Using GLMs to Predict Basketball Games

Joe Despres & Sabrina Ball

Michigan State University

Using GLMs to Predict Basketball Games

Introduction

Every year top division I basketball teams compete in the annual March Madness tournament. Casual fans and enthusiasts submit predictions gambling small sums of money on the outcomes. This study uses logistic, poisson, and multinominal regression models in R (R Core Team, 2020) to predict the March Madness tournament outcomes. Our primary objective is to determine which GLM is the most accurate when predicting the results. The data we use come 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 (non-turniment) 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* with a 0 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 were not co-linear. First, *field goal percentage* which is the rate attempt to score vs actually scoring. *Free-throw percentage*, is the rate that the team has the opportunity to take an unguarded shot. Cumulative *Three-point goals made. Rebounds per game*, counts the times the team recovered the ball after a missed shot. *Steals*, the times 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 depending on predicting the previous round correctly. We did make predictions in that fashion, however our models will be compared by taking independent predictions. Meaning, we will filter our data down to using 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 multinominal, to generate a prediction models, to compare the results, and make a determination as to which is best suited to this problem. Logistic regression most naturally suits this suits this problem because games do not 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 poisson distrubuted. After fitting a poisson model we will predict the amount of points Team A will score, then predict the number of points Team B will score then take whichever team was predicted to score more and record that as the prediction.

We find that these models have an accurate between 59% and 65% which is well over 50%. Therefore, GLM's are helpful in predicting March madness tournament outcomes. We find that the team-level statistics recorded by the NCAA are highly significant and helpful for predicting the tournament outcomes however, come up short of producing truely fantastic results. Also, we find that when predicting the probability of a win or a loss we find that the coefficients are symmetric because of the nature of the zero-sum-contest.

From these results we see that logistic and posisson model was the best at predicting wins and losses. All three models were well over 50% accurate, so this the research goal of having GLM's predict March Madness Outcomes. Also, we confirmed the suspicion that basketball statistics are symmetrical in relation to the probability of winning. Yet are not

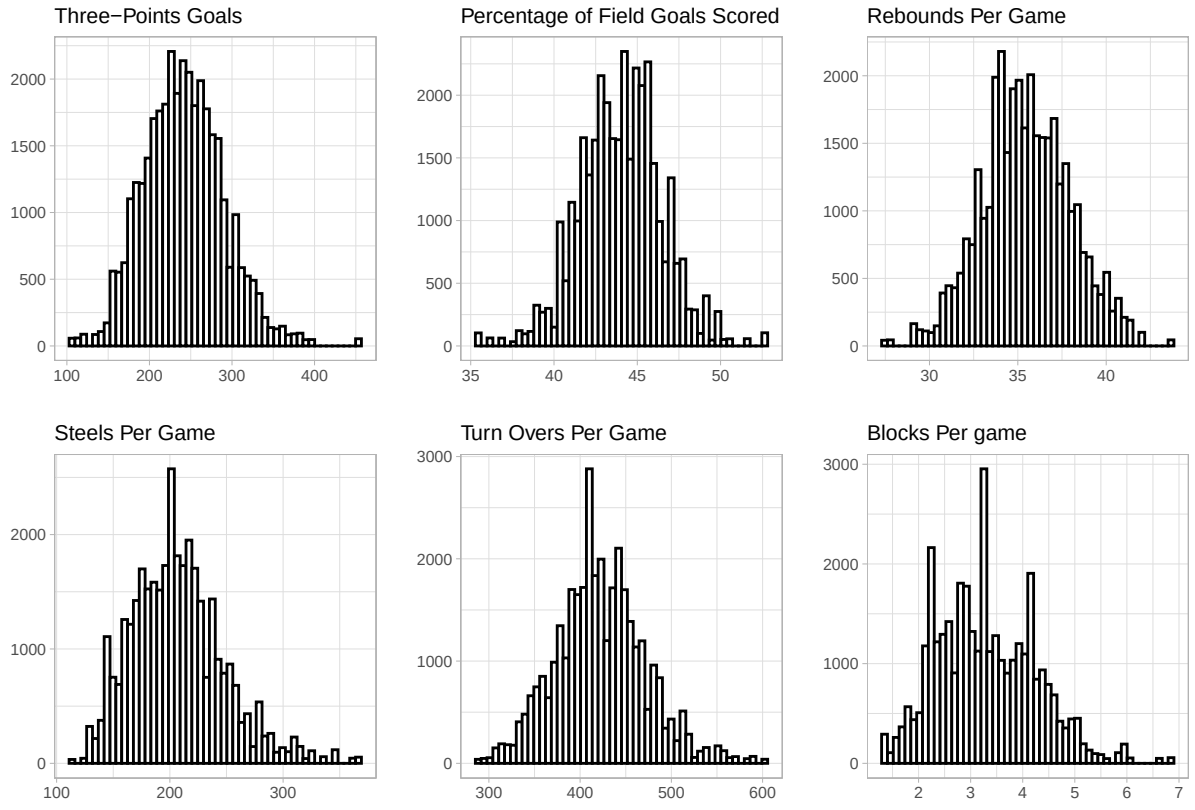
when we are predicting the score.

