Using GLMs to Predict Basketball Games

Joe Despres & Sabrina Ball

Michigan State University

Using GLMs to Predict Basketball Games

Introduction

Every year qualifying Division I basketball teams compete in the annual March Madness tournament. Casual fans and enthusiasts submit predictions gambling small sums of money in hopes of completing a perfect bracket. A perfect bracket is when one correctly predicts the outcome of all 63 games. This study uses logistic, poisson, and multinomial regression models fitted with R (R Core Team, 2020) to predict the outcomes of the March Madness tournament games. Our primary objective is to determine which of these GLMs make the most accurate predictions. To begin, we collect data from three different sources Kaggle (Kaggle, 2021), NCAA (NCAA, 2021), and the tournament results (NCAA, 2021). Kaggle (Kaggle, 2021), provides a comprehensive dataset including all NCAA in-season basketball games from 2001 to 2020. The NCAA (NCAA, 2021) provides team-level statistics for each team. We filter, clean, and combine these data using the tidyverse package (Wickham et al., 2019). Then use the combination of these datasets to fit our models. The objective is to predict the individual game outcomes as accurately as possible and determine which GLM is the most accurate.

Our response terms is the outcome of the game, *win or loss*, without the possibility of a tie. Our aim is to derive a function that will accurately predict this using team-level statistics. In our dataset, we have many predictors however, we only selected the predictors that are not co-linear. First, *field goal percentage* which is the ratio of attempted scores to made scores. *Free-throw percentage*, is the the rate of successfully scoring a penalty shot. Cumulative *Three-point goals made*, field goals made from a sufficient distance. *Rebounds per game*, counts the amount of times a team recovers the ball after a missed shot. *Steals* is when a team was able to remove possession of the ball from the opposing team. *Turnover*, the number of times the team lost possession. *Blocks*, the number of times the team was able to block a shot made by the opposing team.

March Madness is a winner-take-all tournament and teams do not have a second chance to play a game. Therefore, making accurate predictions will be dependent on predicting the previous round correctly. We did make predictions using that method, however our models will be compared by treating the games as independent of the previous round. Meaning, we will filter our data down to include only the 64 teams that qualify, predict every single possible combination of games $\binom{64}{2} = 2016$, then use the 63 games played as a sample from the population of all possible games. Then use basic statistical methods to determine if the predictions were better than random chance, better than betting markets, and better than seeds. Then we use the results to determine which regression method performs the best.

We will use three regression methods, logistic, poisson, and multinomial, to generate prediction models, compare the results, and make a determination as to which is best suited to this problem. Logistic regression most naturally suits this problem because games cannot tie and there is a perfect 50/50 split of wins and losses in our training and testing data. This problem could also be suited to poison regression because the number of points scored is roughly poisson distributed. After fitting a poisson model, we will predict the number of points scored by Team A, predict the number of points scored by Team B, and take the team with the highest predicted score to be the winner. A multinomial model is less naturally suited to this problem, however it may address the shortcomings of the logistic model. In particular, many games are close scoring and are won by merely a few points. Therefore, we will use multinomeal regression to predict the difference in game score by assigning the differences into five categories. From there, assign a predicted winner based on which outcome the model considers to be the most likely.

These models predict the correct outcomes with an accuracy between 59% and 65%. GLMs are accurate relative to randomly guessing March Madness tournament outcomes. Team-level statistics, recorded by the NCAA, are highly significant and helpful predicting the tournament outcomes but do not yield fantastic results. Also, when predicting the probability of a win or a loss using a logit link, the coefficients are symmetric because basketball games are a zero-sum-contest. From our results, we see that logistic and poisson models are the best at predicting wins and losses. All three GLM models are well over 50% accurate, therefore, we claim that using GLM models are more accurate than random chance.

