl2cs   1.102e+01  9.912e+00   1.112  0.26675    
Stl2cs      -2.869e+00  2.836e+00  -1.012  0.31220    
ratFTM1cs   -2.736e-03  9.718e-04  -2.816  0.00508 ** 
OR2cs        3.954e-03  1.968e-02   0.201  0.84086    
logFTM1cs    4.473e+00  1.094e+00   4.091 5.11e-05 ***
sqrtBlk1ma   5.078e-01  2.784e-01   1.824  0.06882 .  
FGM1cs       2.509e-01  1.379e-01   1.819  0.06951 .  
sqBlk1ma    -9.212e-03  6.929e-03  -1.330  0.18436    
sqrtFTA2ma  -3.013e-02  5.328e-02  -0.566  0.57201    
OR2ma       -6.743e-03  1.057e-02  -0.638  0.52371    
```

North Carolina:

```
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -1.051e+03  4.568e+02  -2.301   0.0218 *
logStl2ma   -2.016e-01  1.109e-01  -1.818   0.0698 .
sqrtPF2ma    1.062e+00  1.441e+00   0.737   0.4614  
logScore2ma -3.544e-01  2.067e-01  -1.715   0.0870 .
sqFGM2cs     1.713e-03  2.625e-03   0.653   0.5143  
logTO2ma     5.247e-01  5.818e-01   0.902   0.3676  
sqFGA2cs     1.098e-01  4.663e-02   2.355   0.0189 *
FGA2cs      -3.738e+01  1.600e+01  -2.336   0.0199 *
ratTO1ma    -1.327e-03  1.617e-03  -0.821   0.4123  
ratStl1ma    1.276e-03  1.510e-03   0.846   0.3983  
Score1cs     1.617e-02  6.917e-03   2.338   0.0198 *
logPF2ma    -2.140e+00  3.040e+00  -0.704   0.4819  
PF1ma        5.578e-03  8.878e-03   0.628   0.5301  
sqrtFGM2cs  -1.619e+00  1.385e+00  -1.169   0.2431  
sqrtFGA2cs   3.753e+02  1.613e+02   2.327   0.0204 *
multTO1ma    1.488e-01  2.806e-01   0.530   0.5962  
```

Purdue:

```
              Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -3.4713184 12.6259199  -0.275  0.78349   
multFTM1ma    0.2228893  0.0781163   2.853  0.00452 **
sqrtScore1cs  0.8693863  2.3458348   0.371  0.71110   
sqrtFTA1ma   -0.1184437  0.0758785  -1.561  0.11923   
sqrtFGM2ma    0.1521799  1.5745549   0.097  0.92305   
Stl1cs        0.1241166  0.1070993   1.159  0.24711   
sqrtFTM1cs   -0.7030423  0.9042747  -0.777  0.43729   
PF1ma        -0.0594312  0.1444324  -0.411  0.68091   
sqFGM1cs     -0.0010159  0.0068014  -0.149  0.88133   
sqPF1ma       0.0019967  0.0039344   0.507  0.61205   
sqrtStl1ma    0.0867230  0.0906297   0.957  0.33913   
FGM31ma       0.0432452  0.0158680   2.725  0.00667 **
FGM2ma       -0.0343154  0.1590658  -0.216  0.82929   
sqPF1cs       0.0007669  0.0019614   0.391  0.69598   
logBlk2cs    -0.1902090  0.0878800  -2.164  0.03095 * 
FGA32cs      -0.0203165  0.0099688  -2.038  0.04213 * 
```

Tennessee:

```
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -2.171e+01  2.354e+01  -0.922   0.3569    
logFTA2cs    -2.080e+00  1.242e+00  -1.675   0.0946 .  
sqrtFGA32ma  -1.128e+00  7.020e-01  -1.607   0.1088    
sqrtScore1cs  6.121e-01  5.372e-01   1.139   0.2552    
Ast1ma       -1.457e-04  9.838e-03  -0.015   0.9882    
logFGA31cs    1.265e+00  7.207e-01   1.755   0.0799 .  
sqrtFTA1ma    2.785e-01  2.345e-01   1.188   0.2356    
sqFTA1ma     -6.175e-04  5.871e-04  -1.052   0.2934    
logFGA32ma    2.627e+00  1.521e+00   1.728   0.0848 .  
FTM2ma       -5.142e-02  4.286e-02  -1.200   0.2309    
ratFGM1cs    -2.849e-03  6.526e-04  -4.366 1.57e-05 ***
multFTA1cs   -1.873e+00  1.178e+00  -1.589   0.1127    
Ast1cs        3.747e-02  3.570e-02   1.050   0.2945    
sqrtOR1cs     7.221e+00  9.281e+00   0.778   0.4370    
sqOR1cs      -4.812e-02  5.514e-02  -0.873   0.3833    
logFTM2ma     6.827e-01  6.025e-01   1.133   0.2578    
```

Texas Tech:

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.600606  25.047640   1.980 0.048277 *
FTA2cs       -0.005087   0.014965  -0.340 0.734081
Stl2ma       -0.024513   0.013161  -1.863 0.063164 .
sqStl2cs     -0.001119   0.002099  -0.533 0.594161
sqrtFGA31cs   0.018367   0.375884   0.049 0.961050
logPF2ma      6.537602   3.597485   1.817 0.069833 .
sqrtPF2ma    -2.965789   1.667049  -1.779 0.075897 .
Score2cs     -0.041122   0.007196  -5.714    2e-08 ***
DR2ma         0.003655   0.007671   0.476 0.634004
FTA1cs       -0.063102   0.057956  -1.089 0.276824
logDR1cs    -20.311146  23.258787  -0.873 0.382978
sqrtDR1cs     8.518887  10.374189   0.821 0.411984
TO1cs        -0.129426   0.099296  -1.303 0.193087
logPF1cs      0.699927   1.079115   0.649 0.516916
logPF2cs    -11.925671   3.526571  -3.382 0.000783 ***
sqPF2cs       0.018399   0.005015   3.669 0.000272 ***
```

Villanova:

```
        Estimate Std. Error t value Pr(>|t|)
(Intercept)   48.940238 827.425995   0.059 0.952860
FTM2cs        -0.022521   0.041624  -0.541 0.588736
logFGA1cs     -1.795465   5.198508  -0.345 0.729967
sqrtFGM31cs  -86.887864  27.539301  -3.155 0.001711 **
sqrtFGM1cs     4.504492   2.909627   1.548 0.122282
logScore1cs   38.469142 376.655805   0.102 0.918696
DR1cs         -0.175013   0.119591  -1.463 0.144038
sqrtFTA1cs     1.199401   7.914126   0.152 0.879607
sqFTA1cs       0.002097   0.017555   0.119 0.904970
sqrtFTA2cs     0.219740   0.320054   0.687 0.492701
sqrtBlk1cs     0.744851   0.649592   1.147 0.252128
DR2cs         -0.066947   0.017357  -3.857 0.000131 ***
sqrtScore1cs -12.610631  87.251140  -0.145 0.885144
FGM31cs       16.231975   5.088688   3.190 0.001522 **
sqrtOR1cs     -2.010114   1.032692  -1.946 0.052210 .
ratOR1cs      -0.001124   0.001349  -0.833 0.405039
```

Virginia:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.786e+01  1.710e+01   2.798 0.005369 **
sqrtTO2ma    1.134e-01  7.333e-02   1.546 0.122781
multPF1cs   -1.778e-01  3.308e-01  -0.537 0.591232
Stl1cs       3.075e-02  1.417e-01   0.217 0.828350
logStl2ma   -5.530e-02  9.497e-02  -0.582 0.560630
sqrtDR2cs   -2.630e-01  1.624e-01  -1.619 0.106204
sqAst1cs     4.450e-02  3.439e-02   1.294 0.196375
sqrtAst1cs  -9.131e+00  7.090e+00  -1.288 0.198474
sqrtFGA1cs  -2.975e+00  7.280e-01  -4.087 5.19e-05 ***
FTA1cs       1.051e+00  7.201e-01   1.460 0.145007
sqFTA1cs    -2.087e-02  1.616e-02  -1.292 0.197173
sqrtStl1ma   6.632e-04  8.723e-02   0.008 0.993937
FGM1ma       4.595e-02  1.297e-02   3.544 0.000435 ***
sqStl2cs    -9.948e-04  1.429e-03  -0.696 0.486816
ratScore1ma -2.044e-04  3.985e-05  -5.130 4.34e-07 ***
logDR1cs    -3.505e+00  2.007e+00  -1.747 0.081397 .
```