Exploratory Data Analysis

Table 1

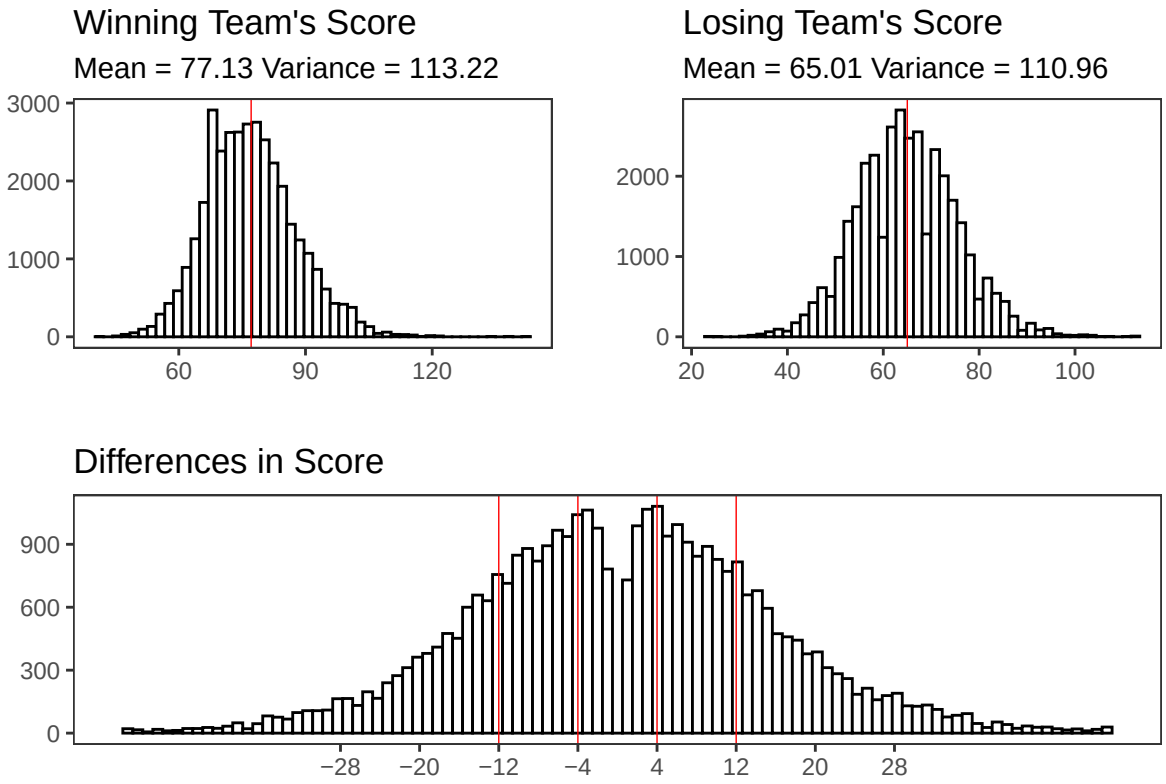
Descriptive Statistics

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 we used, however we omitted the ones that were co-linear. When making a decision we selected the variable using deviance tests and likelihood ratios. For our modeling and prediction we are going to use these 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 are not wildly so and the range is reasionable for the data we have. In the histograms below we remark the predictors roughly normal distribution with no substantial skew, heavy tails, or slow decay.

Now we turn our attention to the outcomes. The logistic model is straight forward all we needed to do was code a win to be 1 and a loss to be 0 then fit the model and make a prediction. The possion model will work if we to taking 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, when we will predict the score for Team A and predict the score for Team B, then take who ever has a higher predictions.



The multinomeal case is not obviously suited to this problem however we can adapt it. We will take a look at the difference in score between winner and loser. This is a common thing in betting markets to gamble on. Therefore we mapped the difference in score into 5 categories using the 0.2, 0.4, 0.6, and 0.8 quantifies. First, the team losing with by more than 12. Second, the team losing by between 12 and 4 points. Third, a very close game between losing by 4 and winning by 4 (This is fairly common as there is really good competition between teams). Fourth, winning by more than 4 and less than 12. In the fifth category, winning by more than 12. We will use the probabilities to predict a winner by suming the proabilities of losing by more than 12 and losing by between 12 and four and summing the probability of winning by more than 12 and winning by between four and 12. Then taking whichever is larger as a prediction.

Description

To make an accurate compartment, we use the same formula for the three models with only the dependent variable being different. Win or loss for our logistic, the amount Team A scores for the poisson and the difference in score for the multinomeal. Now we tried normalizing the data where we subtracted the mean and devided by standard deviation for all out predictors. However, that really did not have that much effect. Also we individually tested each predictor using likelyhood ratio tests to add each term. These all came significant. Also when we were adding terms we found many were co-linear, therefore when we found co-linear predictors we omitted the one with a smaller likelihood ratio.

