Duquesne University
Duquesne Scholarship Collection

Electronic Theses and Dissertations

2012

Modeling the NCAA Tournament Through Bayesian Logistic Regression

Bryan Nelson

Recommended Citation

Nelson, B. (2012). Modeling the NCAA Tournament Through Bayesian Logistic Regression (Master's thesis, Duquesne University). Retrieved from https://dsc.duq.edu/etd/970

MODELING THE NCAA TOURNAMENT THROUGH BAYESIAN LOGISTIC REGRESSION

A Thesis

Submitted to the McAnulty College & Graduate School of Liberal Arts

Duquesne University

In partial fulfillment of the requirements for

the degree of Master of Science in Computational Mathematics

By

Bryan T. Nelson

August 2012

MODELING THE NCAA TOURNAMENT THROUGH

BAYESIAN LOGISTIC REGRESSION

By

Bryan T. Nelson

Approved June 20, 2012

Eric Ruggieri, Ph.D.
Assistant Professor of Statistics
Committee Chair

John Kern, Ph.D.
Associate Professor of Statistics
Committee Member

Stacey Levine, Ph.D.
Associate Professor of Mathematics
Committee Member

Donald Simon, Ph.D.
Chair, Department of Mathematics and
Computer Science
Associate Professor of Computer Science

James Swindal, Ph.D.
Dean, McAnulty College and Graduate
School of Liberal Arts
Associate Professor of Philosophy

ABSTRACT

MODELING THE NCAA TOURNAMENT THROUGH BAYESIAN LOGISTIC REGRESSION

By

Bryan T. Nelson

August 2012

Thesis supervised by Dr. Eric Ruggieri

Many rating systems exist that order the Division I teams in Men's College Basketball that compete in the NCAA Tournament, such as seeding teams on an S-curve, and the Pomeroy and Sagarin ratings, simplifying the process of choosing winners to a comparison of two numbers. Rather than creating a rating system, we analyze each matchup by using the difference between the teams' individual regular season statistics as the independent variables. We use an MCMC approach and logistic regression along with several model selection techniques to arrive at models for predicting the winner of each game. When given the 63 actual games in the 2012 tournament, eight of our models performed as well as Pomeroy's rating system and four did as well as Sagarin's rating system when given the 63 actual games. Not allowing the models to fix their mistakes resulted in only one model outperforming both Pomeroy and Sagarin's systems.

DEDICATION

I dedicate this thesis to my incredible family, whose everlasting love and support has never wavered throughout my entire education. Particularly, I dedicate this thesis to my parents, Scott and Kathy, for believing in me throughout everything I have sought after in life and for always being there for me when needed. I would not be as successful as I am today without their guidance. Also, I dedicate this work to my younger brother Kevin, for keeping me young at heart and for making me laugh during all the times that were difficult and stressful.

A special dedication must go out to my grandfather Andrew Sickle, who never tried to hide how proud he was of my accomplishments, and my grandmother LaVerne Sickle, who truly believed it whenever she told me that I could do anything that I wanted to do in life.

ACKNOWLEDGEMENTS

This thesis marks the end of a spectacular journey here at Duquesne University that began six years ago as a freshman and is now culminating with the completion of my Master's degree. There are several people who I must express my gratitude towards for making this thesis possible. First and foremost, I would like to thank my advisor Dr. Eric Ruggieri for taking me on as a student and guiding me through the entire process of writing this thesis from start to finish. His invaluable advice made performing the research and writing this thesis a truly enjoyable task. I could not have imagined working with someone so dedicated and willing to help at a moment's notice. The amount of knowledge I have gained from working with him is immeasurable.

I would also like to thank Dr. John Kern and Dr. Stacey Levine for taking the time out of their busy schedules to serve as committee members and critique my research. Having sat in so many of their classes over the past few years, I know that I have received a spectacular education that will help carry me through the rest of my future education and well into my career.

Finally, I would like to thank Dr. Donald Simon for approval in allowing me to take on such a fun topic to research to complete my Master's degree, and to all the other professors in the Department of Mathematics and Computer Science with whom I worked who made the past six years an unforgettable experience for me.

TABLE OF CONTENTS

Page

LIST OF TABLES

1. Introduction

Every March, the nation becomes captivated by the NCAA Men's Basketball Tournament, the 68 team single elimination tournament to decide the national champion that is also informally known as March Madness. For readers unfamiliar with the format of the tournament, we will provide a brief overview before going into detail of the mathematics behind the modeling process. The 30 teams winning their conference tournaments plus the Ivy League regular season champion all receive automatic bids into the tournament. The remaining 37 at-large slots are filled by the best teams (according to the tournament selection committee) that did not win their conference tournament. The 68 teams are then ranked from 1 through 68 on an S-curve. The S-curve is then used to seed teams from 1 through 16 in each of four brackets. The top four teams on the S-curve receive the four coveted number 1 seeds; teams ranked fifth through eight are given 2 seeds, and so on, down to placing the 16 seeds from the bottom teams on the S-curve. The bottom four conference tournament winners and the bottom four at-large teams on the S-curve play in four play-in games. These four games occur before what is typically considered the official beginning of the tournament and reduces the field to 64 teams. From this point, the 16 teams in each of the four brackets play a single elimination tournament to determine a regional champion. These teams move on to play in the Final Four. The first round games are determined by the seeds, where the 1 seed in each bracket plays the 16 seed, the 2 seed plays the 15 seed, and so on down to the 8 seeds and 9 seeds playing each other. For further rounds, the advancing teams are not reseeded. Once the four regional champions are determined, another single elimination tournament occurs between these teams in the Final Four to determine the national champion.

For the purposes of this paper, the four play-in games are not taken into consideration. The predictions will begin with games in the Round of 64. Moreover, whereas the NCAA has referred to the play-in games as the “first round” since expanding to 68 teams in 2011, for the purposes of this paper, we will refer to the Round of 64 as the first round and the Round of 32 as the second round. One would be led to believe that choosing the higher seeded team to win each game would result in relatively high accuracy. However, this is not the case. Between 2003 and 2011, choosing the higher seeded team to win in each game would have resulted in 409 correct picks out of 567 games, an accuracy of 72%. Many of these upsets have been 14 and 13 seeds upsetting 3 and 4 seeds, respectively, in the first round. More curiously, in the same time span, seven 12 seeds advanced to the Sweet Sixteen by winning two games and two 11 seeds even advanced to the Final Four after four tournament victories. These are teams that are ranked in the bottom third on the S-curve.

Other rating systems that attempt to improve upon choosing teams simply based on seeding exist. Two of the more famous include the Sagarin ratings and Pomeroy ratings. These methods use each team’s regular season statistics to create a single rating for each team. When confronted with a matchup, the team with the higher rating is favored to win. Predicting the winners in each bracket from 2003 through 2011 based solely on the above ratings increases accuracy slightly; the Sagarin ratings were 73% accurate, and the Pomeroy ratings reached an accuracy of 74%.

In this paper, we introduce another method of predicting winners of March Madness games by identifying a model that will compare the two teams playing in each of the 63 games each year on a head-to-head basis using their statistics from the regular season. This will allow for the realistic possibility of choosing an upset if the lower seeded team has a favorable matchup against the higher seeded team, despite the higher seeded team appearing to be better in all other rating systems. We will use a Markov chain Monte Carlo (MCMC) approach in identifying the model that best fits the data as well as finding the coefficients of regression. Logistic regression will be used to identify the predicted probability that the higher seeded team will win the game, and the accuracy of the prediction will be used to assess how well the model works. The goal is both to render the seeding of teams in the NCAA tournament as artificial and to show that a rating system that creates an ordering of teams based on regular season statistics can be outperformed by analyzing the individual matchup using similar statistics.

The structure of this paper is as follows. Section 2 introduces the methods and algorithms used in the model selection process. It also elaborates on the processes used to calculate the regression coefficients for those models. Section 3 begins by explaining the data collection process. It then continues by comparing how the models fare in numerous settings. Section 4 states the conclusions that can be drawn from the entire process. Finally, the last section offers some possibilities for further discussion, including problems that were encountered, other techniques that exist that were not explored in this project, and the potential for future research.

2. Methods

Given the dependent variable Y and n predictor variables $X_1, \ldots X_n$, the logistic function is given by

$$P(Y) = \frac{e^{\beta_0 + \sum_{i=1}^{n} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{i=1}^{n} \beta_i X_i}}$$

where $P(Y)$ is the predicted probability of a success given the n predictor variables $X_1, \ldots X_n$ and β_i is the regression coefficient corresponding to predictor variable X_i. In the context of the problem, $X_1, \ldots X_n$ are basketball statistics (which will be explained in more detail in Section 3.1) for the teams playing in the matchup. Assume that there are N observations in the data set. The dependent variable Y is a vector of length N where each Y_j is coded as either 0 or 1, where 0 is a failure (the lower seeded team wins) and 1 is a success (the higher seeded team wins). Each of the independent variables $X_1, \ldots X_n$ is a vector of length N as well. Define a model as some subset of $X_1, \ldots X_n$. We seek to find the model that maximizes the likelihood of the model given the data. In developing this model, the ideal scenario would be to calculate the likelihood of each of the 2^n possible models and choose the model with the maximum likelihood. However, when n becomes large, this process becomes impossible to carry out efficiently. Instead, we use the Metropolis-Hastings algorithm, a Markov chain Monte Carlo (MCMC) approach, to select the model.

