Lo-Hua Yuan, Anthony Liu, Alec Yeh, Aaron Kaufman, Andrew Reece, Peter Bull, Alex Franks, Sherrie Wang, Dmitri Illushin and Luke Bornn*

A mixture-of-modelers approach to forecasting NCAA tournament outcomes

Abstract: Predicting the outcome of a single sporting event is difficult; predicting all of the outcomes for an entire tournament is a monumental challenge. Despite the difficulties, millions of people compete each year to forecast the outcome of the NCAA men's basketball tournament, which spans 63 games over 3 weeks. Statistical prediction of game outcomes involves a multitude of possible covariates and information sources, large performance variations from game to game, and a scarcity of detailed historical data. In this paper, we present the results of a team of modelers working together to forecast the 2014 NCAA men's basketball tournament. We present not only the methods and data used, but also several novel ideas for post-processing statistical forecasts and decontaminating data sources. In particular, we highlight the difficulties in using publicly available data and suggest techniques for improving their relevance.

Keywords: basketball; data decontamination; forecasting; model ensembles.

DOI 10.1515/jqas-2014-0056

1 Introduction

Predicting the NCAA Men's Division I Basketball Championship, also known as March Madness, has become big business in recent years. In 2011, an estimated $3–12 billion was wagered on the competition (Matuszewski 2011), and in 2014, billionaire Warren Buffet offered $1 billion to anyone who could correctly predict all 63 games. Modeling wins and losses encompasses a number of statistical problems: very little detailed historical championship data on which to train models, a strong propensity to overfit models on post-tournament historical data, and a large systematic error component of predictions arising from highly variable team performance based on potentially unobservable factors.

In this paper, we present a meta-analysis of statistical models developed by data scientists at Harvard University for the 2014 tournament. Motivated by a recent Kaggle competition, we developed models to provide forecasts of the 2014 tournament that minimize log loss between the predicted win probabilities and realized outcomes:

$$\log\text{loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{1} y_{i,j}\ \log p_{i,j} \tag{1}$$

where N is the number of games, $y_{i,j}$ indicates if game i is won by the home (j=0) or away (j=1) team, and $p_{i,j}$ references the corresponding predicted probabilities. Correctly predicting a win with 55% confidence merits a lower score than predicting the same win with 95% confidence; however, incorrectly predicting a win with 95% confidence is penalized much more harshly than incorrectly predicting a win with 55% confidence (Cesa-Bianchi and Lugosi 2001). This contrasts with most common March Madness brackets, in which the winning bracket is that which successfully predicts the most game outcomes.

During the 2014 March Madness period, our group collectively produced more than 30 different models of NCAA basketball performance metrics. The models incorporated a wide variety of team- and player-level archival data, spanning several years of regular and post-season games. Our most successful models for predicting the 2014 out-of-sample results were those which incorporated sufficient regularization, and which did not suffer from data "contamination". Contaminated data, as we define the term here, refers to archival data for a given NCAA season which incorporated the results of the final tournament from that year. Many publicly available NCAA datasets contain contaminated data. Features extracted from these contaminated data sets

***Corresponding author: Luke Bornn,** Harvard University – Statistics, Cambridge, Massachusetts, USA, e-mail: bornn@stat.harvard.edu
Lo-Hua Yuan, Anthony Liu, Alec Yeh, Alex Franks, Sherrie Wang and Dmitri Illushin: Harvard University – Statistics, Cambridge, Massachusetts, USA
Aaron Kaufman: Harvard University – Government, Cambridge, Massachusetts, USA
Andrew Reece: Harvard University – Psychology, Cambridge, Massachusetts, USA
Peter Bull: Harvard University – Institute for Applied Computational Science, Cambridge, Massachusetts, USA

were more likely to overfit models to results from a particular season, and therefore led to poor out-of-sample performance.

In this paper, we present an in-depth look at the mechanisms and successes of our March Madness models, given the same fixed feature set. Specifically, our goal is to compare models and loss functions rather than identify the best features and predictor variables. For completeness, we start in Section 3 with an overview of the complete set of features employed before discussing the models we used to incorporate those features. Out of the dozens of models attempted, the majority were variants of logistic regression, decision trees, and neural networks, and as such we highlight a selection of these models in Section 4. This leads into a discussion of loss functions in Section 5. This section covers issues related to overfitting, as well as issues of predictive adaptability across a range of evaluation metrics. Lastly, we explore the issue of data decontamination in Section 6, in which we consider the use of pre- vs. post-tournament data in model building, and how employing contaminated features can quickly lead to overfitting.

2 Literature review

Contributions to the current literature on NCAA tournament prediction fall into three broad categories, characterized by the type of problem they address.

Problem 1: Seeds already offer good predictive power. Each team enters the tournament with a ranking, or seed, which is not only an indicator of regular season performance, but also serves as a predictor of post-season success, since higher-seeded teams face lower-seeded teams in the early rounds of the tournament (Smith and Schwertman 1999; Harville 2003; Jacobson and King 2009). Seeds are not infallible, however; while strongly predictive for early rounds, their forecasting power deteriorates as the difference in quality between teams in advanced rounds diminishes (Boulier and Stekler 1999). As such, some models attempt to predict where seeding will *go wrong* (Schwertman, McCready, and Howard 1991). This approach generates an upset probability for each game, and then using some threshold λ, for all games with an upset probability $p > \lambda$, the higher-seeded team is predicted to *lose*. While this method has seen some success in recent years (Bryan, Steinke, and Wilkins 2006), there is also evidence that it leads to systematic over-prediction of upsets, ultimately reducing the predictive accuracy of these models to below the seeding baseline itself (McCrea and Hirt 2009).

Problem 2: Success metrics are diverse. The evaluation metric or loss function used to rank tournament predictions can vary widely, ranging from number of games predicted correctly to complicated point structures to log loss functions in which predictions have attached probabilities. Simulations have shown that the choice of loss function can have significant impact on modeling choices (Kaplan and Garstka 2001).

Problem 3: Archival data is a trove of predictive features. This category of the literature, which represents the majority of contributions, focuses on model and variable selection, with an emphasis on mining historical data for predictive features. Iterative computational ranking systems such as those by Sagarin (2014) and Pomeroy (2014) have risen to prominence in recent years. Sagarin, for example, combines winning margin, strength of schedule, and performance against well-ranked teams while incorporating home- or away-game metadata.[1] Sokol's logistic regression Markov chain (LRMC) method first estimates head-to-head differences in team strength, then leverages a Markov chain model to converge upon rankings (Sokol 2014).[2] After computing these features, the LRMC method fits a logistic regression model in which a win or loss is a function of both teams' rankings within each feature set (Carlin 1996; Koenker and Bassett Jr. 2010). Many of these models have enjoyed good predictive success, and have withstood challenges from larger and more complicated models (Toutkoushian 2011).

In this paper, we expand upon the existing literature by experimenting with a large set of features and a rich collection of predictive models. Our findings support the idea that parsimonious feature sets and relatively simple algorithms tend to outperform more complicated models with numerous features.

3 Features

The game of basketball provides numerous statistics for performance evaluation. Evaluation metrics at the college level cover a variety of resolutions – from the individual player to a subset of players to the team to the entire university athletic program. Aggregate statistics, which

1 An update to these ranking schemes weights more recent performance more highly to reflect improvement; see, for example, Chartier Timothy et al. (2011).