Exploratory Data Analysis

Table 1

Descriptive Statistics

Variable	Mean	Median	Std	Min	Max	Range
Three-Points goals Scored	242.08	241.00	48.18	107.0	454.00	347.00
Field Goals Scored Percentage	44.03	44.10	2.46	35.4	52.60	17.20
Rebounds Per Game	35.48	35.35	2.49	27.6	43.81	16.21
Steels Per Game	206.34	202.00	40.59	116.0	369.00	253.00
Turn Overs	421.84	419.00	49.01	290.0	604.00	314.00
Blocks Per game	3.30	3.20	0.94	1.3	6.80	5.50

As mentioned above, there are substantially more variables in the dataset than we used, however we omitted the ones that were co-linear. When making a decision on which to keep, we selected the variable using deviance and likelihood ratio tests. For our modeling and prediction we are going to use the statistics associated with both teams. Basic descriptive statistics can be found in Table 1. We notice that the mean of the predictor is relatively close to the median. The standard deviations can be high, but not substantially high, and the ranges are reasonable for the data we have. We remark on the predictors being roughly normally distributed with no substantial skew, heavy tails, or slow

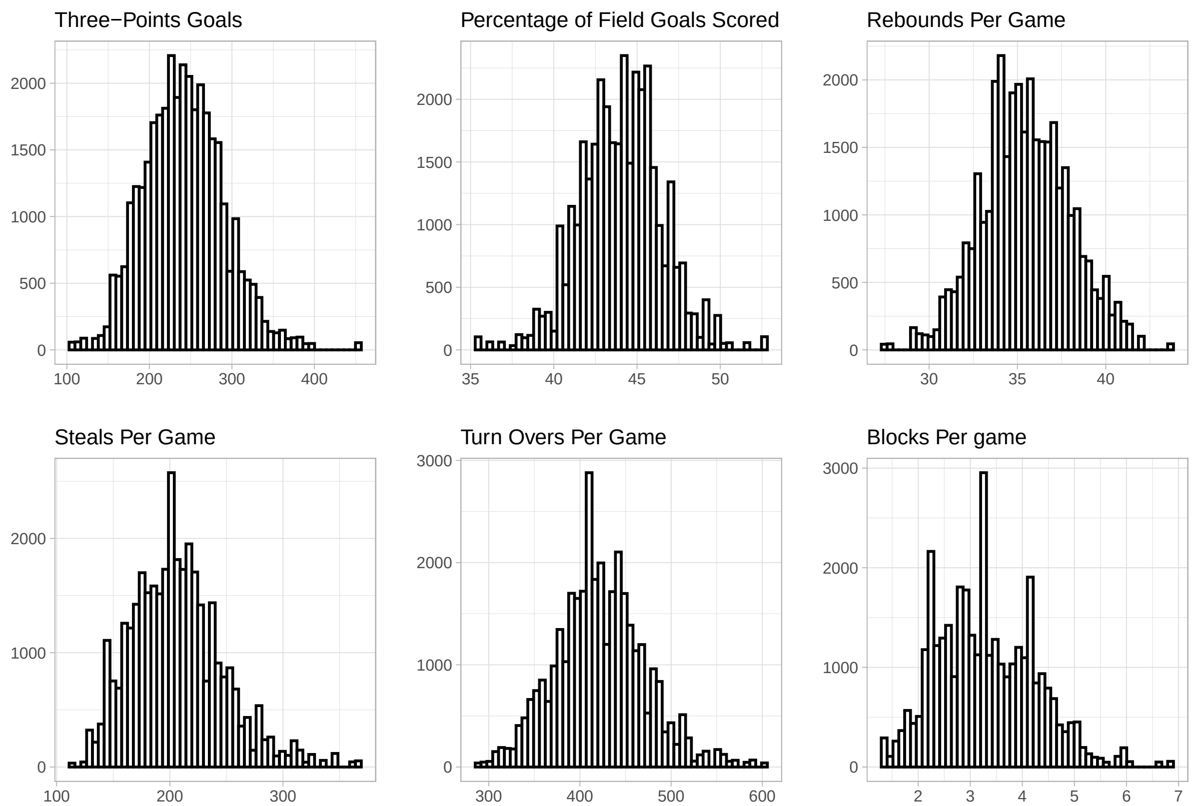


Figure 1

decay as shown in Figure 1.

Now we turn our attention to the outcomes. The logistic model is straight forward, only needing a win to be coded as 1 and a loss to be coded as 0. We fit a poisson model by taking a score as a count. This is not perfectly suited to poisson because a winning team's score has a mean of 77.13 and a variance of 113.22. Regardless we will see how it performs by predicting the score for Team A and Team B, then choose the one with the highest predicted score as the winner.