2.1 Likelihood of a Model

We begin with an explanation of the likelihood of a model. Given the data, the likelihood function of a model $M_0 = \{X_1, X_2, \ldots, X_k\}$ is a measure of how well the model fits the data, where k is the size of the model with $k < n$. The likelihood of a model is the product of three separate factors:

1. The likelihood function for the model with variables $X_1, X_2, \dots, X_k$ and the set of regression coefficients $\beta_0, \beta_1, \dots, \beta_k$.
2. The joint prior distribution $\pi(\beta_0, \beta_1, \dots, \beta_k | M_0)$ for the regression coefficients $\beta_0, \beta_1, \dots, \beta_k$.
3. The prior distribution $\pi(M_0)$ on the model itself.

The likelihood function is given by

$$P(Y|M_0, \beta_0, \beta_1, \dots, \beta_k) = \prod_{j=1}^{N} \left(\frac{e^{\beta_0 + \Sigma_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \Sigma_{X_i \in M_0} \beta_i X_i}} \right)^{Y_j} \left(1 - \frac{e^{\beta_0 + \Sigma_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \Sigma_{X_i \in M_0} \beta_i X_i}} \right)^{1 - Y_j}.$$

Assume for each β_i that $\beta_i \sim N(\mu_i, \sigma_i^2)$. Then the prior distribution for each β_i is

$$\pi(\beta_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{\frac{-(\beta_i - \mu_i)^2}{2\sigma_i^2}}.$$

Furthermore, for the purposes of this paper, assume that $\mu = \mu_0 = \mu_1 = \cdots = \mu_n$ and $\sigma^2 = \sigma_0^2 = \sigma_1^2 = \cdots = \sigma_n^2$. Then the prior distribution for each β_i is

$$\pi(\beta_i) = \frac{1}{\sigma \sqrt{2\pi}} e^{\frac{-(\beta_i - \mu)^2}{2\sigma^2}}.$$

Denote the joint prior distribution of $\beta_0, \beta_1, \dots, \beta_k$ by $\pi(\beta_0, \beta_1, \dots, \beta_k | M_0)$. Then

$$\pi(\beta_0, \beta_1, \dots, \beta_k | M_0) = \left(\frac{1}{\sigma \sqrt{2\pi}} \right)^{k+1} e^{\Sigma_{i=0}^{k} \frac{-(\beta_i - \mu)^2}{2\sigma^2}}.$$

Assume that the prior distribution on the model $\pi(M_0)$ is uniform. This renders all models equally likely, and returns a prior distribution on the model of

$$\pi(M_0) = \frac{1}{2^n}.$$

Since the marginal posterior distribution is proportional to the product of the likelihood function and the prior distributions, we must integrate out $\beta_0, \beta_1, \dots, \beta_k$ to find the normalization constant. Integrating out the regression coefficients also leaves us with just the likelihood of the model given the data. Thus, the marginal posterior for M_0 is

$$P(M_0|Y) = \int \dots \int P(Y|M_0, \beta_0, \beta_1, \dots, \beta_k)\ \pi(\beta_0, \beta_1, \dots, \beta_k|M_0)\pi(M_0) d\beta_0 \dots d\beta_k.$$

However, the above integral does not have a closed form. Instead, Monte Carlo integration is used to approximate the integral. We can approximate the integral using a uniform prior to approximate the normal prior by integrating each β_i over the interval $[\mu - c, \mu + c]$. Since there are k variables and one constant term, each assumed to have the same prior distribution, there are $k + 1$ uniform priors in the approximation. The marginal posterior for M_0 can then be approximated by

$$P(M_0|Y) \approx \int \dots \int P(Y|M_0, \beta_0, \beta_1, \dots, \beta_k)\ (2c)^{k+1} \left(\frac{1}{2^n}\right) d\beta_0 \dots d\beta_k.$$

As none of the regression coefficients appear in the uniform prior for the regression coefficients or the uniform prior for the model, these priors can be brought outside the integral, resulting in an approximation of the likelihood of

$$P(M_0|Y) \approx (2c)^{k+1}\left(\frac{1}{2^n}\right)\int \dots \int P(Y|M_0, \beta_0, \beta_1, \dots, \beta_k)\, d\beta_0 \dots d\beta_k = (2c)^{k+1}\left(\frac{1}{2^n}\right) f_{ave}.$$

Recall from the uniform prior that we assume each β_i may come from $[\mu - c, \mu + c]$. Let A be a $(k+1) \times r$ matrix where r is the number of random samples we wish to generate to approximate the above integral. Note that r must be large in order to approximate the above integral well enough to be considered approximately equal to the true likelihood. Let A_s denote the s^{th} column of matrix A. Uniformly sample $(k+1)r$ random variables from $[\mu - c, \mu + c]$ and place them in A. Through Monte Carlo integration, it follows that

$$f_{ave} \approx \left(\frac{1}{2\text{c}}\right)^{\text{k+1}} \frac{1}{r}\sum_{s=1}^{r} P(Y|M_0, A_s)$$

where $P(Y|M_0, A_s)$ is the likelihood of model M_0 using the uniformly sampled set of regression coefficients from A_s. The reader interested in the specifics behind Monte Carlo integration should consult [1].

Multiplying the above approximation by the k uniform priors, the likelihood of the model is well approximated by

$$P(M_0|Y) = (2c)^{k+1}\left(\frac{1}{2^n}\right)\frac{1}{r}\sum_{s=1}^{r} P(Y|M_0, A_s).$$

2.2 Bayesian Model Selection

One common way to develop a model for a set of data is through Bayesian model selection. Section 2.2.1 will describe the Metropolis-Hastings algorithm that is used to build the most likely model based on the likelihood from Section 2.1. Section 2.2.2 will then explain how the regression coefficients are calculated for a given model using Metropolis sampling.

2.2.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm [1] is a process by which variable selection can be performed. The goal is to maximize the likelihood of the model by proposing and accepting a new model and moving to a new state with some probability using the likelihood ratio test. Let $M = \{M_1, M_2, \dots\}$ be the ordered set that contains all models evaluated during the course of the algorithm where $M_t \in M$ is a set containing the variables included in the model at iteration t.

The algorithm contains seven steps:

1. **Initialization**: Begin by letting $M_1 = \{\}$ so that the logistic function is $P(Y) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$. Note that the model will return the same predicted probability for each observation in the dataset since there are no variables, and all predictions are being made solely on the estimation of the intercept.
2. **Calculate the Likelihood of $\boldsymbol{M_t}$**: Use Monte Carlo integration as described in Section 2.1 to approximate the likelihood of M_t. Denote this likelihood by L_t.
3. **Propose a New Model to Compare Against $\boldsymbol{M_t}$**: Generate a random integer u between 1 and n. If $X_u \notin M_t$, then set $M_{t+1} = M_t \cup \{X_u\}$. If $X_u \in M_t$, then set $M_{t+1} = M_t - \{X_u\}$.
4. **Calculate the Likelihood of $\boldsymbol{M_{t+1}}$:** As in Step 2, use Monte Carlo integration to approximate the likelihood of model M_{t+1}. Let this likelihood be denoted by

L_{t+1}. Note that the proposed model differs from the current model by only a single variable.

5. **Likelihood Ratio Test**: Form the following ratio: $L = \frac{L_{t+1}}{L_t}$. Let the probability of accepting the new model be $P = \min\{1, L\}$.
6. **Changing States**: Generate a random uniform number v on the interval $[0,1]$. If $v \leq P$, then change states and accept M_{t+1} as the current state. If $v > P$, then set $M_{t+1} = M_t$, and continue using M_t as the current state.
7. **Update**: Increment t by 1, and repeat steps 2 through 6 as necessary.

Observe that if $L_{t+1} > L_t$, then model M_{t+1} is more likely to represent the data than model M_t. Moreover, since $L_{t+1} > L_t$, it follows that $L > 1$. Thus, $P = 1$, so no matter what value of v is sampled from $[0,1]$, we can guarantee that we change states so that we accept M_{t+1} as the new "current state". However, if $L_{t+1} < L_t$, then $L = P < 1$. This implies that model M_{t+1} is less likely than model M_t, but the existence of a nonzero probability P allows us to change states. This occurs to prevent the algorithm from getting permanently stuck in a local maximum. Even for very small values of P, the algorithm will eventually allow for a change of states in search of the true maximum likelihood.

One downfall to the Metropolis Hastings algorithm is the inability to recognize that the true maximum likelihood has been reached. Since the algorithm does allow for a change of states with some small probability, it is possible that it may move away from the true maximum and gravitate towards a local maximum. The interested reader can refer to [2] for more information on the Metropolis-Hastings algorithm.