2 This model has been updated to replace the MC step with an empirical Bayes model, showing significant improvement (Brown and Sokol 2010).

are generated from combined raw statistics as a means of reducing the dimensionality of a team's overall performance, are popularly used for predicting tournament brackets. For example, the NCAA uses the Ratings Percentage Index (RPI), which combines a team's win percentage, opponents' win percentages, and opponents' opponents' win percentages, to seed teams for the March Madness tournament. Two well-known independent aggregate statistics used in our models are Ken Pomeroy's Pomeroy Ratings (Pomeroy 2014) and Sonny Moore's Computer Power Ratings (Moore 2014). Pomeroy Ratings exist from the 2003–2004 season onward, and are based on Pythagorean expectation (Hamilton 2011), which is a way of estimating a team's expected winning percentage against an average Division I team, accounting for offensive efficiency (points scored per 100 possessions) and defensive efficiency (points allowed per 100 defensive possessions). Moore's Power Ratings, available since 1997, represent the expected score of a given team (with a 3.7 point advantage given to the home team). We culled these aggregate statistics from publicly available online sources. We also obtained NCAA raw statistics from ESPN's archive of men's basketball statistics, which includes archival data going back to the 2001–2002 season.

All our modeling teams had access to the same datasets, but not all teams employed the same features in their models. In particular, almost all teams used the Pomeroy, Moore, and Massey metrics, whereas only a subset of the teams leveraged RPI and the ESPN statistics. We reserve details of individual modeling teams' feature datasets for Section 4. Here, we provide an overview of the data sources we used:

- **Pomeroy** *(9 statistics)* – Ken Pomeroy's statistics are based on offensive and defensive efficiency (i.e. the number of points scored or allowed per 100 possessions adjusted for the strength of competition). He also provides a measure for "strength of schedule" and an index of "luck," represented by the deviation between a team's actual winning percentage and what one would expect from its game-by-game efficiency scores (Pomeroy 2014).
- **Moore** *(1 metric)* – Sonny Moore's Power Ratings (PR) provides a systematic ranking of all the tournament teams, where each team's PR indicates the team's forecast relative performance, if all teams were to play against each other (Moore 2014). A team's PR reflects how the team has performed in previous games and takes into consideration wins and losses, the opposing teams' PRs, and the actual score difference of the games played. The difference between two teams' PRs is equal to the predicted point differential if a game were played on a neutral court (3.7 points would be added for a home team advantage).
- **Massey** *(33 ranking systems)* – Kenneth Massey's website (Massey 2014) provides rankings for the tournament teams based upon a collection of 33 different human and algorithmic ranking systems, including Associated Press, Sagarin Predictor, Moore, Baker Bradley-Terry, and others. Note that these are ordinal rankings rather than specific ratings.
- **ESPN** *(7 statistics total)* – Official NCAA Division I men's basketball regular-season team level summary statistics are provided by ESPN's website (ESPN 2014). The ESPN raw metrics we collected were: points per game, average scoring margin, number of personal fouls per game, turnovers per game, total rebounds per game, offensive efficiency, and defensive efficiency.
- **Rating Percentage Index** – RPI is a quantity used to rank sports teams based on a team's wins and losses, corrected for its strength of schedule. For college basketball, a team's RPI is defined as:

$$\begin{aligned}\text{RPI} = &(\text{Winning Percentage}*0.25) + \\ &(\text{Opponents' Winning Percentage}*0.50) \\ &+ \text{Opponents' Opponents' Winning Percentage}*0.25).\end{aligned}$$

3.1 Contamination

When selecting archival data to feed into predictive models of sporting events, it is critical to ensure that "the past" doesn't contain its own future. Our predictions of the 2014 NCAA tournament results were derived from models in which previous tournament results were a function of covariates from previous years. If those covariates include metrics that incorporate previous tournament outcomes, the input features essentially already "know" the future that they aim to predict. As a result, seemingly successful prediction will, in fact, rely heavily upon anachronous metrics – rendering such models inept for true future prediction. We refer to this phenomenon as *data contamination*.

Contamination can be a serious problem if the data on which one trains a model contains different information from the data on which one predicts future results. For example, running a logistic regression on all features independently reveals that the strongest predictive statistic for how well a team performs is the number of games played (GP) in a given season. All NCAA basketball teams complete 29–31 games in regular season play (MomentumMedia 2006). During March Madness, however, the tournament is single-elimination, so the

two best teams always play the most tournament games (and therefore have the highest overall GP for that year). As such, in tournament predictions prior to 2014, GP proved an excellent, but contaminated, predictor of March Madness performance. Attempting to use 2014 team GP as a predictive feature would necessarily result in poor performance, since by the start of the tournament, all teams had played roughly the same number of games. In this way, inadvertent use of contaminated predictors in archival models can easily lead to disastrous outcomes in actual forecasting.

Other statistics with contaminated information include the Pomeroy Ratings and Moore Power Rankings, two metrics for which the publicly available historical data sets are made of post-tournament rankings. Note, however, that the Moore Power Rankings are available monthly, so pre-tournament iterations are still accessible. After the 2014 tournament, Ken Pomeroy generously provided us with pre-tournament Pomeroy Ratings. As such, in Section 6 we use this gold standard to study the magnitude and effects of contamination.

4 Models

We produced over 30 models to predict win probabilities for all 2278 possible matchups[3] in the 2014 NCAA Men's Division I Basketball tournament. Initial attempts at model development were highly diverse, including pairwise-comparison algorithms such as the Bradley-Terry logit model (Huang, Weng, and Lin 2006) as well as general classification methods like probabilistic support vector machines (Platt 1999). Of the final models, the majority were variants of logistic regression, though boosted tree models and neural networks also achieved some measure of success.

We trained the models on 10 years of historical NCAA tournament results from seasons 2003–2004 through 2012–2013.[4] To guide model development, we used "leave-one-season-out" cross validation. For each left out season, we trained the models on tournament outcomes from the other nine seasons and computed the log loss on the test tournament.

Being limited to 10 years' worth of archival data constrained our choices for modeling algorithms. For example, we found that random forests, implemented using the `randomForest` package in `R` (Liaw and Wiener 2002), performed worse than logistic regression because there wasn't enough data to grow trees of sufficient depth. Although we could obtain numerous descriptive features to use as predictors, the amount of information we had for making playoff predictions was ultimately limited by the number of games for which we had results. Covariate selection was limited by the need to use metrics that extended back 10 years.

Following, we describe our most successful modeling approaches.

4.1 Logistic regression

The most successful approach across multiple evaluation metrics was logistic regression. Logistic regression has many attractive properties for the March Madness challenge: it naturally produces win probabilities, provides an intuitive interpretation of fitted values in terms of odds ratios, and is readily implemented in standard software packages. Its success is also unsurprising, given that log loss was the performance measure and that we trained our models on a relatively small number of tournament games (a total of 10 seasons, 63 games each). The logistic regression model assumes a binomial distribution for a binary response alongside a logit link function. Let n index the number of Bernoulli trials that constitute a particular binomial observation and N index the number of observations. Let y_i, i=1, ..., N, be the proportion of "successes" out of n_i independent Bernoulli trials, so $n_i y_i \sim Bin(n_i, \pi_i)$, with $E[y_i] = \pi_i$ independent of n_i. Let x_{ij}, i=1, ..., N, j=1, ..., p, be the j^{th} predictor for observation i. The logistic regression model is then expressed as

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \sum_{j=1}^{p} \beta_j x_{ij} \quad or \quad \pi_i = \frac{\exp\left(\sum_{j=1}^{p} \beta_j x_{ij}\right)}{1+\exp\left(\sum_{j=1}^{p} \beta_j x_{ij}\right)} \tag{2}$$

Considering the large number of covariates and relatively few data points, overfitting and multicollinearity were notable concerns. The majority of our logistic regression models employed regularized regression to address this problem, with common choices including L1 (LASSO) and L2 (ridge) regularization (Demir-Kavuk et al. 2011), as well as stepwise subset selection. In each of our logistic regressions, we standardized all the feature values by season prior to model fitting.

3 With 68 teams in contention at the start of the tournament, there are $\binom{68}{2} = 2278$ total possible pairings.

4 Our choice to limit archival data input to the past 10 years was due to the Pomeroy data only including the most recent 10 NCAA seasons.

For all the models we describe below, we used the difference of competing teams' (standardized) metrics as model predictors. All teams were randomly assigned unique integer IDs, and the response was defined as whether or not the team with the smaller ID won the match.