The multinomeal case is not obviously suited to this problem, however, we can adapt it. Commonly seen in betting markets, we will take a look at the difference in score between winner and loser. Then assign it into five categories using the quintiles as shown in Figure 2. First category, is the probability of the team losing by more than twelve points. Second, the probability of the team losing between twelve and four points. Third, a

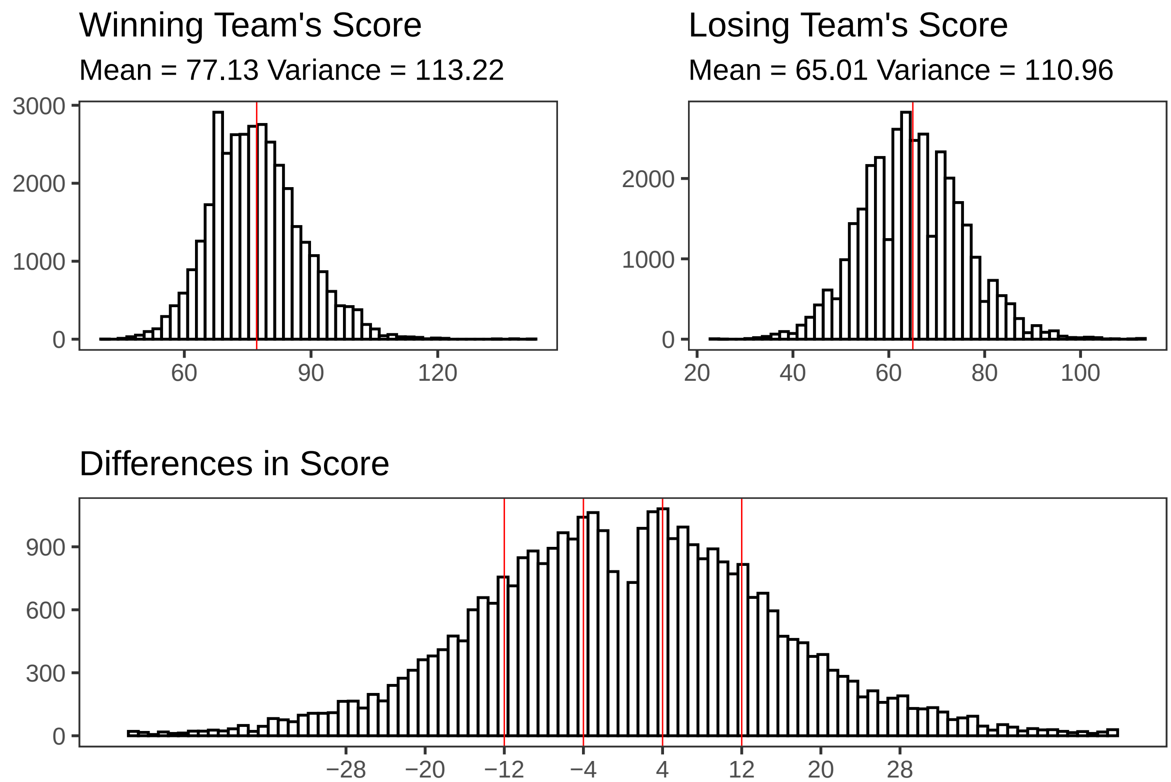


Figure 2

very close game either losing by four or less or winning by four or less (this is fairly common as there is fierce competition between the teams). Fourth, winning by more than four and less than twelve points. In the fifth category, the probability of winning by more than twelve points. We will use the probabilities to predict a winner by comparing the sum of the predicted probabilities of being in the first and second quintiles to being in the fourth and fifth quintiles.