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

Results

In the Table 2 made manually with the kableExtra package (Zhu, 2020), you can see that we have the results for poisson, logistic, and multinomeal models. We have a sample of over 35,000 and all of the coeffecients are highly significant. This is to be expected because these are the statistics that the NCAA records and ones they have determined to being useful to measure a team's performance. 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 is that in the case of the logistic and multinomeal models we see that when comparing a factors that affect a team's probability of winning we see the associated coefficient is quite similar for the factor capturing the opposing teams. That is because basketball is a zero-sum-game, and anything good for team A is proportionately bad for team B. Note this is not true of the multinomeal and poisson models. The reasion that is is because the winner picked by thoes models is a function of the score. Therefore, thoes models are looking at coeffecients that are oging to increase the scores. Take three pointers in the poisson model, for instance, a team 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.

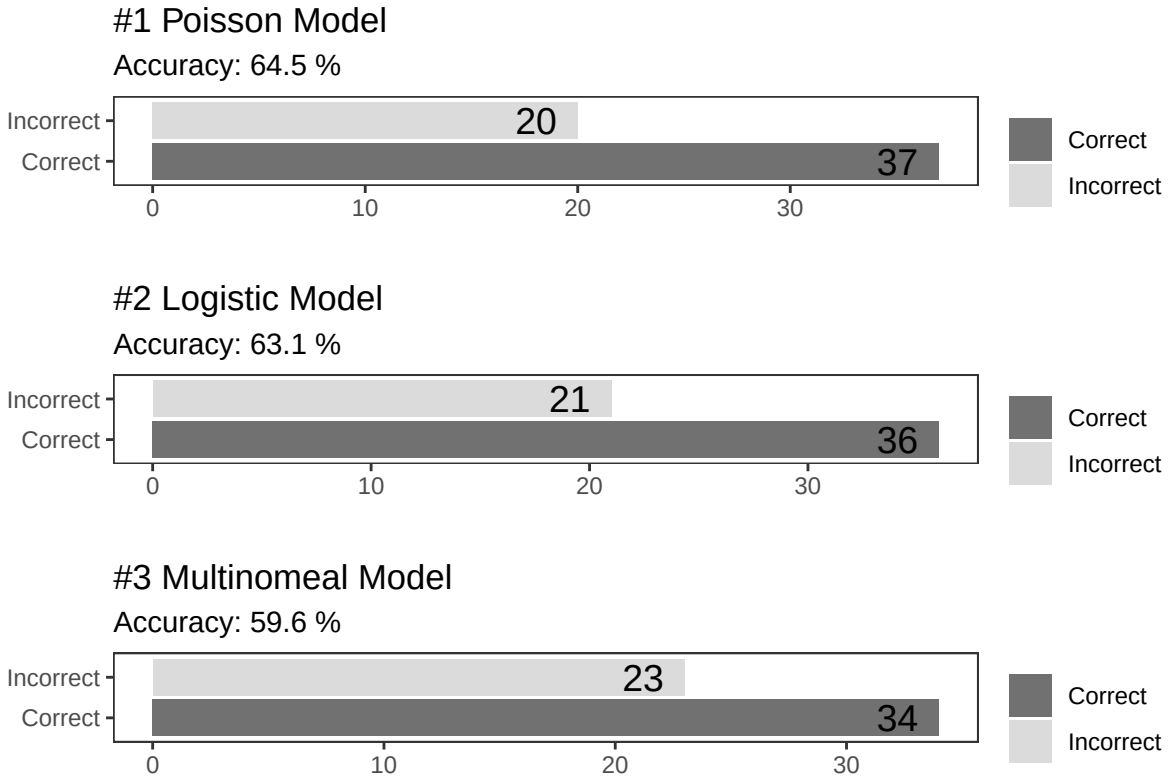
[3] All the above terms are significant at $P < 0.01$.

[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 webscraping 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 turniment and a sample of the 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 twards 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 for this specific turniment this is approprite.



Conculsion

From these results we see that logistic and posisson model was the best at predicting wins and losses. All three models were well over 50% accurate, so this the research goal of having GLM's predict March Madness Outcomes. Also, we confirmed the suspicion that basketball statistics are symmetrical in relation to the probability of winning. Yet are not when we are predicting the score.

The limits of this study have a lot to do with data limitations. Ideally, we would have more tournaments playing with lesser skilled teams. would be much more robust if the tournament was much larger. We had a lot of data points over 35,000 which I think was more than sufficient, however, we lacking non co-linear covariates. We only had 7 predictors. An accuracy of 64.5% is not superb considering betting markets take these exact factors into account.

Future studies could be focused on fitting more models and getting a better understanding of season games before making predictions like this. I would like to get more robust predictions, therefore, I think it could be beneficial to go back through the seasion games data and draw out the times each team in the March Madness tournament played each other and test our model against that. (Well see if we feel like doing that). While exploring, I found that teams from different parts of the country tended to have a different coefficients relationship between statistics and outcomes. I think the team's conferences are a random effect that should be taken into account. I think the season games could be modeled very nicely using linear mixed models.

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