4.1.1 Model A: Logistic regression with backward elimination

We fitted a logistic regression model, including all covariates, to the training data. Covariates with significance levels < 0.1 were then iteratively pruned to produce the final fitted model. Here, we used Pomeroy, Moore, Massey, RPI, and ESPN features.

4.1.2 Model B: Logistic regression with L2 regularization

L2-regularized regression, or ridge regression, minimizes the residual sum of squares under a constraint on the sum of the squared coefficients:

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \sum_{i=1}^{N} -\log p(y_i | \mathbf{x}; \boldsymbol{\beta}) \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq \lambda \quad (3)$$

The tuning parameter λ controls the amount of shrinkage applied to the parameter estimates. The optimal value of λ is chosen by 10-fold cross validation over the training set, i.e., the tournament games of all preceding seasons, back to year 2003. In particular, the largest value of λ within 1 standard error of the minimum is used, so as to prevent overfitting. An advantage of ridge regression over sequential variable selection (as in model A) is stability: whether a predictor is included or excluded by a stepwise variable selection method often comes down to very slight variations in the data. As ridge regression is a continuous process that shrinks coefficients towards zero, it is less sensitive to the natural variability of the data than sequential selection approaches. Here, we used Pomeroy, Moore, Massey, and RPI features.

4.1.3 Model C: Logistic regression with L1 regularization

L1-regularized regression, commonly known as the LASSO (Tibshirani 1996), combines the stability of ridge regression with the easy interpretability offered by stepwise variable selection. LASSO minimizes the residual sum of squares under a constraint on the sum of the absolute value of the coefficients:

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin} \sum_{i=1}^{N} -\log p(y_i | \mathbf{x}; \boldsymbol{\beta}) \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq C \quad (4)$$

Like λ in the case of ridge regression, C controls the amount of shrinkage applied to the LASSO parameter estimates. We chose the optimal value of C by performing 10-fold cross validation over the training set, i.e., the tournament games of all preceding seasons, back to year 2003. In particular, we used the largest value of C within 1 standard error of the minimum, to prevent overfitting. Unlike ridge regression, LASSO encourages parameter sparsity. We used Pomeroy, Moore, and Massey metrics as input features for this model.

The resulting features used in each implementation of the logistic regression models are summarized in Table 1. The numbers under each data source indicate the total number of features from that source that was included in the final fitted model; that is, after variable selection was completed.

4.2 Model D: Stochastic gradient boosting

Boosting is an iterative algorithm that takes a weighted average of simple classification rules with mediocre misclassification error-rate performance (known as base learners) to produce an overall highly accurate classification rule (Friedman 2001). The method originates from Probably Approximately Correct (PAC) computational learning theory, which states that when appropriately combined, classifiers that individually exhibit a performance slightly better than random guessing can perform very well. Boosting trains the base learners and determines their weights by relying on a gradient descent search, iteratively adding basis functions in a greedy fashion so that each additional basis function further reduces the selected loss function.

Table 1: Summary of logistic-based models' feature categories. The numbers under each feature category name indicate the total number of metrics from that source that was included in the final fitted model.

Model	Feature category				
	Massey	Pomeroy	ESPN	Moore	RPI
A	9	3	1	Yes	No
B	12	9	0	Yes	Yes
C	9	8	0	Yes	No

Stochastic gradient boosting (SGB) is a modification of boosting wherein at each iteration of the gradient descent search, a base learner is fit on a stochastic subsample of the training set. Using a random subsample of the training set acts as a kind of regularization to help prevent overfitting, and also reduces computation time. This method was implemented using the `gbm` package (Ridgeway 2007) in `R`.

Stochastic gradient boosting can be implemented with different loss functions. In keeping with the primary assessment metric for the March Madness Kaggle competition, we used log loss. We performed 10-fold cross-validation on the training set of previous tournament seasons 2003–2004 through 2012–2013 to optimize tuning parameters, resulting in a SGB model for season 2014 parameterized by 10,000 trees, a shrinkage parameter of 0.0005, and an interaction depth of 5. Pomeroy, Moore, and Massey features were used as inputs to the model.

4.3 Model E: Neural networks

Neural networks are a biologically inspired class of nonlinear learning algorithms (Cochocki and Unbehauen 1993) that have been implemented in many circumstances, notably the `neuralnet` package in `R` (Fritsch, Guenther, and Guenther 2012). Each vertex, or *neuron*, receives an input x_i from each incoming edge, and sends an output $f(\mathbf{x}) = f(x_1, ..., x_n)$ along each outgoing edge, where

$$f(\mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})$$

for weights $\mathbf{w} = (w_1, ..., w_n)$ and activation function g: $\mathbb{R} \to \mathbb{R}$. In other words, each neuron computes a weighted sum of its inputs, and then passes that sum through an activation function. The simplest neural networks are feed-forward neural networks, which structure their neurons into three sections of layers: an input layer, which receives the input variables; an output layer, which computes the predictions; and any number of hidden layers in between. When all nodes in each layer are connected to all nodes in the following layer, the network is "fully connected". The weights on a neural network are learned by minimizing a cost function such as squared error or log loss, often through gradient descent. The main algorithm for performing gradient descent on the weights of a neural network is called backpropagation, described in detail in Hastie, Tibshriani, and Friedman (2009).

The neural network used to predict the 2014 NCAA Tournament victory probabilities was a fully connected feed-forward neural network. The number of hidden layers, the number of nodes in each hidden layer, as well as the type of activation function at each node (logistic vs. sigmoid) were determined via cross-validation. Specifically, we used 10-fold cross validation to select these model features, ultimately choosing the model using the fewest nodes within one standard deviation of the minimum error. For the final model, we chose a neural network with a single 5-node hidden layer and a single-node output layer. Every node used a logistic activation function, and model performance was evaluated by log loss. Training continued until the partial derivative of the log loss was <4 (a threshold also chosen by cross validation).

A variant of the backpropagation algorithm known as resilient backpropagation with weight backtracking (RPROP+) was used to perform the gradient descent (Riedmiller and Braun 1993). The inputs to the network were 12 Massey ordinal rankings, 4 Pomeroy rankings, and the Moore metric.

4.4 Ensembles

In ensemble learning methods, predictions from multiple learning algorithms ("weak learners") are combined together ("stacked") in some weighted fashion by an umbrella regression or classification model ("meta learner") in order to produce final predictions. Following is a discussion of four ensemble models built after the NCAA 2014 tournament took place, combining the results of the best-performing models produced prior to the tournament. All four ensembles were produced using logistic regression as the meta learner. That is, we first trained our logistic regression meta learner on individual weak learners' predictions for five historical tournament seasons (seasons 2008–2009 through 2012–2013). After determining the weights for each weak learner, we then fed the weak learners' 2014 March Madness predictions into the ensemble model to generate stacked predictions for the 2014 season.

Our largest ensemble stacked all five models presented above: three logistic regression models (models A, B, C), the stochastic gradient boosted tree model (model D), and the neural network (model E). A second ensemble stacked the three logistic regression models (A, B, and C) to see how much additional predictive accuracy we could obtain by combining results from slight variations in logistic regressions. Models B and D were stacked in a third ensemble and models B and E were stacked in a fourth ensemble to see how the best performing logistic regression model stacked with each of the non-logistic regression models would perform. Out of the four ensembles, the

BD ensemble gave the smallest log loss for season 2014. A reason for this may be that models B and D capture very different predictive qualities of the metrics. Indeed, the predicted victory probabilities produced by models B and D are essentially uncorrelated (0.02). In contrast, the correlations between predictions produced by models A, B, C, and E range between 0.72 and 0.96. All the stacked algorithms' log loss performances are summarized in Table 2, along with log loss values for the individual models and several benchmark log losses obtained from the Kaggle website.