Description

To make an accurate comparison, we use the same formula for the three models with only the dependent variable being different. Win or loss for our logistic model, the amount Team A scores for the poisson model, and the difference in score for the multinomeal model. We tried normalizing the data by subtracting the mean and dividing by the standard deviation for all predictors. However, that really did not have an effect on the prediction accuracy. Also, we individually tested each predictor using the likelihood ratio test before adding each term. When we were adding terms, we found many to be co-linear. When we found co-linear predictors, we omitted the one with a smaller likelihood ratio statistic.

$$\begin{aligned}\beta_0 + \beta_1(\text{x3fg}) + \beta_2(\text{opposingx3fg}) + \beta_3(\text{fg\_percent}) + \beta_4(\text{opposingfg\_percent}) +\\ \beta_5(\text{ft\_percent}) + \beta_6(\text{opposingft\_percent}) + \beta_7(\text{rpg}) + \beta_8(\text{opposingrpg}) +\\ \beta_9(\text{st}) + \beta_{10}(\text{opposingst}) + \beta_{11}(\text{to}) + \beta_{12}(\text{opposingto}) +\\ \beta_{13}(\text{opposingbkpg}) + \beta_{14}(\text{bkpg})\end{aligned}$$

Results

In Table 2, made manually with the kableExtra package (Zhu, 2020), you can see that we have the results for the poisson, logistic, and multinomial models. We have a sample of over 35,000 and all of the coefficients are highly significant. Which is to be expected because these are the statistics that the NCAA collects as the metrics useful in measuring a team's ability to win games. The aim of this study is to compare predictive models so we will not cover it exhaustively or include individual z-statistics and p-values. First notice, in the case of the logistic and multinomial models we see that when comparing factors that affect a team's probability of winning, the associated coefficient is nearly equal to the factor capturing the opposing teams metric. That is because basketball is a zero-sum-game, anything good for team A is proportionately bad for team B. Note this is not true for the poisson model because that is measuring points scored. Take three pointers in the poisson model, for instance, where an opposing team scores a lot of three pointers has a significant coefficient for the amount of points scored. More points scored is not deterministic of winning, however, it is an indicator.

Table 2

Regression Output

Terms	Multinomial Model	Logistic Model	Poisson Model

<-12	-1.2516	–	–
-12:-4	-0.28546	–	–
-4:4	0.54181	–	–
4:13	1.54645	–	–
Three Pointers	-0.00061	0.00243	4e-04
O* Three Pointers	0.00067	-0.00255	0.00013
Field Goals	-0.07605	0.18322	0.01421
O* Field Goals	0.0752	-0.17632	-0.0052
Free-throws	-0.01624	0.03242	0.00424
O* Free-throws	0.01418	-0.03059	0.00078
Rebounds	-0.06812	0.14321	0.01286
O* Rebounds	0.06823	-0.14476	-0.00234
Steals	-0.00287	0.00655	0.00039
O* Steals	0.00276	-0.0064	-0.00021
Turnovers	0.00234	-0.00603	-0.00021
O* Turnovers	-0.00251	0.0058	0.00045
Blocks	0.06195	-0.14284	-0.01497
O* Blocks	-0.05622	0.16488	0.00263

[1] Sample size 35248 games.

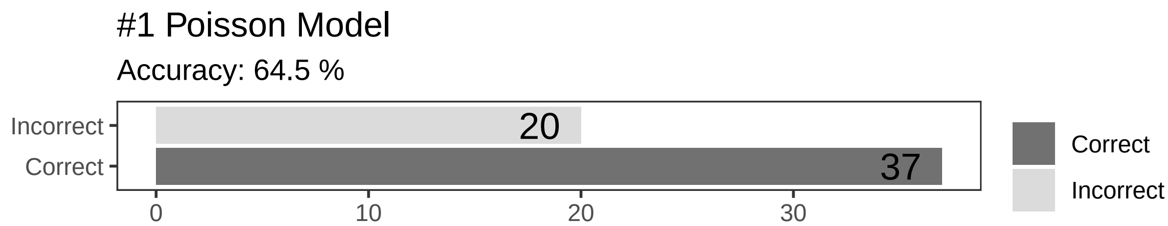
[2] O* is the term asscoiated with the opposing team.

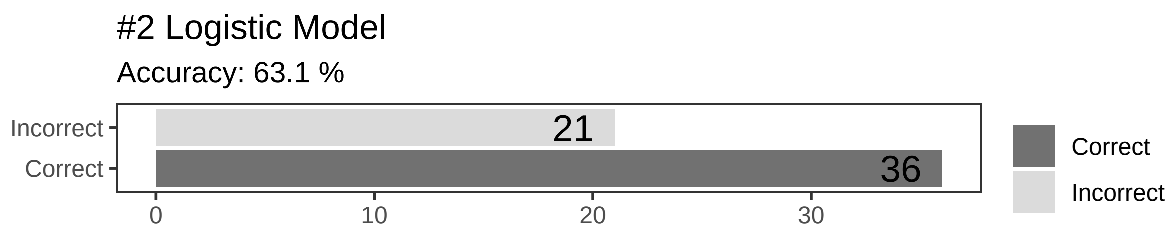
[3] All the above terms are significant at $P < 0.01$.