2.2.2 Metropolis Sampling

Let M_T be the model for which we need to find regression coefficients for the included variables. We can write the model as follows:

$$P(Y) = \frac{e^{\beta_0 + \sum_{X_i \in M_T} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_T} \beta_i X_i}}$$

where variable X_i is included in the model only if it is also in the set M_T. Assume M_T contains k unique variables. Denote the variables contained in M_T by $X_1, X_2, \dots, X_k$. Note that here, the subscript on the variable does not correspond to the subscript on the variable from Section 2.2.1; instead, we are simply putting an ordering on the k variables that are included in this particular model. We will now solve for the β_i that corresponds to each above X_i using Metropolis sampling.

Since the joint posterior distribution for $\beta_0, \beta_1, \dots, \beta_n$ is proportional to the product of the likelihood function, the prior distributions for each β_i, and the prior distribution on the model, the joint posterior is given by:

$\pi(\beta_0, \beta_1, \dots, \beta_k | Y, M_T) \propto$

$$\prod_{j=1}^{N} \left(\frac{e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right)^{Y_j} \left(1 - \frac{e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right)^{1 - Y_j} \left(\frac{1}{\sigma \sqrt{2\pi}} \right)^{k+1} e^{\sum_{i=0}^{k} \frac{-(\beta_i - \mu)^2}{2\sigma^2}} \left(\frac{1}{2^n} \right).$$

The algorithm to calculate $\beta_0, \beta_1, \dots, \beta_n$ contains seven steps:

1. **Initialization**: Let $\widehat{\beta_0}$ be a vector of length $k + 1$ so that $\widehat{\beta_0} = \langle \mu_0, \mu_1, \dots, \mu_k \rangle$. Then $\widehat{\beta_0}$ is the initial guess for the true values of $\beta_0, \beta_1, \dots, \beta_k$.

2. **Evaluate the Joint Posterior**: Plug the values in $\widehat{\beta_0}$ into the joint posterior to calculate the initial joint posterior. Denote this number by θ_0.
3. **Change a Single Beta**: Randomly sample another guess for μ_0 from the distribution $N(\mu_0, \sigma^2)$ and call it $\widehat{\mu_0}$. Create a new vector $\widehat{\beta_1} = \langle \widehat{\mu_0}, \mu_1, \ldots, \mu_k \rangle$. Evaluate the joint posterior above using $\widehat{\beta_1}$, and denote this number by θ_1.
4. **Likelihood Ratio Test**: Form the following ratio: $\theta = \frac{\theta_1}{\theta_0}$. Let the probability of accepting $\widehat{\beta_1}$ as the new guess for the true values of $\beta_0, \beta_1, \ldots, \beta_k$ be $P = \min\{1, \theta\}$.
5. **Acceptance/Rejection of Beta**: Generate a random uniform number v on the interval $[0,1]$. If $v \leq P$, then change states and accept $\widehat{\beta_1}$ as the current set of regression coefficients. If $v > P$, then set $\widehat{\beta_1} = \widehat{\beta_0}$, and continue using $\widehat{\beta_0}$ as the current set.
6. Repeat steps 2 through 5 k additional times, once for each regression coefficient, being sure to increment the subscript on μ_i.
7. Repeat steps 2 through 6 as necessary, generating many sets of $\widehat{\beta_0}$. (At least 2,500 is suggested.) Let B be a matrix with k columns and a finite number of rows. Set the lag equal to W so that we save a set of $\widehat{\beta_0}$ every W iterations. If the iteration number is congruent to 0 modulo W, then save this particular set of $\widehat{\beta_0}$ in the next empty row of B. Otherwise, the set does not need to be saved.
8. **Calculation of Final Vector of Betas**: Once all sets of $\widehat{\beta_0}$ have been generated and saved, take the mean of each column of B. This results in a vector $\hat{\beta} = \langle \beta_0, \beta_1, \ldots, \beta_k \rangle$ that serves as the Bayesian approximation of the regression

coefficients for model M_T.

Observe that since each β_i is sampled from a distribution that depends upon the previously sampled value, the sets of coefficients that are generated consecutively are not independent. This is the reason we must include a lag when saving sets of regression coefficients. For a detailed explanation of Metropolis sampling, one can consult [3].

2.3 Least Squares Model Selection

The second way to identify a model is through mixed stepwise regression, a least squares approach. Section 2.3.1 explains the Newton-Raphson method of maximum likelihood estimation, which is used to calculate the regression coefficients for any given model. Section 2.3.2 describes how mixed stepwise regression is used to arrive at a model.

2.3.1 Newton-Raphson Method

The Newton-Raphson method is a method of maximum likelihood estimation that we will use to maximize the likelihood of a given model. Similar to the section on Metropolis sampling, let M_T be the statistical model for which we need to find coefficients for the included variables. Then the logistic model is:

$$P(Y) = \frac{e^{\beta_0 + \sum_{X_i \in M_T} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_T} \beta_i X_i}}$$

where variable X_i is included in the model only if it is also in the set M_T. Again, assume M_T contains k unique variables that are denoted by $X_1, X_2, \ldots, X_k$. Finding the least squares coefficients is equivalent to maximizing the likelihood function of the model. The likelihood function used in this method is the same as in Section 2.1:

$$P(Y|M_0, \beta_0, \beta_1, \dots, \beta_k) = \prod_{j=1}^{N} \left(\frac{e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right)^{Y_j} \left(1 - \frac{e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right)^{1-Y_j}.$$

However, the likelihood function is computationally difficult to maximize, so instead we maximize the natural logarithm of the likelihood function, denoted by l. Then l is given by

$$l = \sum_{J=1}^{n} \left[Y_j \ln \left(\frac{e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right) + (1 - Y_j) \ln \left(1 - \frac{e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right) \right]$$

which simplifies to

$$l = \sum_{j=1}^{N} \left[Y_j \left(\beta_0 + \sum_{X_i \in M_0} \beta_j X_{ij} \right) - \ln\left(1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}\right) \right].$$

We will use the Newton-Raphson algorithm to solve for the regression coefficients that maximize the log-likelihood function. The algorithm contains five steps:

1. **Initialization**: Let $\widehat{\beta_0}$ be a vector of length $k + 1$ such that $\widehat{\beta_0} = \langle \beta_0, \beta_1, \dots, \beta_k \rangle$, where $\widehat{\beta_0}$ is the initial prediction for the values of $\beta_0, \beta_1, \dots, \beta_k$ in the logistic function.
2. **Gradient Vector**: Calculate the gradient vector $\frac{dl}{d\widehat{\beta_0}} = \left(\frac{dl}{d\beta_0}, \frac{dl}{d\beta_1}, \dots, \frac{dl}{d\beta_k} \right)$ where each $\frac{dl}{d\beta_j}$ is given by

$$\frac{dl}{d\beta_j} = \sum_{i=0}^{N} \left(X_{ij} Y_i - \frac{X_{ij} e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}} \right).$$

Note that $X_{ij} = 0$ whenever $j = 0$.

3. **Hessian Matrix**: Calculate the Hessian matrix

$$\frac{d^2 l}{d\widehat{\beta_0}} = \begin{bmatrix} \frac{d^2 l}{d\beta_0^2} & \cdots & \frac{d^2 l}{d\beta_0 d\beta_k} \\ \vdots & \ddots & \vdots \\ \frac{d^2 l}{d\beta_0 d\beta_k} & \cdots & \frac{d^2 l}{d\beta_k^2} \end{bmatrix}$$

where each $\frac{d^2 l}{d\beta_j d\beta_k}$ is given by

$$\frac{d^2 l}{d\beta_j d\beta_k} = \sum_{i=1}^{N} \frac{X_{ij} X_{ik} e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}}{\left(1 + e^{\beta_0 + \sum_{X_i \in M_0} \beta_i X_i}\right)^2}.$$

4. **Update**: Update the values of $\beta_0, \beta_1, \ldots, \beta_k$ by setting $\widehat{\beta_1} = \widehat{\beta_0} + \left(\frac{d^2 l}{d\widehat{\beta_0}}\right)^{-1} \left(\frac{dl}{d\widehat{\beta_0}}\right)$.

5. Define $\varepsilon > 0$ to be some tolerance used as a stopping criterion. Calculate $\left\|\widehat{\beta_1} - \widehat{\beta_0}\right\|_2$. If $\left\|\widehat{\beta_1} - \widehat{\beta_0}\right\|_2 < \varepsilon$, then stop. Otherwise, set $\widehat{\beta_0} = \widehat{\beta_1}$, and repeat steps 2 through 5 as necessary until the stopping criterion is satisfied.

The Newton-Raphson algorithm will converge to the least squares estimates of the regression coefficients once the difference between $\widehat{\beta_0}$ and $\widehat{\beta_1}$ is small enough.