Table 2: 2014 NCAA tournament log losses for logistic regression models (A, B, C), stochastic gradient boosted tree model (D), neural network (E), various stacked algorithms, and two naive benchmarks. The average Kaggle log loss score was 0.58.

Model	Log loss
A	0.67
B	0.61
C	0.71
D	0.98
E	0.84
Stacked (A, B, C, D, E)	0.77
Stacked (A, B, C)	0.75
Stacked (B, D)	0.63
Stacked (B, E)	0.86
All 0.5 benchmark	0.69
0.5+0.03* (Seed difference) benchmark	0.60
Mean Kaggle score	0.58

It is interesting to note that the ABC ensemble resulted in a log loss that is higher than that of its individual constituents. Stacking algorithms are generally expected to perform at least as well as the individual weak learners that comprise it, but we did not find this to be the case in our ensembles of models A through E. While model overfitting may have been the culprit, another possibility is that some of the features we used to train our models were "contaminated", in the sense that they incorporated post-tournament information. We further address this hypothesis in Section 6, where we find evidence that many of our metrics were indeed contaminated, leading to poor predictive performance in the 2014 tournament.

A series of heatmaps (Figures 1 and 2) provide a visual representation of the various models' performance for each round of the NCAA tournament. We graphed the absolute value of the actual outcome of each game from the tournament minus the victory probability we predicted. For example, suppose in a game between team 1 and team 2, we predicted a probability of 0.94 that team 1 would win, and team 1 in fact did win. The *distance of prediction from observed* for this game would then be $|1-0.94|=0.06$. On the other hand, if team 2 won, the resulting *distance of prediction from observed* would be $|0-0.94|=0.94$. Note that the smallest distance possible is 0, while the largest possible distance is 1. The winners are always listed first in the pairing.

Though the heatmaps do not report log loss, they do let us visualize both correlations and successes/failures

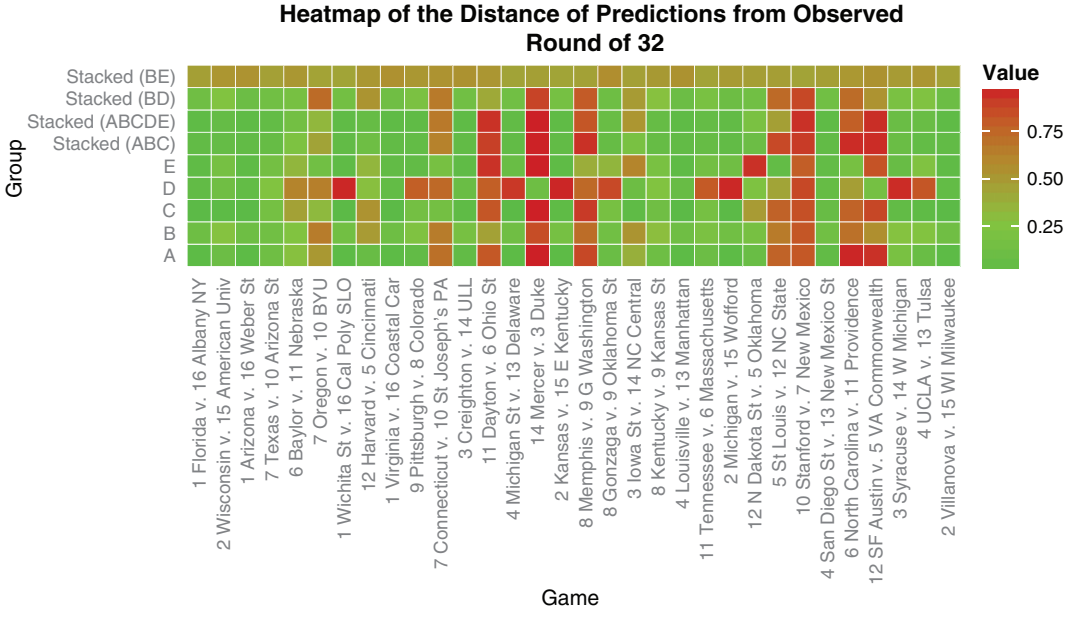

Figure 1: Heatmap comparing various models' predictions across the round of 32 of the 2014 NCAA tournament.

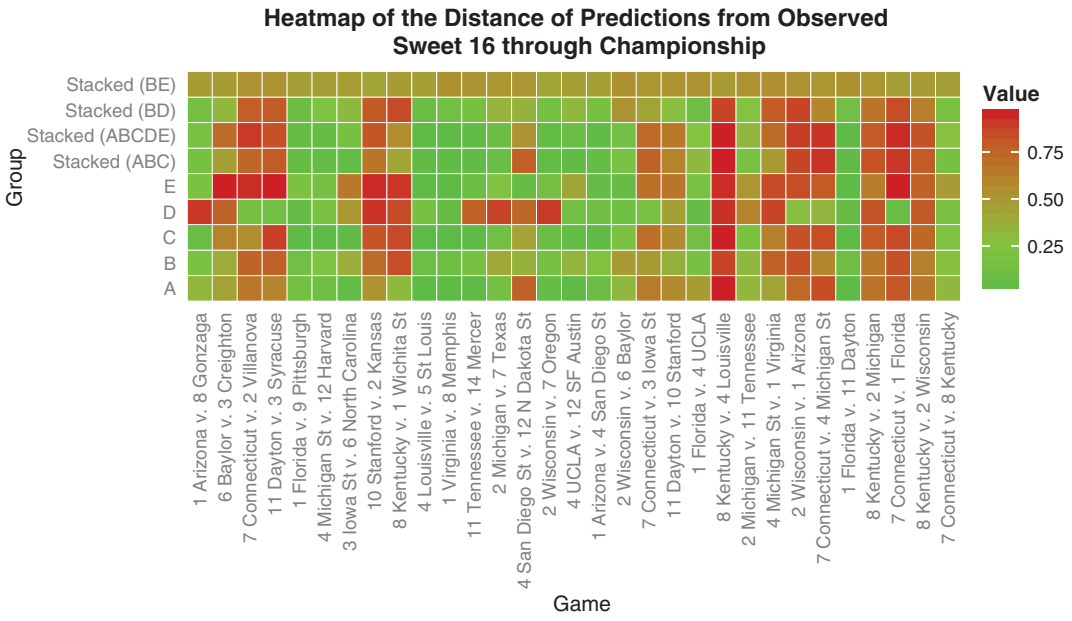

Figure 2: Heatmaps comparing various models' predictions across the round of 16, round of 8, round of 4, semifinals, and finals of the 2014 NCAA tournament.

of our models on *specific* games. Together with the log losses provided in Table 2, we were able to examine our models' specific strengths and weaknesses. We first noted the strong correlations between predictions from models A, B, C and predictions from ensemble models ABC and ABCDE. This makes intuitive sense since the stacked predictions are composites of the individual A, B, and C models. The strong correlation between models A and C can be explained by the close relationship between a simple logistic regression and a logistic regression with L1 regularization. Model B (another logistic regression, but with L2 regularization) is, unsurprisingly, also strongly correlated with models A and C. However, model B outperforms the other two – a phenomenon illustrated in the heatmaps: model B, while often predicting the same outcomes as models A and C, has lighter shades of red. On the other hand, model B also tends to have darker shades of green. This indicates that model B generally has less confident predictions than models A or C. As discussed later and shown in Figure 3, under a log loss criterion, the penalty of an overly confident incorrect answer is severe – more extreme than the reward of having a confident correct answer. The relatively low confidence of model B predictions visualized in the heatmaps thus corresponds

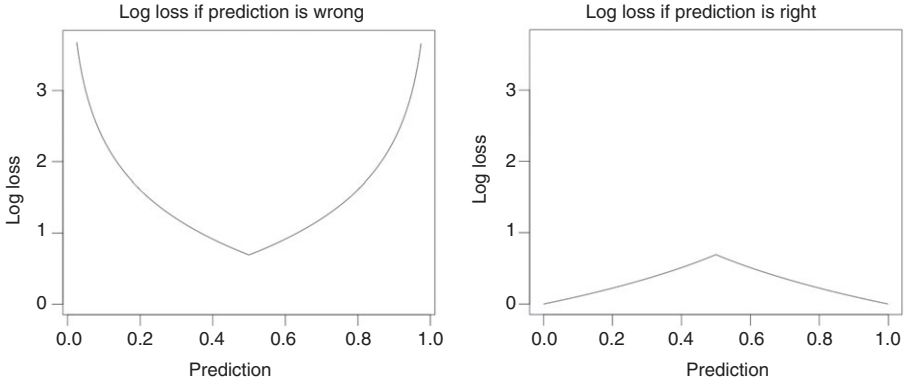

Figure 3: Log loss as a function of the predicted probability. Note that when the prediction is wrong, log loss approaches infinity at both 0 and 1.

to superior log loss performance of the model, compared to the other models we implemented. That said, models that give timid predictions across the board do not perform well, either, as illustrated by the BE ensemble, whose majority of predictions fall within the range of 0.35 and 0.65.