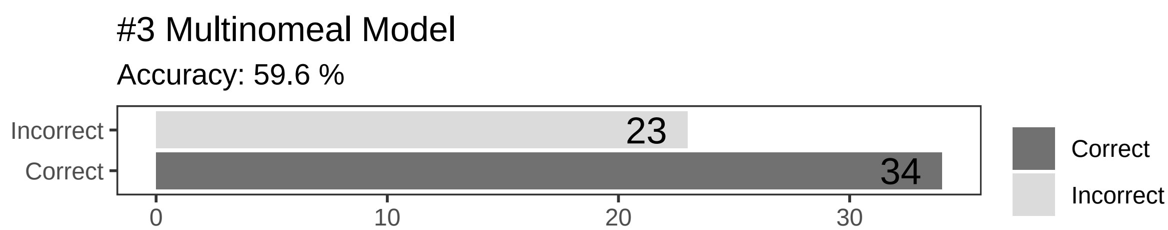
[4] Multinomeal Intercepts are differences in predicted score.

Goodness of Fit

Now that we have fitted the model, lets take a look at how well it performed. Using the tidyr [@-tidyr] package, we make every combination of teams and associated statistics in one dataframe. Then wrote custom functions that would take two of the 64 teams as inputs and output the probabilities of winning, predicted amounts of points scored, or probabilities of the score resulting in one of the above mentioned multinomeal categories. After that we used the purrr (Henry & Wickham, 2020) package to iterate over all possible games that could be played. From there, wrote and ran a web scraping script to obtain a dataframe of the results. Then counted the games each model predicted correctly and incorrectly. In this case our population is all $\binom{64}{2}$ possible games in this tournament and a sample is the 63 games that were played. We are not assuming that we have a random sample from the whole population as these are the best teams in the league. Also, there is a selection bias towards games that were played. Therefore, the teams that played in the finals are represented 6 times in our sample and 32 of the 64 teams only are represented once. Therefore, we do not claim that these results will hold for games in general. However, for the purpose of comparing models that make predictions in this specific tournament this is appropriate.







Conculsion

From these results we see that logistic and poisson model are best at predicting wins and losses. All three models are well over 50% accurate, confirming the claim that GLMs can predict March Madness outcomes better than random chance. Also, we confirmed the suspicion that basketball statistics are symmetric in relation to the probability of winning and are not when we are predicting the game score.

The limits of this study have a lot to do with data limitations. Ideally, we would have more tournaments played with lower performing teams. Also, the results would be much more robust if the tournament was larger. We had a lot of data points. over 35,000, which is more than sufficient, however, we do not have an abundance of non co-linear covariates. An accuracy of 64.5% is not superb considering betting markets take these exact factors into account, however, being higher than random chance indicates that GLMs are useful in predicting March Madness outcomes.

Future studies could be focused on making a better fit and getting a better understanding of season games before making predictions like this. To get more robust predictions, it would be beneficial to go back through the season games data and draw out the times each team in the March Madness tournament played each other and test our model against that. While exploring, we found that teams from different parts of the country tended to have a different coefficient on statistics and outcomes. A team's conferences is likely a random effect that should be taken into account. Therefore, additional research into the random effect would certainly yield more robust predictions.

References

Henry, L., & Wickham, H. (2020). *Purrr: Functional programming tools*. Retrieved from https://CRAN.R-project.org/package=purrr

Kaggle. (2021). *March machine learning mania 2021*. Retrieved from https://www.kaggle.com/c/ncaam-march-mania-2021

NCAA. (2021). *Men's basketball*. Retrieved from https://www.ncaa.com/stats/basketball-men/d1

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Zhu, H. (2020). *KableExtra: Construct complex table with 'kable' and pipe syntax*. Retrieved from https://CRAN.R-project.org/package=kableExtra