2.3.2 Mixed Stepwise Logistic Regression

Mixed stepwise logistic regression is a second method of selecting a model from the list of n variables at our disposal. Rather than including and removing variables from a model with some probability, mixed stepwise regression is an algorithmic process that will always arrive at the same conclusion each time it is executed. It uses the likelihood ratio test as the test statistic to include variables if it is less than some threshold $\alpha = \alpha_0$

and remove variables if rises above some other threshold $\alpha = \alpha_1$. The algorithm has four steps:

1. **Initialization**: Begin with a model M' containing only the intercept and no variables. Use the Newton-Raphson method from Section 2.3.1 to find the least squares estimate for β_0. Calculate the log likelihood of this model using the log likelihood function given in Section 2.3.1, and call it l_0.
2. **Forward Selection**: Let n' be the number of variables not included in model M', and let L' be a vector of length n'. Begin by selecting the first variable not in M' and adding it to M'. Use the Newton-Raphson method to calculate the regression coefficients, and then calculate the log-likelihood of this model, denoted l_1. Calculate the test statistic D between the null model M' and the alternative model where a variable is added to M' using $D = -2(l_0 - l_1)$. Place this value in the first cell of L'. Remove the first variable from M' and add the second variable that was not originally in M'. Repeat the same process of adding a variable to M', calculating the regression coefficients, computing the log-likelihoods, finding the test statistic, and placing it in L' until each of the n' variables has been tested with M'. Choose the value of L' that results in the smallest p-value from the test statistic and add it to model M' assuming its significance level is less than α_0. Call this new model M'', and let the likelihood of M'' be l_1.
3. **Backward Selection**: Let n'' be the number of variables included in model M'', and let L'' be a vector of length n''. Begin by selecting the first variable included in M'' and remove it. Use the Newton-Raphson method to calculate the regression coefficients, and then calculate the log-likelihood of this model,

denoted l_2. Calculate the test statistic D between the null model M'' and the alternative model where a variable is removed from M'' using $D = -2(l_1 - l_2)$. Place this value in the first cell of L''. Add this first variable back into the model and remove the second variable (if it exists) that was included in M''. Repeat the same process of removing a variable from M'', calculating the regression coefficients, computing the log-likelihoods, find the test statistic, and placing the it in L'' until each of the n'' variables has been removed from M'' individually. Choose the value in L'' that results in the largest p-value. If this p-value is greater than α_1, remove the variable from the model. Repeat Step 3 until all variables that are no longer significant are removed. Once only significant variables remain in the model, call this new model M', and let the likelihood of M' be l_0.

4. Repeat Steps 2 and 3, adding and removing variables from the model until $M' = M''$ at the conclusion of the Backward Selection process.

Once the algorithm finishes running, the least squares model selection process is complete. We can define the least squares model to include all variables in M'. Use the Newton-Raphson algorithm one final time to calculate the least squares regression coefficients. To delve deeper into the specifics behind stepwise logistic regression, the reader should consult [4].

3. Results

Returning back to the original problem, the results will be presented in the following manner. First, the data collection process will be explained since typical statistics in college basketball are not used directly in the analysis. Next, the implementation of the Metropolis Hastings algorithm and the Metropolis sampling will be discussed in the context of the problem. Third, we will describe the model selection process and describe the models that were used in the comparisons. Finally, we will test the models in three different settings and present the results.

3.1 Data Collection

In order to build the model, data from the 2001-2002 NCAA basketball season through the 2010-2011 season were first collected. The statistics were all accumulated from statsheet.com and the predictor variables are listed in Table 1 below [5].

Table 1: List of Statistics Collected for Each Team in the NCAA Tournament

General Team Statistics	Team Game Statistics	Team Game Statistics (cont.)
• Conference • Tournament champion • Wins in last 10 games • AP Poll preseason ranking • Starting five years of seniority • Overall winning percentage • Conference winning percentage	• Points per game • Field goal percentage • Free throw shooting percentage • 3 point field goal percentage • Offensive rebounds/game • Defensive rebounds/game • Assists per game • Steals per game • Blocks per game • Turnovers per game	• Personal fouls per game • Points per possession • Effective field goal percentage • True shooting percentage • Assist percentage • Steal percentage • Block percentage • Turnover percentage • Assist to turnover ratio

The process of building the dataset occurred in several steps. The procedure is not straightforward, so we will intertwine an example with the explanation. Note that all data collected was from the regular season only and did not include any games in the NCAA Tournament. First, for each of the 64 teams in the tournament, the seven statistics in the first column were collected. Moreover, for the remaining 19 statistics in the second and third columns, each teams' offensive season statistics were gathered, as well as the corresponding defensive statistics for a total of 45 statistics for each of the 64 teams. For example, during the 2010-2011 NCAA basketball season, Butler scored an average of 72.81 points per game while giving up 64.66 points per game to its opponents. Similarly, Old Dominion averaged 65.85 points per game, but allowed its opponents to score only 58.30 points per game. Next, the teams were paired according to the matchups that actually occurred in the year's NCAA tournament. As the tournament is single elimination, this resulted in 63 games per year and 630 games in the above time frame. In the 2011 tournament, eighth seeded Butler and ninth seeded Old Dominion were placed in the same bracket. As a result, they faced off in the first round of the tournament, so this accounts for one of the 63 games in the 2011 tournament. Third, the statistics used in the dataset were calculated as follows. The first seven variables were the general team statistics for the higher seeded team. The next seven variables were the general team statistics for the lower seeded team. For the 19 statistics reflecting each team's offensive performance, the lower seeded team's statistics were subtracted from the higher seeded team's statistics. From our example, since Butler was seeded higher than Old Dominion, we take Butler's 72.81 points scored per game and subtract Old Dominion's 65.85 scored points per game, yielding a 6.96 point advantage for Butler. Thus, for the difference in points scored per game, the statistic for this game would be 6.96. For the 19 statistics reflecting each team's opponents' performance, the higher seeded team's statistics were subtracted from the lower seeded team's statistics. In our example, we take Old Dominion's 58.30 points allowed per game and subtract Butler's 64.66 points allowed per game to get a −6.36 point advantage or a 6.36 point disadvantage for Butler. Thus, for the difference in points allowed per game, the statistic for this game in the dataset is −6.36. As a result, positive numbers indicate an advantage for the higher seeded team, while negative numbers indicate an advantage for the lower seeded team. This process was repeated for each of the 38 statistics. Each number was entered into the dataset for a total of 52 variables for each of the 630 games. The dependent variable is nominal and coded as '1' if the higher seeded team won the game and '0' if the lower seeded team won. The same process was repeated for teams competing in the 2012 tournament, where these 63 games were used as the validation set. For a complete list of each individual variable in the dataset and its corresponding number used to reference it in the code, consult Appendix I.

Most of the statistics collected are self explanatory. All but the first two are treated as continuous variables. The reader interested in the true definitions of the above basketball statistics may consult [5] or any one of many other resources available. However, the one variable that must be addressed directly is the team's conference. Here, conference is a nominal variable with three levels of measurement. In college basketball, there are roughly 340 teams divided into 31 conferences. These conferences are not of equal talent, and are generally divided into power conferences, mid-major conferences, and small conferences. Here, we use the Ratings Percentage Index (RPI), a statistical measure based primarily on a team's wins, losses, and strength of schedule to rank a team based on conference. The RPI of a conference is calculated by summing the RPIs of all teams in the conference and dividing by the number of teams. If a team is in a conference whose RPI is at least .550, then conference is coded as '1' in the database for that team. If its conference RPI is between .500 and .550 inclusive, then conference is coded as '2'. For conference RPIs under .500, the team is assigned '3' for its conference.

Since conference is coded nominally, we must create two dummy variables, Z_1 and Z_2 in order to perform the regression without using ANACOVA. Using the mid-major teams, coded as '2' as the reference group, if a team's conference is coded as '1', then $Z_1 = 1$ and $Z_2 = 0$. If the conference is coded as '3', then $Z_1 = 0$ and $Z_2 = 1$. Finally, if conference is coded as '2', then $Z_1 = Z_2 = 0$. Doing this for both teams in each game brings the total number of variables to choose from to 54. Note that the variable tournament champion is also nominal, but since it has only two levels of measurement ('1' if the team won the tournament, and '0' if not), it is already coded as if a dummy variable existed.

3.2 Implementation of Algorithms

Given the 630 games that actually occurred between the 2002 and 2011 tournaments, 120 of these were first round matchups between 1 and 16, 2 and 15, or 3 and 14 seeds. The lower seeded team won only three of these games. (Only Kansas in 2005, Iowa in 2006, and Georgetown in 2011 were given a top three seed and lost). Rather than include these games in the dataset and try to get the model to predict these outcomes, it was decided to remove these games from the dataset and move teams seeded 1, 2, and 3 on to the second round with probability 1. This leaves us with 510 games in the dataset.

From here, two different ways to use the data to develop a model were implemented. The first involved using the Metropolis-Hastings algorithm on all 510 games at once to create one single model for all of the data. However, another approach is to divide the data up into the rounds in which the games occurred. Teams with a high seed often play a different type of game in the first round than they would in later rounds due to playing an easier opponent in the first round. It is unlikely that all variables that are important in the first round have the same importance when playing in the championship game and vice versa. As a result, we propose dividing the 510 games into three separate datasets: one containing the remaining 200 first round games between 2002 and 2011, one containing the 160 second round games, and one containing the final 150 games between rounds three and six. Abiding by this process will result in a piecewise model where the model used to predict the results of the NCAA tournament depends on the round the game is being played.