Although models D and E did not perform as well as the logistic regression variants, we expected them to provide alternate predictive strengths not reflected in models A, B, and C. Examination of Figures 1 and 2 suggested that a complementary advantage model D provides is the correct prediction of upsets. In the world of NCAA March Madness, upsets are often the most exciting games, yet also the most difficult to predict. Upsets in the round of 32 like No. 14 Mercer beating No. 3 Duke or No. 12 SF Austin beating No. 5 VA Commonwealth (Figure 1), or upsets in the later games like No. 11 Dayton beating No. 3 Syracuse or No. 7 Connecticut beating No. 1 Florida, could be detrimental to our models' predictive performance. Interestingly, it is in these upsets where model D stands out as being the sole model that predicts the correct outcome. Even though model D was the worst performing model overall, it managed to predict upsets that the other models failed to predict. Therefore, when stacked with model B (the best performing individual model) the resulting BD ensemble ended up having a markedly smaller log loss than the other ensemble methods we investigated.

Model E, on the other hand, had the ability to make correct predictions for games where our other models heavily favored the underdog. This was especially noticeable in two games: No. 5 St. Louis vs. No. 12 NC State and No. 6 North Carolina vs. No. 11 Providence (Figure 1). In both games, the higher-ranked team won. However, our logistic regression models clearly favored the underdog. Why? A closer look at historic data revealed that St. Louis and Providence had very few appearances in previous tournaments. St. Louis, for example, had only two previous appearances in the tournament. Providence only had one. Logistic regression models are relatively sensitive to the number of observations used to estimate model parameters. A lack of sufficient data can result in huge errors and wildly inaccurate point estimates. This might have been why models A, B, and C failed to make the correct predictions for these two games. The lack of historic data for St. Louis and Providence resulted in logistic models that sported both inaccurate and imprecise point estimates. In contrast, models D and E performed quite well in predicting the outcome of these two games. Both models slightly favored St. Louis and North Carolina, both of whom ended up being the winner. Since models D and E are flexible nonlinear models, they rely less on trying to fit these teams with few historical observations into an oversimplified parametric model, and more on using observed empirical results.

Lastly, by focusing on the reddest columns of our heatmaps, we realized that our models' main failure was in predicting the two championship teams' progression to the final match. At the start of the tournament, Connecticut and Kentucky were neither highly ranked nor lowly ranked, yet both teams were incredibly successful in their bid for the championship. In one respect, one could say these two teams had a string of lucky upsets. A more insightful conclusion, however, would be to posit that our models fail to capture important qualitative effects specific to the tournament. First, regular season and playoff games are markedly different, yet our models ultimately treat them similarly. Our models do not reflect the fact that while Connecticut and Kentucky had moderate seasons, historically during tournament play, these two teams step up their game entirely. Our models also fail to capture effects such as motivation. Both basketball teams, with huge fan bases rallying behind them and intense sports coverage online, may have held an advantage by being more highly motivated teams. Furthermore, these two teams appeared in almost all ten seasons of our data (Connecticut in eight and Kentucky in nine). In other words, they had been successful every year in making the playoffs and winning titles (Connecticut won in 2004 and 2011; Kentucky won in 2012). As a result, these teams had more experience playing in the tournament. Not only did the players have more experience, but the coaches were also more seasoned leaders. Our models' inability to see that Connecticut and Kentucky had so much success in the past resulted in treating them as only mediocre teams, rather than the champion teams that they actually were.

5 Loss functions

We used the log loss metric to compare performance among models; however, several other cost functions can be applied to assess predictive accuracy in the context of tournament forecasting. In this section, we explore some of the properties of log loss and consider our models' relative performance across other cost functions.

5.1 Log loss

Log loss (Equation 1) is a convenient metric, in part because of its connection to the Kelly betting strategy. Specifically, the Kelly criterion says that proportional

gambling is log-optimal, meaning that a gambler should bet exactly $p_j^\star$ percent of her wealth on outcome j if the chance of outcome j is $p_j^\star$. This allocation strategy maximizes the bettor's doubling rate at $\sum p_j^\star \log p_j^\star + C$ where C is a function of the payoffs, but not the bet allocations (Cover and Thomas 2012).

By the law of large numbers, in repeated bets i=1, ..., N of the same game, the log loss converges:

$$\log\text{loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{1} y_{i,j}\log p_{i,j} \to -\sum_{j=0}^{1} p_j^\star \log p_j$$

Thus, the log loss function approximates the long run doubling rate given betting strategy p_j, under true outcome probabilities $p_j^\star$. The difference between the doubling rates of the optimal strategy and chosen strategy, $\sum p_j^\star \log p_j^\star - \sum p_j^\star \log p_j$, is minimized at zero when $p_j^\star = p_j$.

This loss function highly penalizes confident-but-incorrect predictions, even for a single observation, since the log of an incorrect prediction diverges to $-\infty$ (i.e. the bettor loses all of her wealth). Figure 3 shows how the loss function performs for correct versus incorrect predictions. The penalty for a confident but incorrect prediction is much larger than that for an incorrect, unconfident prediction. A model's goal is to minimize the value of the evaluating cost function. Therefore, we aimed to be confident in our correct answers and unconfident in our incorrect ones.

5.2 Fencing log loss

The log loss penalty grows exponentially as an incorrect prediction approaches perfect confidence. A small handful of highly confident incorrect predictions can have an enormous negative impact on overall loss minimization. Accordingly, tournament games that resulted in unexpected upsets have an outsized effect on the total log loss score. One approach to address this is to make the predictions systematically less confident. Consider a set of predictions with many win probabilities around 0.99 and others close to 0.01. Being wrong about any of these extreme votes is very costly (and very likely given the frequency of upsets in NCAA tournaments), so we might consider fencing in our predictions. This entails establishing an artificial limit at a given distance from absolute confidence (i.e. zero or one) in both directions. All predictions which fall outside this range are then mapped onto the extrema of the new, fenced-in range. With a fence of 0.1, a probability of 0.97 would become 0.9 and a probability of 0.08 would become 0.1. Figure 4 shows the results of fencing on the log loss of each of the models' predictions.