We used the algorithms described in Section 2 to generate models on each of the above four datasets under the following assumptions:

1. The main Metropolis-Hastings algorithm was allowed to run for 100,000 iterations, beginning with the model that included only the intercept. The burn in period was 100 iterations.
2. In order to approximate the likelihood of each model using Monte Carlo integration, a uniform prior on the interval $[-2,2]$ was used instead of the typical normal prior as suggested in Section 2.1.
3. When performing the numerical integration, a matrix with k rows and 10,000 columns was created. Each entry was filled with a random integer from the uniform interval $[-2,2]$ using the random number generator in MATLAB R2007a.

4. In order to calculate the regression coefficients for a model using Metropolis sampling as described in Section 2.2.2, 5,000 sets of coefficients were accepted using a lag of 10 to guarantee independence of the samples.

3.3 Model Selection Techniques

After running the Metropolis-Hastings algorithm under the above conditions on each of the four datasets, the 100,000 models that resulted were reduced to the unique models. The number of times each of the unique models appeared in those 100,000 iterations were counted. The results are presented in Table 2.

Table 2: Number of Unique Models and Models Appearing at Least 500 Times

Dataset	Number of Unique Models	Models Appearing at least 500 Times
Round 1	1960	36
Round 2	1103	37
Rounds 3-6	2698	35
All Rounds	1147	19

From here, a number of different model selection techniques were used to identify potential models that would be good predictors for the NCAA Tournament. In the first technique, which we will call Method 1, we identified all of the models in each dataset that appeared at least 0.5% of the time, or 500 of the 100,000 iterations. Marginal probabilities for these models were calculated, allowing us to calculate the marginal probabilities for each variable. Then, the data for each variable was multiplied by its corresponding marginal probability. Metropolis sampling was used to calculate the final regression coefficients for these particular models. When creating the final models, we chose to use one model that included all variables in proportion to their marginal probabilities and one that allowed only variables with a marginal probability of at least 30% to be included in the model. This model selection technique creates four models:

one with no threshold that predicts games in all rounds, one with a threshold at 30% that predicts games in all rounds, a piecewise model created by applying the model selection technique to the three smaller datasets and predicting games by using the model that corresponds to the appropriate round, and another piecewise model where the variables included in each piece abides by the 30% threshold for the marginal probability.

The second technique, which we will call Method 2, involved using marginal probabilities on the models themselves. Using all models that appeared at least 0.5% of the time again, Metropolis sampling was used to calculate the regression coefficients for each individual model. These coefficients were then multiplied by the corresponding marginal probability for the model. Taking the sum across all models for each variable and dividing by the total included probability resulted in the final set of regression coefficients for the second method. No thresholding was used in this method. We accumulate two additional models to use for predictive purposes: one that is used to predict all games and another piecewise model.

We may also choose individual models generated from the Metropolis-Hastings algorithm to use as predictors. The model that appeared the most often out of the 100,000 iterations for each dataset was chosen. We again built the piecewise model by using the most likely model from each of the smaller datasets to develop a second full model from this technique. Metropolis sampling was used to calculate the regression coefficients for each of the four models above.

The final Bayesian technique involved selecting the most likely model of each size in each of the datasets. From here, Metropolis sampling was used to calculate the regression coefficients in each of the individual models. The accuracy of each of the models was then calculated using the datasets, and the model with the highest accuracy in each of the datasets was chosen. Note that if two models had the same accuracy for the same dataset, the one with fewer predictor variables was selected. The accuracies of the most likely model of each size are included in the table below, and the chosen model is in bold.

Table 3: Accuracy of Most Likely Model of Each Size for Each Dataset in Test Set

No. of Variables	All Games Data	Round 1 Data	Round 2 Data	Round 3 Data
1	.651	.635	.6938	.6267
2	.651	.635	.6938	.64
3	.651	.635	.6938	.6267
4	.649	.64	.7438	.62
5	.6451	.67	.6688	.6667
6	.6627	.645	.70	.6733
7	.6373	.68	.6938	.6667
8	.6431	.645	.7438	.6733
9	**.6765**	.635	.70	.6767
10	.6765	.68	.7188	.6733
11	.6686	.70	.7375	.66
12	.6608	**.71**	**.7563**	.66
13		.70	.6818	.6667
14		.70	.7563	.6533
15		.69	.7375	.6733
16			.7375	**.68**

In addition to the Bayesian models, we will compare the results to the least squares models that have been developed using mixed stepwise regression as described in Section 2.2.2 and the Newton-Raphson method from Section 2.3.2. The variable needed to be statistically significant at the $\alpha = .25$ level in order to be included in the model during the forward selection process. If a variable was already included, but its statistical significance rose above the $\alpha = .25$ level at some point during the backward selection process, then it was removed from the model. Running these algorithms on each of the four datasets will result in two final models: one for all games, and one piecewise model.

In summary, below we define the models that will be tested against the Pomeroy ratings, Sagarin ratings, and choosing winners of games based on the teams' seeds. From this point forward, these names are how we will refer to the models.

- Model 1: The model used for all games developed using Method 1.
- Model 2: The piecewise model created using Method 1.
- Model 3: The model used for all games from Method 1 using a threshold of 30%.
- Model 4: The piecewise model created from Method 1 with a threshold of 30%.
- Model 5: The model used for all games generated from Method 2.
- Model 6: The piecewise model developed using Method 2.
- Model 7: The most likely model that was used from the dataset with all 510 games.
- Model 8: The piecewise model created from the most likely model from each of the three small datasets.
- Model 9: The most accurate model out of the most likely model of each size used for all games.
- Model 10: The piecewise model generated by selecting the most accurate model out of the most likely model of each size in each of the smaller datasets.
- Model 11: The model used for all games created using the least squares method.
- Model 12: The piecewise model developed using the least squares method on each of the three smaller datasets.

Though Models 11 and 12 were devised using least squares methods, the regression coefficients used in the analysis in all 12 models were calculated using Metropolis sampling. From this point forward, Models 1 through 10 will be referred to as the Bayesian models, while Models 11 and 12 will be called the least squares models. A list of the included variables in each model can be found in Appendix II.

3.4 Performance of the Models

How each of the models performed as well as a comparison to picking winners by seed, Pomeroy ratings, and Sagarin ratings will be divided into three sections. The first will present the results for how each model did in its respective test set. The second section will be used to predict the winners of the 63 games that actually occurred in the 2012 NCAA Tournament. In the third section, we will use each of the models to actually fill out a bracket round by round to see how well each model does without knowing how far each team will advance. Note that this is different from the second section since in the second section, we know which games will occur beyond the first round, but in the third section, the games are dependent upon which teams the model predicted would win in the previous rounds. In all two sample t-tests that were performed, the degrees of freedom were calculated using the equation:

$$df = \left\lfloor \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\left(\frac{1}{n_1 - 1}\right)\left(\frac{s_1^2}{n_1}\right) + \left(\frac{1}{n_2 - 1}\right)\left(\frac{s_2^2}{n_2}\right)} \right\rfloor$$

where s_1 and s_2 represent the sample standards deviations of groups 1 and 2 respectively, and n_1 and n_2 represent the sample sizes of groups 1 and 2 respectively.

3.4.1 Performance in the Test Set

Over the ten year test period, there were 320 first round games played, 160 second round games, 80 third round games, 40 fourth round games, 20 fifth round games, and ten sixth round games. Recall the decision to automatically advance all 1, 2, and 3 seeds to the second round of the tournament; this gives us an additional 117 games correct in the first round. The number of games correct in each round for each of the models as well as selecting winners based on seed is listed in the following table. The exact pre-tournament Pomeroy and Sagarin ratings between 2002 and 2011 were not available to determine the accuracy to which their systems would have performed.

Table 4: Number of Games Predicted Correctly in Each Round for Each Model in the Test Set

Model	**Round 1**	**Round 2**	**Round 3**	**Round 4**	**Round 5**	**Round 6**	**Total**
Model 1	**252**	**114**	**58**	**23**	**10**	**6**	**463**
Model 2	**268**	**125**	**60**	**26**	**12**	**9**	**500**
Model 3	**254**	**109**	**53**	**23**	**9**	**5**	**453**
Model 4	**260**	**116**	**54**	**26**	**11**	**9**	**476**
Model 5	**251**	**114**	**57**	**24**	**11**	**5**	**462**
Model 6	**256**	**125**	**55**	**25**	**12**	**9**	**482**
Model 7	**250**	**115**	**57**	**24**	**10**	**5**	**461**
Model 8	**253**	**132**	**59**	**23**	**11**	**8**	**486**
Model 9	**254**	**111**	**57**	**23**	**11**	**5**	**461**
Model 10	**259**	**121**	**58**	**26**	**12**	**7**	**483**
Model 11	**248**	**121**	**55**	**24**	**9**	**7**	**464**
Model 12	**260**	**126**	**54**	**25**	**10**	**9**	**484**
Seed	244	99	47	15	5	1	411

Observe first that all of our models greatly outperformed choosing the higher seeded team to win. On average, the Bayesian models predicted 472.7 games correctly over the ten year span for an accuracy of 75%. Compare this to choosing the winners based on the seeds, which was 65.2% accurate. Model 3, the worst one at just under 72%, was almost 7% better than choosing the winners from the teams' seeds. Model 2 nearly eclipsed 80% accuracy, which would have resulted in missing only 13 games per year given that the model knows all of the games ahead of time. Even more impressive is the Bayesian models' ability to choose the winner of the national championship game. On average, the Bayesian models predicted 6.8 of the 10 championship games correctly; choosing based on seed correctly forecasted only Florida in 2007 to win the title. Three of the Bayesian models missed only one championship game: Syracuse in 2003.