While we found that fencing does help reduce log loss, it is only effective among those models which performed poorly in the first place. Model B, our best performing model, gets successively worse as the predictions are shrunk towards 0.5. Models D and E, which performed the worst, improve rapidly as the predictions approach

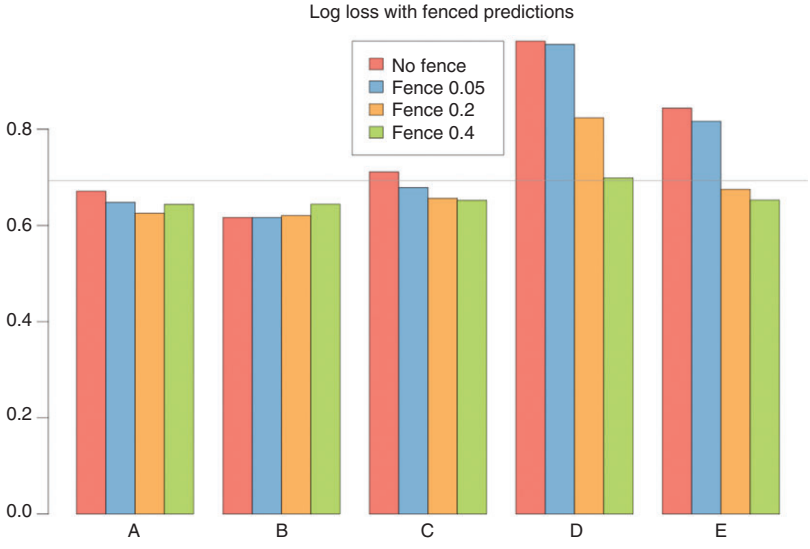

Figure 4: The result of fencing each model different distances. The grey line is a predictive baseline, indicating the log loss if all predictions are set to 0.5. A lower log loss is a better score.

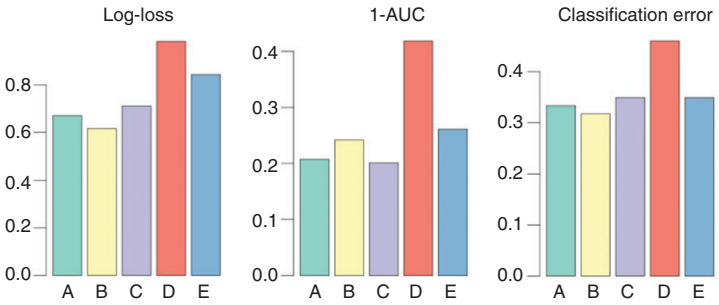

Figure 5: Model performance under three metrics: 1) log loss, 2) 1–AUC, 3) classification error.

0.5. More specifically, fencing the log loss has the effect of increasing expected loss (at least in the case of logistic regression), but because it prevents a major accumulation of loss after an upset, it therefore controls the variance of the observed loss.

5.3 Performance on other metrics

In addition to log loss, we considered both classification error and area under the curve (AUC). To calculate classification error, we set a decision boundary at a probability of 0.5. In Figure 5, rather than display AUC, we used the quantity 1–AUC in order to compare the three cost functions we wished to minimize.

Interestingly, we found that the choice of loss function does not have a large impact on model assessment. Given the small size of the test set – 63 games to predict – and only a handful of models, all of which predict roughly two-thirds of game outcomes correctly, changing the cost function does not drastically change our interpretation of which models are most effective.

6 Data decontamination

Data contamination, wherein predictive feature values incorporate post-tournament results, seriously hindered our models' success. A post-hoc analysis of our models allowed us to investigate the effects of this data contamination and develop methods to ameliorate it. After the conclusion of the NCAA tournament, we obtained pre-tournament Pomeroy ratings (4 total) and pre-tournament Massey system rankings (13 systems total), and attempted to recover the prediction accuracy we would have obtained had we used uncontaminated pre-tournament data. This was accomplished through two "data decontamination" procedures in which we regress 1) post-tournament features, or 2) the difference between pre- and post-tournament features, on the round in which a team was eliminated from the tournament, thereby controlling for tournament performance.

6.1 Pre-tournament vs. post-tournament features

It is reasonable to expect that team-level metrics obtained prior to the NCAA tournament differ from those obtained after the tournament, reflecting the difference between average team performance during the tournament and average team performance during the season. Simply put, a team's average statistic during the tournament will likely not be the same as that team's average statistic during the season. As an illustration, we provide box plots showcasing pre- and post-tournament values for two predictors used in most of our models: Pyth,[5] a Pomeroy metric, and MOR, a Massey ranking. Figures 6 and 7 show the pre- and post-tournament Pyth and MOR values of the 346 teams that played in the 2013 season, plotted against the round in which a team lost and was eliminated from the tournament. It appears that teams eliminated in the first round had, on average, better scores on these metrics before the tournament than after, reflecting poor tournament performance. The difference between pre- and post-tournament scores is larger for teams that perform better in the tournament and thus get eliminated in later rounds. We call predictors that follow this pattern "contaminated". Including contaminated features in models designed to predict tournament performance reduces predictive accuracy through overfitting.

To better quantify the impact of using contaminated features for tournament predictions, we re-ran several

5 The Pyth metric is a continuous Pomeroy statistic that estimates a team's expected winning percentage against an average Division-I team.

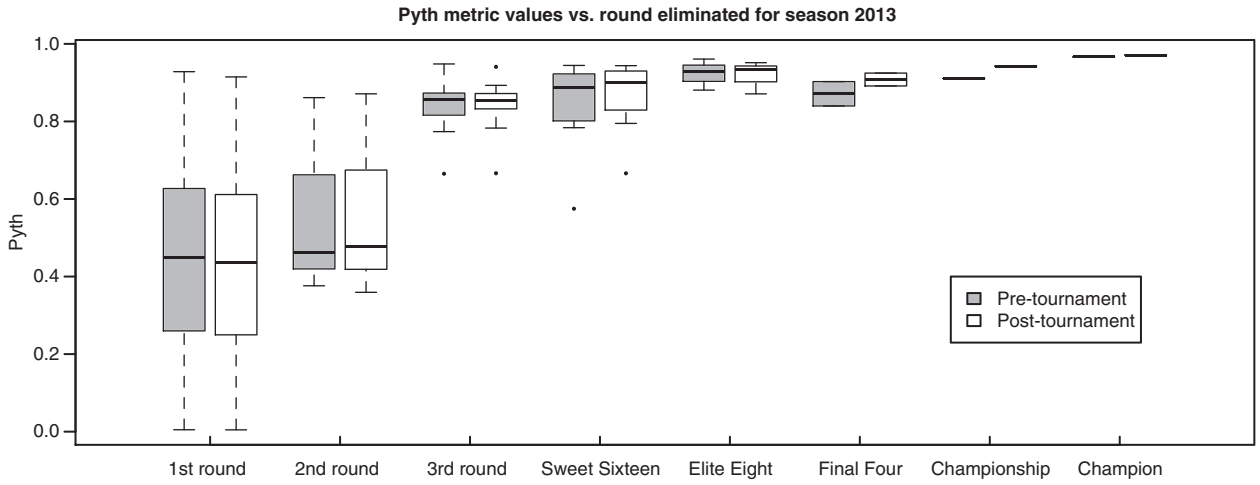

Figure 6: Pomeroy's Pyth metric for the 346 teams that played in the 2013 season, plotted against the round in which each team lost a match and was eliminated from the NCAA tournament. The difference between pre- and post-tournament Pyth distributions reflects the degree to which the post-tournament version of the Pyth scores incorporates tournament performance. For example, the average pre-tournament Pyth value for the two teams that lost in the Final Four is 0.87 while the average post-tournament Pyth value for the two teams that lost in the Final Four is 0.91.

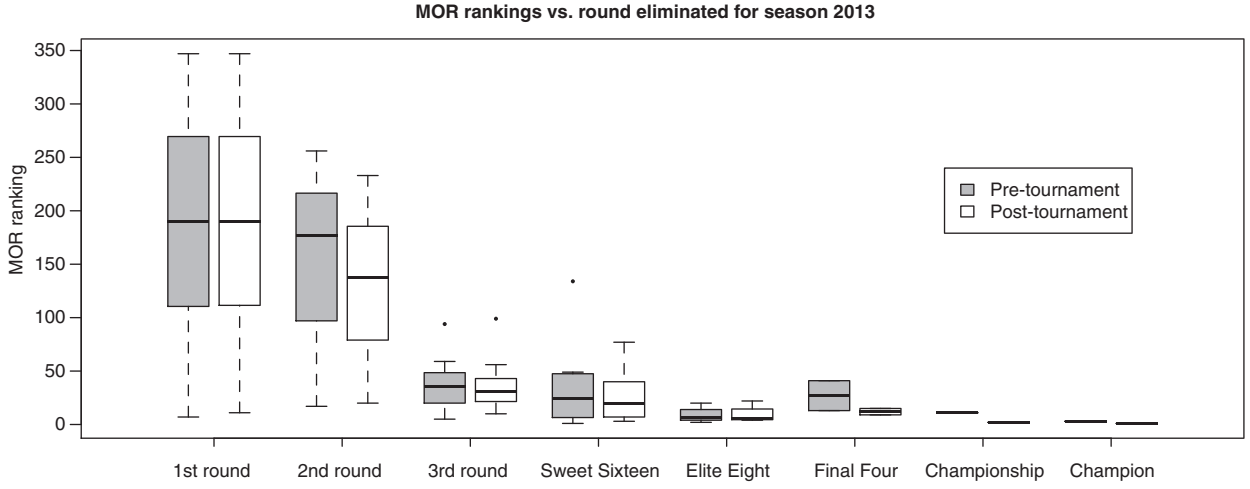

Figure 7: MOR ordinal rankings for the 346 teams that played in season 2013, plotted against the round in which each team lost a match and was eliminated from the NCAA tournament. The difference between pre- and post-tournament MOR distributions reflects the degree to which the post-tournament version of the MOR scores incorporates tournament performance. For example, the average pre-tournament MOR ranking for the two teams that lost in the Final Four is 27 while the average post-tournament MOR ranking for the two teams that lost in the Final Four is 12.

of our models using only pre-tournament Pomeroy and Massey metrics, then compared the resulting log losses to the same models fitted with the post-tournament Pomeroy and Massey statistics. To make predictions for seasons 2009–2013, we trained our models on (either pre- or post-tournament) data from seasons 2004–2013, excluding the season we aimed to predict. The log losses reported for season 2014 were generated from models trained on data from seasons 2004–2013. In the following section, we discuss ways to mitigate the negative effect of training on post-tournament data.

6.2 Decontaminating post-tournament data

To reduce the impact of post-tournament contamination, we regressed each team-level post-tournament metric which we believed to be contaminated, on the round in

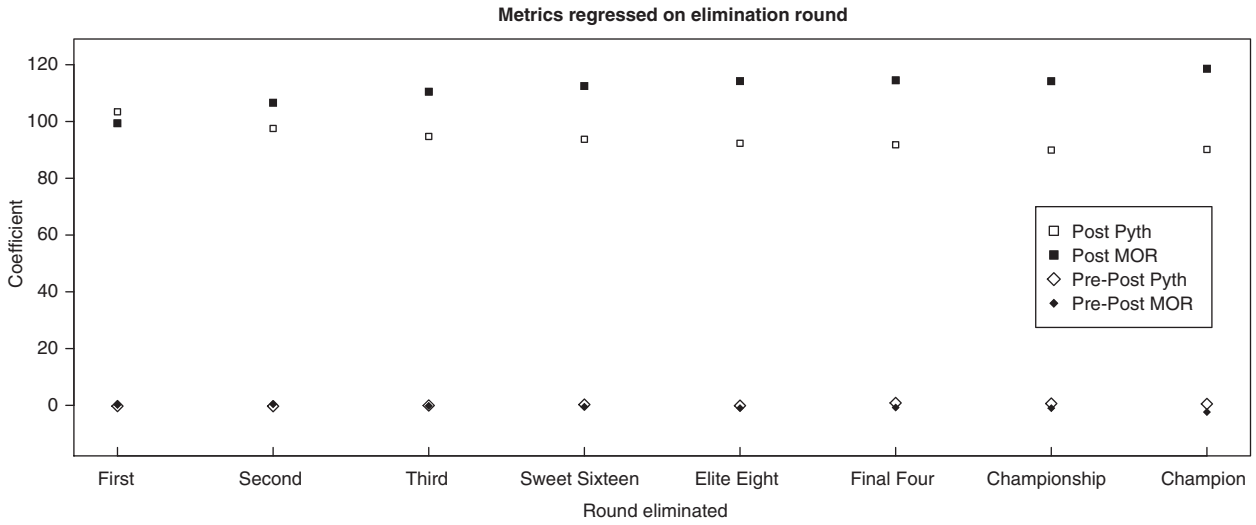

Figure 8: Elimination round slope coefficients for Pyth and MOR metrics, obtained via Model I (Post) vs. Model II (Pre-Post).

which that team was eliminated. This allowed us to control for the information in the contaminated metric reflecting a team's tournament success. We called this regression method Model I. We hypothesized that substituting the residual errors from Model I regressions as "decontaminated metrics" in place of the original contaminated features might lead to lower log loss.

A second way to remove the effects of contamination is to first regress the *difference* between pre- and post-tournament metric values for each team, for example $\text{Pyth}_{\text{pre}} - \text{Pyth}_{\text{post}}$, on the round in which each team was eliminated. We called this regression Model II. We then substituted the original 2004–2013 seasons' post-tournament features with {Post-tournament metric values + predictions from Model II} in order to train our classification models and make season 2014 predictions. Figure 8 illustrates how elimination round regression coefficients differ when metrics are decontaminated using either Model I or Model II. Note that Model II, the "Pre-Post" method, successfully controls for tournament performance, as the coefficient value is uncorrelated with elimination round.

The results are summarized in Table 3.[6] Using post-tournament features (from seasons 2004–2013) to train our models and make predictions for seasons 2009–2013 generally improved the log losses for those seasons, especially among logistic regression models. However, log losses for the 2014 season were much worse when we trained our models using post-tournament rather than pre-tournament features.

Table 3 also summarizes log loss results when decontaminated data was used in place of the original post-tournament metrics. Using Model II to obtain decontaminated data generally produced lower log loss than using Model I. For seasons 2009–2013, using decontaminated data did not produce better log loss than using the original contaminated data, but for season 2014, the log loss for predictions derived from decontamination Model II was lower than the contaminated log losses, and closer to the log losses we would have obtained using only pre-tournament metrics.

7 Conclusions

In this paper, we presented a suite of models for predicting the outcome of the 2014 NCAA Men's Basketball tournament. We intend this work to serve as an empirical evaluation of available forecasting methods for tournament basketball, as well as highlight some of the challenges that inevitably arise with limited, and potentially biased, data.

We analyzed model performance on a number of different cost functions, with log loss as the primary metric. Across a wide range of modeling efforts, we found that logistic regression, stochastic gradient boosting, and neural network algorithms provided the best performance. We also experimented with a number of ensembling approaches, but discovered that they did not

6 Since we obtained pre-tournament data only from Pomeroy and Massey, and only for years 2004–2010 and 2012–2014, these were the only two data sources and the only years we considered in our data decontamination analyses. In this section of the paper, when we reference a time span such as 2004–2013, it is with the understanding that year 2011 data is excluded.

Table 3: In the first four rows: Log losses for winning probabilities predicted by logistic regression (model A), stochastic gradient boosting (SGB) (model D), and neural network (model E), trained on pre-tournament data (Pre), post-tournament data (Post), or post-tournament data corrected by decontamination Model I or II (I or II). Last row: Log losses of predicted winning probabilities for season 2014, based on models trained on Pre, Post, Method I decontaminated, and Method II decontamined tournament data from seasons 2004–2013. Input features for season 2014 predictions were pre-tournament metrics.

Season	Logistic regression				SGB				Neural network			
	Pre	Post	I	II	Pre	Post	I	II	Pre	Post	I	II
2009	0.51	0.36	0.72	0.43	0.51	0.40	0.66	0.42	0.53	0.49	0.80	0.58
2010	0.57	0.36	0.74	0.41	0.55	0.23	0.56	0.42	0.55	0.29	0.84	0.57
2012	0.57	0.27	0.70	0.45	0.59	0.40	0.43	0.36	0.58	0.28	0.53	0.46
2013	0.62	0.19	0.74	0.66	0.61	0.45	0.61	0.45	0.61	0.41	0.78	0.47
2014	0.61	1.06	1.50	1.03	0.63	0.89	0.81	0.75	0.66	0.84	0.78	0.74

outperform our individual models, perhaps due to overfitting and data contamination.