A comparison between the average number of games correct in each round for the Bayesian models and the least squares models is available in the table below:

Table 5: Comparison of Bayesian and Least Squares Models in Test Set

Model	Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Total
Bayesian	255.7	118.2	56.8	24.3	10.9	6.8	472.7
Least Sq.	254	123.5	54.5	24.5	9.5	8	474
Difference	***1.7***	*-5.3*	***2.3***	*-0.2*	***1.4***	*-1.2*	*-1.3*

The Bayesian models were more accurate in rounds 1, 3, and 5 while the least squares models outperformed the Bayesian models in rounds 2, 4, and 6 and overall. Testing for the equality of means assuming unequal variances in each round and overall with a significance level of $\alpha = .05$ and the following number of degrees of freedom divulges the following information:

Table 6: T-Tests Performed on the Difference Between Bayesian and Least Squares Models for Each Round in Test Set

Model	Degrees of Freedom	t-Statistic	P-value
Round 1	1	0.273	.831
Round 2	3	-1.559	.217
Round 3	**6**	**2.684**	**.026**
Round 4	3	-0.390	.780
Round 5	2	2.370	.141
Round 6	2	-1.041	.407
Overall	1	-0.118	.926

Through the above t-tests, it is revealed that only the difference in round 3 between the Bayesian and least squares models is statistically significant. The other six, including the overall total, are not significantly different, which leads us to conclude that the Bayesian and least squares models performed approximately equally in all aspects. However, with no more than three degrees of freedom in any of the other tests, there is not much of an opportunity to discover a significant effect, lowering the statistical power.

3.4.2 Predicting the Actual 63 Games in the 2012 NCAA Tournament

Testing each of the models on the games that occurred in the 2012 NCAA tournament returns the following results. Here, we allow each model to fix its mistakes if it made an incorrect prediction in the previous round and make a prediction on the correct game. We again assign a probability of 1 to the 1, 2, and 3 seeds moving on to the second round. There are 32 first round games played, 16 second round games, eight third round games, four fourth round games, two fifth round games, and one sixth round game to determine the national champion. We include the results for games chosen by seed, the Pomeroy ratings, and the Sagarin ratings here as well.

Table 7: Number of Games Predicted Correctly in Each Round for Each Model in Actual 63 Games

Model	Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Total
Model 1	20	**14**	**7**	1	**1**	**1**	44
Model 2	19	**13**	**7**	2	**2**	**1**	44
Model 3	**22**	12	6	1	**1**	**1**	43
Model 4	21	**16**	6	**3**	**2**	**1**	**49**
Model 5	20	**14**	6	1	**2**	**1**	44
Model 6	21	**15**	6	2	**2**	**1**	**47**
Model 7	21	**13**	**7**	1	**2**	**1**	45
Model 8	**23**	**15**	6	2	**2**	**1**	**49**
Model 9	20	**13**	6	1	**2**	**1**	43
Model 10	**24**	**13**	6	2	**2**	**1**	**48**
Model 11	20	11	6	2	**1**	**1**	41
Model 12	20	11	6	**3**	**1**	**1**	42
Pomeroy	22	12	5	3	1	1	44
Sagarin	22	13	7	3	1	1	47
Seed	22	13	6	2	2	1	46

Comparing the Bayesian models to the brackets filled out using the Pomeroy and Sagarin ratings uncovers a slightly different story. In this year's tournament, the Sagarin ratings predicted three more games correctly than the Pomeroy ratings did. Three of our ten Bayesian models choose more games correctly than the Sagarin ratings, and a fourth equaled his total of 47. Conversely, five of our Bayesian models outperformed the Pomeroy total of 44, while another three tied his total. The final two models were only one game behind at 43 games correct. It is interesting to note that this year, choosing the true games by seed resulted in 46 correct picks, a better accuracy than the Pomeroy ratings. Thus, we can conclude that, given the true games that occurred, most of our models can be expected to perform at least as well as the Pomeroy and Sagarin ratings; some will be expected to do better. Observe also that of the five piecewise Bayesian models, four of them performed as well as if not better than Sagarin's ratings, and all five have accuracies equal to or better than Pomeroy's ratings. However, none of the five Bayesian models generated from the dataset containing all 630 games did as well as Sagarin, and only three did as well as Pomeroy. This provides some justification for splitting up the data by round and creating different models for different rounds. It also implies that some statistics are more important later on in the tournament than in the first round, and vice versa.

We will now compare the 2012 performance of the Bayesian models to the least squares models. The averages across the two methods are displayed in the following table:

Table 8: Comparison of Bayesian and Least Squares Models in the Actual 63 Games

Model	**Round 1**	**Round 2**	**Round 3**	**Round 4**	**Round 5**	**Round 6**	**Total**
Bayesian	21.1	13.8	6.3	1.6	1.8	1	45.6
Least Sq.	20	11	6	2.5	1	1	41.5
Difference	***1.1***	***2.8***	***0.3***	*-0.9*	***0.8***	*0*	***4.1***

The Bayesian models predicted more games correctly on average than the least squares models in rounds 1, 2, 3, and 5, as well as overall, while the least squares model did better in round 4. Each of the 12 models selected Kentucky to defeat Kansas in round 6, so there is no difference in means or variance within either group. Testing for the equality of means using a significance level of $\alpha = .05$ and the following number of degrees of freedom reveals the following:

Table 9: T-Tests Performed on the Difference Between Bayesian and Least Squares Models for Each Round in Actual 63 Games

Model	Degrees of Freedom	t-Statistic	P-value
Round 1	**9**	**2.282**	**.048**
Round 2	**9**	**7.203**	**.00005**
Round 3	9	1.964	.081
Round 4	1	-1.646	.348
Round 5	**9**	**6.000**	**.0002**
Overall	**7**	**4.494**	**.003**

Since there was no variation between the predictions in round 6, a significance test could not be performed. Here we discover that the differences in rounds 1, 2, and 5, as well as the total games correct are all statistically significant. Thus, we can conclude that the Bayesian models predicted the winners of the 63 actual tournament games in 2012 better than the least squares models in most aspects with rounds 3 and 4 undetermined. However, the test in round four only had one degree of freedom, making it a weak test.

3.4.3 Using the Models to Fill Out the 2012 Bracket

In the third measurement of accuracy, we used each model and system to fill out a bracket as if it were the beginning of the tournament and only the first round games had been determined. Different from the previous section, we do not allow the models to fix their mistakes and force them to make predictions on the games they believed would occur, even if those teams were actually eliminated earlier in the actual 2012 tournament. In order to predict the game correctly, it is not imperative that both teams in the matchup are correct. As long as the team that won the game in the tournament is predicted to win in the model, it is counted as a success even if their opponent is different. This is the scenario that we are most interested in because most March Madness competitions do not allow participants to select games round by round.

Table 10: Number of Games Predicted Correctly in Each Round for Each Model when Choosing Games to Fill Out the 2012 Bracket

Model	Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Total
Model 1	20	**10**	**5**	1	**1**	**1**	38
Model 2	19	8	**5**	1	**1**	**1**	35
Model 3	**22**	9	4	1	**1**	**1**	38
Model 4	21	**11**	**6**	2	0	0	40
Model 5	20	**11**	**5**	1	**1**	**1**	39
Model 6	21	9	4	1	**1**	**1**	37
Model 7	21	**11**	**5**	1	**1**	**1**	40
Model 8	**23**	**10**	4	1	0	0	38
Model 9	20	**11**	**5**	1	**1**	**1**	39
Model 10	**24**	**11**	**7**	2	**1**	**1**	**46**
Model 11	20	8	**5**	2	**2**	**1**	38
Model 12	20	8	**6**	2	**1**	**1**	38
Pomeroy	22	9	5	3	1	1	41
Sagarin	22	10	5	3	1	0	41
Seed	22	11	5	1	1	1	41

Comparing the brackets chosen using the ten Bayesian models to the ones from the Pomeroy and Sagarin ratings or by seed, most performed worse with Model 10 being the exception. We will discuss Model 10 later, leaving it out of this analysis, and focus on why the other models failed to achieve a higher accuracy. Note that the accuracies were relatively low this year for all models, including Pomeroy and Sagarin (65% for both versus their ten year averages of 74% and 73% respectively) due to two major upsets in the first round. Missouri and Duke, both awarded two seeds, fell to fifteen seeds Norfolk State and Lehigh. Nine of our Bayesian models as well as Pomeroy and Sagarin had Missouri and Duke winning at least their first two games; some of our models advanced