Unfortunately, most of our models did not perform better than an all 0.5 baseline prediction. This is unsettling, and urges us to ask why our models didn't perform as well as a naive model. Perhaps we should have defined features differently (for example, used individual team metrics as input features, rather than the difference of competing teams' feature values as inputs). Perhaps we should have used simple but tried-and-proven metrics like seed difference. Furthermore, some of our models used regular season game outcomes to predict 2014 playoff results, which – in light of anecdotal evidence that predicting regular season games is fundamentally different from predicting tournament playoffs – may have hurt us.

In addition, our selection of learning algorithms was limited by the quality of data available to us. In particular, many powerful algorithms, including random forests and some forms of neural networks, need a lot of data to train properly. With only a few dozen games per tournament season, and going back only 10 years, the size of our data restricted our modeling choices. Many of our models also encountered significant overfitting issues due to the contamination of predictive features with post-tournament results. Attempts to decontaminate post-tournament data suggest that this bias can be partially corrected for, though decontaminated models still underperformed against models which only employed "pure" pre-tournament data.

We conclude with four recommendations for statisticians interested in predictive modeling of the NCAA March Madness tournament: pay careful attention to the issue of feature contamination and feature selection, choose modeling approaches befitting the quality and quantity of available training data, customize algorithm selection based on target loss function, don't cast aside simple models, as they may very well outperform complex models that are sensitive to many tuning parameters. Our findings suggest that the adoption of these recommendations will yield superior performance for future modeling efforts.

References

Boulier, Bryan L. and Herman O. Stekler. 1999. "Are Sports Seedings Good Predictors?: An Evaluation." *International Journal of Forecasting* 15(1):83–91.

Brown, Mark and Joel Sokol. 2010. "An Improved LRMC Method for NCAA Basketball Prediction." *Journal of Quantitative Analysis in Sports* 6(3):1–23.

Bryan, Kevin, Michael Steinke, and Nick Wilkins. 2006. Upset Special: Are March Madness Upsets Predictable? Available at SSRN 899702.

Carlin, Bradley P. 1996. "Improved NCAA Basketball Tournament Modeling via Point Spread and Team Strength Information." *The American Statistician* 50(1):39–43.

Cesa-Bianchi, Nicolo and Gabor Lugosi. 2001. "Worst-Case Bounds for the Logarithmic Loss of Predictors." *Machine Learning* 43(3):247–264.

Cochocki, A. and Rolf Unbehauen. 1993. *Neural Networks for Optimization and Signal Processing*. 1st ed. New York, NY, USA: John Wiley & Sons, Inc., ISBN 0471930105.

Cover, Thomas M. and Joy A Thomas. 2012. *Elements of Information Theory*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Demir-Kavuk, Ozgur, Mayumi Kamada, Tatsuya Akutsu, and Ernst-Walter Knapp. 2011. "Prediction using Step-wise L1, L2 Regularization and Feature Selection for Small Data Sets with Large Number of Features." *BMC Bioinformatics* 12:412.

ESPN. 2014. *NCAA Division I Men's Basketball Statistics – 2013–14, 2014*. (http://kenpom.com/index.php?s=RankAdjOE). Accessed on February 22, 2014 and March 28, 2014.

Friedman, J. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics* 2:1189–1232.

Fritsch, Stefan, Frauke Guenther, and Maintainer Frauke Guenther. 2012. "Package 'Neuralnet'." *Training of Neural Network* (1.32).

Hamilton, Howard H. 2011. "An Extension of the Pythagorean Expectation for Association Football." *Journal of Quantitative Analysis in Sports* 7(2). DOI: 10.2202/1559-0410.1335.

Harville, David A. 2003. "The Selection or Seeding of College Basketball or Football Teams for Postseason Competition." *Journal of the American Statistical Association* 98(461):17–27.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer.

Huang, Tzu-Kuo, Ruby C. Weng, and Chih-Jen Lin. 2006. "Generalized Bradley-Terry Models and Multi-Class Probability Estimates." *Journal of Machine Learning Research* 7(1):85–115.

Jacobson, Sheldon H. and Douglas M. King. 2009. "Seeding in the NCAA Men's Basketball Tournament: When is a Higher Seed Better?" *Journal of Gambling Business and Economics* 3(2):63.

Kaplan, Edward H. and Stanley J. Garstka. 2001. "March Madness and the Office Pool." *Management Science* 47(3):369–382.

Koenker, Roger and Gilbert W. Bassett, Jr. 2010. "March Madness, Quantile Regression Bracketology, and the Hayek Hypothesis." *Journal of Business & Economic Statistics* 28(1):26–35.

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

Massey, Kenneth. 2014. *College Basketball Ranking Composite*. (http://www.masseyratings.com/cb/compare.htm). Accessed on February 22, 2014 and March 28, 2014.

Matuszewski, Erik. 2011. "March Madness Gambling Brings Out Warnings From NCAA to Tournament Players." *Bloomberg News*, March 2011. (http://www.bloomberg.com/news/2011-03-17/march-madness-gambling-brings-out-warnings-from-ncaa-to-tournament-players.html).

McCrea, Sean M. and Edward R. Hirt. 2009. "March Madness: Probability Matching in Prediction of the NCAA Basketball Tournament". *Journal of Applied Social Psychology*, 39(12):2809–2839.

MomentumMedia. 2006. *NCAA Eliminates Two-in-four Rule*. (http://www.momentummedia.com/articles/cm/cm1406/bbtwoinfour.htm). Accessed on February 22, 2014 and March 28, 2014.

Moore, Sonny. 2014. *Sonny Moore's Computer Power Ratings*. (http://sonnymoorepowerratings.com/m-basket.htm). Accessed on February 22, 2014 and March 28, 2014.

Platt, John C. 1999. *Probabilities for SV Machines*. MIT Press. (http://research.microsoft.com/apps/pubs/default.aspx?id=69187). Accessed on February 22, 2014 and March 28, 2014.

Pomeroy, Ken. 2014. *Pomeroy College Basketball Ratings, 2014*. (http://kenpom.com/index.php?s=RankAdjOE). Accessed on February 22, 2014 and March 28, 2014.

Ridgeway, Greg. 2007. "Generalized Boosted Models: A Guide to the GBM Package." *Update* 1(1):2007.

Riedmiller, Martin and Heinrich Braun. 1993. "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm." Pp. 586–591 in *IEEE International Conference on Neural Networks*.

Sagarin, Jeff. 2014. *Jeff Sagarin's College Basketball Ratings, 2014*. (http://sagarin.com/sports/cbsend.htm). Accessed on February 22, 2014 and March 28, 2014.

Schwertman, Neil C., Thomas A. McCready, and Lesley Howard. 1991. "Probability Models for the NCAA Regional Basketball Tournaments." *The American Statistician* 45(1):35–38.

Smith, Tyler and Neil C. Schwertman. 1999. "Can the NCAA Basketball Tournament Seeding be Used to Predict Margin of Victory?" *The American Statistician* 53(2):94–98.

Sokol, Joel. 2014. *LRMC Basketball Rankings, 2014*. (http://www2.isye.gatech.edu/ jsokol/lrmc/). Accessed on February 22, 2014 and March 28, 2014.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* 288:267–288.

Timthy P. Chartier, E. Kreutzer, A. Langville and K. Pedings. 2011. "Sports Ranking with Nonuniform Weighting." *Journal of Quantitative Analysis in Sports* 7(3):1–16.

Toutkoushian, E. 2011. *Predicting March Madness: A Statistical evaluation of the Men's NCAA Basketball Tournament*.