Missouri to the Final Four. Since these upsets are nearly impossible to predict, they affected all brackets that we are comparing here in a similar manner. Excluding Model 10, the other nine Bayesian models averaged 20.78 games correct in the first round versus 22 for both Pomeroy and Sagarin. Two major disparities between our models and Pomeroy's and Sagarin's ratings systems made the first round slightly less accurate. Both Pomeroy and Sagarin predicted fourth seeded Louisville to defeat thirteenth seeded Davidson in the first round. However, all of our models except Model 10 predicted a Davidson upset. This appears to be a case where a strong team coming from a weaker conference had its regular season statistics skewed by playing half of its games against teams that were inferior. Since Louisville advanced to the Final Four, each of these models immediately lost the opportunity to get four games correct by eliminating Louisville in the first round. Second, each of our models also failed to predict a Kansas State victory over Southern Mississippi in the first round, though Pomeroy and Sagarin both did; this was not as damaging since Kansas State was defeated by Syracuse in the next round. The second and third round predictions in our models were comparable to Pomeroy and Sagarin. Our models correctly identified an average of 10 Sweet Sixteen teams (versus 9 for Pomeroy and 10 for Sagarin) and 4.78 Elite Eight teams (against 5 for both Pomeroy and Sagarin). Four of the nine Bayesian models did correctly predict 11 Sweet Sixteen teams, which is notable considering two solid candidates in Missouri and Duke were eliminated in the first round. However, predicting teams that advanced to the Final Four proved to be a challenge for our models. Whereas Pomeroy and Sagarin both identified Kentucky, Ohio State, and Kansas as Final Four teams, our nine other models failed to predict more than two Final Four teams correctly. Each one predicted Kentucky to advance to the Final Four. Model 4 was the only one to predict a second Final Four team correctly (Kansas), but failed to predict either team that would play for the national championship, opting for Michigan State over Syracuse.

We will now compare the Bayesian methods to the least squares methods. Observe that the Bayesian models averaged 39 games correct, whereas the least squares methods averaged 38 games correct. The Bayesian models outperformed the least squares models in the second round, choosing an average of 10.1 Sweet Sixteen teams correctly against the least squares average of eight. However, the least squares models slightly outperformed the Bayesian models from this point forward. In the third round, Models 11 and 12 correctly identified an average of 5.5 Elite Eight teams versus five for the Bayesian models. The Bayesian models struggled to choose the Final Four teams correctly, with only two of ten models getting half of the teams correct; two models even failed to predict that either Kentucky or Kansas would play for the National Championship. The least squares models both predicted that Kentucky would advance to the Final Four. Model 11 correctly predicted that Kentucky would defeat Kansas in the championship game, while Model 12 believed Ohio State would join Kentucky in the Final Four. Overall, the Bayesian methods appeared to predict games more accurately in the first and second rounds of the tournament, but performed slightly worse than the least squares models in the last four rounds. However, given the close small differences in means and the small sample sizes in number of games, none of the differences are statistically significant.

Model 10 was the silver lining among all of the other models that underachieved. It achieved the third highest accuracy in the test set at 76.67%. Correctly predicting the winners of 46 of the 63 spots in the bracket (in spite of the two major first round upsets) resulted in an accuracy of 73%, far exceeding the accuracy of Pomeroy's and Sagarin's brackets, as well as the on chosen based on seeds, each of which was 65% accurate. Model 10 was primarily successful by minimizing the damage done in a volatile bottom left bracket, missing only four games; Pomeroy and Sagarin both missed on seven of the fifteen games here. Model 10 succeeded in not only choosing Louisville to defeat Davidson in the first round, but by predicting them to advance to the Final Four, the only above method of choosing a bracket to do so. Along the way, Louisville defeated New Mexico, St. Louis, and Murray State. It is worth noting that St. Louis being predicted to win over Michigan State proved to be crucial since the model would have picked Michigan State over Louisville had they met. Sending Louisville to the Final Four provided an extra three wins over Pomeroy's and Sagarin's brackets, both of whom had New Mexico defeating Louisville in the second round. Also assisting in the accuracy of Model 10 was its ability to identify Missouri as a team that would lose early. Although we advanced them to the second round immediately by virtue of their seed, Model 10 correctly predicted that Florida would advance to the Sweet Sixteen, knocking out Missouri in our bracket instead of Norfolk State in reality. This paved the way for Murray State to advance to the Elite Eight. Otherwise, the model would have predicted Missouri to defeat both Murray State and Louisville in the third and fourth rounds respectively. Model 10 was the only system that advanced seven teams correctly to the Elite Eight, missing only Florida. Kentucky and Louisville were the only Final Four teams it got correct, missing on the opportunity to choose Kentucky over Kansas in the championship game. Model 10 achieved 83% accuracy between the top left and top right brackets. In the top left bracket, it missed on only the first round games between Notre Dame and Xavier, and Duke and Lehigh. Since the winners of these games faced off in the next round, it was impossible to predict the second round game correctly as our model chose Duke over Notre Dame, whereas Xavier defeated Lehigh in the tournament. In the top right bracket, like all of our models, Model 10 favored Southern Mississippi over Kansas State for the first miss, and opted for Syracuse as the regional champion over Ohio State. In the bottom right bracket, Model 10 performed somewhat worse than did Pomeroy and Sagarin, missing seven games compared to their five incorrect picks. Our model missed the same five games that the rating systems did, but also missed North Carolina State defeating San Diego State and Kansas emerging as the regional champion instead of North Carolina.

Strangely enough, Model 2, which achieved the highest accuracy (79.36%) of any model in the test set, performed the worst in the 2012 tournament with only 55.56% of the games correct. Though it often predicted a lower seeded team to win, it was often wrong. Of the ten upsets that occurred in the first round of the 2012 tournament, Model 2 correctly identified only three of them. Conversely, Model 2 incorrectly chose an additional six lower seeded teams to upset higher seeded teams, leading to a first round accuracy of only 59%. Of the fifteen games in the lower left bracket, Model 2 predicted only four games correct, all in the first round. It even sent thirteenth seeded Davidson to the Elite Eight. Model 2 struggled in the bottom right bracket in a similar manner that

Model 10 did, missing the same seven games. In the top left, Model 2 missed the same three games as Model 10, but also predicted Wichita State to defeat Virginia Commonwealth in the first round. Model 2 missed both games that Model 10 missed in the top right bracket, but also advanced Florida State to the Elite Eight instead of having them lose to Cincinnati in the second round.

Recall from Section 3.4.2 how Models 4 and 8 both missed only 14 of the 63 actual games, yielding the best accuracy of all the Bayesian methods. Both were good at fixing their mistakes and choosing the correct winner give the actual matchup. However, when using these models to choose winners from the beginning, neither performed admirably. Model 4 returned an accuracy of 63.5% without allowing it to fix its mistakes. Its biggest mistakes were picking Michigan State to win the national championship and advancing Missouri to the Elite Eight; these two teams accounted for seven of the 23 incorrect picks. Paired with four incorrect picks in toss-up games and having Syracuse in the championship game, this model made some bold picks that simply did not evolve. It did correctly identify Kentucky and Kansas as Final Four teams. Model 8, whose accuracy was worse at 60.3%, fell into the same trap as Model 4, choosing Michigan State as the champion and putting Missouri in the Elite Eight. However, this model also picked Florida State as Michigan State's opponent in the championship game. These three teams accounted for 11 of the 25 incorrect predictions. Though Model 8 missed badly in the last four rounds, it did perform well in the first round, making 23 correct predictions. Model 8 trailed only Model 10 in first round accuracy, missing the same eight games, as well as picking Davidson over Louisville.

4. Conclusion

Throughout this analysis of predicting the winners of games in the NCAA tournament, we have exhibited an ability to generate models that will retrospectively predict the winners of tournament games better than seeding alone does. Each of our models achieved a higher accuracy in the test set than simply by selecting the higher seeded team to win. However, this ability to predict the winners of the current year's games given that we do not know all 63 games at the beginning does not necessarily translate well to individual tournaments. Nine of our ten Bayesian models were outperformed by the Pomeroy and Sagarin ratings, as well as choosing the games based on seed. The Bayesian models, on average, did perform better than the least squares model though. On the other hand, given that we know the 63 actual games in the tournament, the Bayesian models performed about as well as the other rating systems on average. All ten outperformed both of the least squares models in this scenario.

The disappointing performance of Model 2 in the 2012 tournament may be partly due to the fact that model averaging was used to generate the model. When predicting outcomes retrospectively, this process should theoretically return the best results since all variables are included in proportion to their marginal probabilities. However, for making prospective predictions on small datasets, this technique appeared to fail. On the other hand, Model 10 performed quite well in all aspects. Recall that Model 10 was created by selecting the most accurate model out of the most likely models of each size. Doing this allowed each part of the piecewise model to identify different factors that were important during different rounds of the tournament. Choosing the most accurate models from the simulations ensured that the models did well in the test set. Since they performed well in the past, but also included only those variables that were important and excluded ones that were not statistically significant, Model 10 appears to be the best model to use to fill out a bracket at the beginning of the NCAA tournament.

Based on the predictions from one year's tournament, it is impossible to say whether or not any one of our Bayesian models is truly a success. The models generated from model averaging do not appear to do well in predicting a tournament from the beginning, although they performed about as well as we expected given the actual tournament games. Choosing the most accurate model from the Metropolis-Hastings algorithm provides some initial hope based on its success against the well known Pomeroy and Sagarin ratings. However, we will need to use future tournaments to ultimately reach a conclusion on its true success. Ultimately, matching up the teams playing in each game based on their regular season statistics and using those differences as the data proved to be an effective technique for making a prediction as to which team will win.

5. Further Discussion

Though we chose to use the two above described model selection techniques to perform variable selection, many other methods exist that would theoretically work just as well. Occam's window is a third technique that could have been used [6]. We also could have removed any model that had a more probable submodel from the analysis. With regards to sampling techniques, a Gibbs sampler could have been used in place of Metropolis sampling, although this would have made the process much less efficient since a larger lag would have been required to guarantee independence of the samples. One possible limitation to our results arises from restricting the uniform prior distributions on the regression coefficients to the interval $[-2,2]$. When each model's least squares regression coefficients were compared against the Bayesian regression coefficients, most fell into the above interval; however, for the few variables whose least squares regression coefficient was outside this interval, our results may have turned out better by allowing the regression coefficients to come from an interval with a larger width.

Similarly, while we used stepwise logistic regression to identify the least squares models, other least squares techniques exist. Branch and bound is a popular technique that searches the entire sample space of models by generating a sequence of variables that continually increase the likelihood, and eliminating the inclusion of a variable that does not contribute significantly to the likelihood [7]. Another least squares technique to help identify a logistic model is through partitioning, although one has to be careful so as not to force relationships in the data, thereby including variables that are not actually statistically significant even though the partition identifies a relationship.

Within the Newton-Raphson algorithm, there exists the potential for two problems, one of which we encountered. The first is the possibility that the algorithm will not converge to the least squares estimates of the regression coefficients, but rather to a different local maximum. We did not come across this situation, but the opportunity for this to occur exists. The second problem, which we did run across, was that the Hessian matrix in the third step of the algorithm was often singular and could not be inverted. To solve this problem, we used the pseudoinverse of the Hessian in its place.

One of the larger problems encountered throughout the course of this project in terms of basketball was the presence of strong mid-major and small conference teams that were so much better than the competition within their own conferences that their statistics were skewed to make these teams look much stronger than they actually were. Even imposing a penalty by classifying the teams by conference was not enough to make up for the difference. This was particularly evident with Davidson in the 2012 tournament, who was predicted to defeat eventual Final Four team Louisville in the first round in 11 of our 12 models, but in neither Pomeroy's nor Sagarin's ratings. Future research would try to find a way to accommodate for this skewed data. Another problem with these models is the inability to account for injuries and/or suspensions. However, this will be encountered in all models and rating systems as it is impossible to put a numeric value on a player and exactly how much he adds to the probability of his team winning.

The successful results from this paper have now left the door open for future research in this direction. Instead of using the season averages as data, the distribution of each statistic for each team could be used. A simulation could be created by sampling from each of those distributions and playing the game under those conditions instead of assuming the team is going to perform at its average level. This could open the door to the prediction of more upsets in cases where an inconsistent higher seeded team that takes risks is playing a consistent lower seeded team that always plays a solid game that does not deviate much from the mean. Similarly, one could cluster all the teams in NCAA Division I basketball according to some chosen set of regular season statistics. Similar opponents to the one in the tournament game could then be found and predictions could be made by comparing performances in similar games. Finally, a method of determining how volatile the tournament is going to be would be helpful in predicting winners. In recent years, we have seen tournaments with an unprecedented number of upsets, such as 2011 where none of the 1 and 2 seeds advanced to the Elite Eight; conversely, just two years earlier, the 2009 tournament saw all 12 1, 2, and 3 seeds advance to the Sweet Sixteen for the first time in history. Delving deeper into the regular season statistics and using teams outside of the tournament could reveal other information about what we should expect in terms of upsets.

REFERENCES

[1] Ross, S.M. (2006), “Simulation,” *Academic Press*, 38-42.

[2] Bolstad, W.M. (2010), “Understanding Computational Bayesian Statistics,” *Wiley*, 127-148.

[3] Hastings, W.K. (1970), “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, 57. 97-109.

[4] Hosmer, D.W. and Lemeshow, S. (2000), “Applied Logistic Regression,” *Wiley*, 116-127.

[5] Automated Insights. (2012), StatSheet. Retrieved from statsheet.com/mcb

[6] Raftery, A.E., Madigan, D., and Hoeting, J.A. (1997), “Bayesian Model Averaging for Linear Regression Models”, *American Statistical Association*, 179-189.

[7] Miller, A. (2002). “Subset Selection in Regression,” *Chapman & Hall*. 52-54.

Appendix I. Potential Variables Used for the Variable Selection Process

Var. No.	Variable Name	Var. No.	Variable Name
1	Constant	29	Diff. in Points Allowed per Game
2	Higher Seed- Conference =1 (1 or 0)	30	Diff. in Opponent Field Goal Pct.
3	Higher Seed- Conference =3 (1 or 0)	31	Diff. in Opponent Free Throw Pct.
4	Higher Seed- Won Conf. Tourn. (1 or 0)	32	Diff. in Opponent 3 Point Pct.
5	Higher Seed- Wins in Last 10 Games	33	Diff. in Opponent OR/Gm.
6	Higher Seed- Years of Seniority	34	Diff. in Opponent DR/Gm.
7	Higher Seed- Preseason AP Poll Ranking	35	Diff. in Opponent Assists/Gm.
8	Higher Seed- Winning Percentage	36	Diff. in Opponent Steals/Gm.
9	Higher Seed- Conference Winning Pct.	37	Diff. in Opponent Blocks/Gm.
10	Lower Seed- Conference =1 (1 or 0)	38	Diff. in Opponent Turnovers/Gm.
11	Lower Seed- Conference =3 (1 or 0)	39	Diff. in Opponent Per. Fouls/Gm.
12	Lower Seed- Won Conf. Tourn. (1 or 0)	40	Diff. in Points Scored per Possession
13	Lower Seed- Wins in Last 10 Games	41	Diff. in Effective Field Goal Pct.
14	Lower Seed- Years of Seniority	42	Diff. in True Shooting Pct.
15	Lower Seed- Preseason AP Poll Ranking	43	Diff. in Assist Pct.
16	Lower Seed- Winning Percentage	44	Diff. in Steal Pct.
17	Lower Seed- Conference Winning Pct.	45	Diff. in Block Pct.
18	Diff. in Points Scored per Game	46	Diff. in Turnover Pct.
19	Diff. in Field Goal Pct.	47	Diff. in Assist/Turnover Ratio
20	Diff. in Free Throw Pct.	48	Diff. in Opponent Pts per Possession
21	Diff. in 3 Point Pct.	49	Diff. in Opponent Effective FG Pct
22	Diff. in Offensive Rebounds/Gm.	50	Diff. in Opponent True Shooting Pct
23	Diff. in Defensive Rebounds/Gm.	51	Diff. in Opponent Assist Pct.
24	Diff. in Assists/Gm.	52	Diff. in Opponent Steal Pct.
25	Diff. in Steals/Gm.	53	Diff. in Opponent Block Pct.
26	Diff. in Blocks/Gm.	54	Diff. in Opponent Turnover Pct.
27	Diff. in Turnovers/Gm.	55	Diff. in Opponent Assist/TO Ratio
28	Diff. in Personal Fouls/Gm.		

Appendix II. Variables Included in Each of the Twelve Models

Model	Round(s)	Included Variables
1	All	All
2	1 2 3 through 6	All All All
3	All	1 8 9 16 17 22 27 34 37 46 47 53 55

4	1
	2
	3 through 6
		1 2 3 9 10 17 22 33 37 40 46 47 48 53 55
		1 2 5 7 8 9 17 26 40 47 48
		1 3 5 8 9 12 16 34 35 40 44 46 47 48 52

5	All	1 3 4 6 7 8 9 10 12 15 16 17 19 21 22 27 28 31 34 37 39 40 41 42 45 46 47 48 53 54 55
6	1 2 3 through 6	All except 6, 14, 18, 21, 30, 42, 51 All except 6, 15, 21, 28, 29, 31, 45 All except 21, 32, 49, 50, 51
7	All	1 8 9 17 22 27 34 37 46 47 53 55

8	1
	2
	3 through 6
		1 22 33 37 40 46 48 50 53 55
		1 3 4 7 9 12 14 17 20 40
		1 3 5 8 12 16 27 35 40 44 46 47 48 52 54

9	All	1 16 17 27 34 37 47 53 55

10	1
	2
	3 through 6
		1 3 11 17 26 29 36 37 40 46 47 48
		1 7 9 16 17 26 36 37 42 43 47 53
		1 2 8 9 11 12 16 25 31 34 36 38 40 41 47 48

11	All	1 7 10 11 13 15 17 23 24 37 39 40 43 44 48

12	1
	2
	3 through 6
		1 7 14 20 26 34 44 46 48 50 53
		1 5 7 18 23 39 41 43 44 45 53
		1 15 20 23 31 34 44