Fall Madness: Predicting the Outcomes of NCAA Tournament Games

Jacob Anderson
School of Computer Science
University of Oklahoma
Norman, OK
Jacob.W.Anderson-1@ou.edu

Joseph McGill
School of Computer Science
University of Oklahoma
Norman, OK
Joseph.A.McGill-1@ou.edu

Abstract

This paper explores the application of two supervised machine learning techniques, linear regression and random forest regression, to predict the outcomes of NCAA Men's Division I basketball tournament games. The models are trained using regular season game data from Kaggle and are evaluated using accuracy, LogLoss, and ESPN's Tournament Challenge bracket scoring rules. The models are compared to similar approaches used as entries in previous versions of the "March Machine Learning Mania" Kaggle competition as well as an agent that picks the winners of games randomly. The linear regression model outperforms both the random agent and the random forest regression model in predicting the winners of tournament games.

Introduction

Every year, at the conclusion of the NCAA Men's Division I basketball season, a 7-round, 68-team tournament is held to determine that year's champion. The participating teams consist of the 32 Division I conference champions and 36 additional teams chosen by a NCAA selection committee. Of the latter, 8 teams play in an initial round consisting of 4 "play-in" games where the losers are eliminated. The remaining 64 teams then participate in a standard 7-round, single-elimination tournament. The overall bracket is divided into 4 regions of 16 teams, each with an initial tournament "seed" that determines matchups and orders teams by their estimated quality (i.e. the best 4 teams in the tournament will each be a 1-seed in their respective region).

This tournament, known as March Madness, has become one of the most popular sporting events in the United States with nearly 30 million viewers tuning in to the 2015 championship game [1]. In addition to watching the games, millions of people fill out brackets to predict the tournament results. Because of the unpredictability of college sports and the large number of games to predict, creating a perfect bracket is practically impossible. As a result, extravagant prizes (most notably, $1 billion [2]) have been offered to anyone who can create such a bracket. Smaller prizes are awarded to the winners of various bracket competitions held by companies such as ESPN and CBS. These prizes, along with general sports gambling, make predicting March Madness games a worthwhile problem that machine learning techniques can be applied to.

In this vein, the website Kaggle.com hosts the "March Machine Learning Mania" competition each year where competitors apply machine learning techniques to predict the outcomes of tournament games. Competitors are tasked with producing a set of 2,278 prediction values (one for each potential tournament matchup) each representing a predicted probability that the first team will win. Submissions are scored using the LogLoss function, which rewards correct predictions and penalizes incorrect predictions more/less when the confidence level of the prediction is high/low. Past participants in this competition have used support vector machines (SVMs), logistic regression, and linear regression with some success [3][4][5]. Some unique approaches come from Ji et al. [6] and Gumm et al. [7] who used matrix completion for artificial neural network (ANN) input and an ensemble of statistical regression methods, respectively.

The approach used in this paper involves both random forest regression and linear regression implementations with the goal of generating accurate predictions for March Madness games as well as generally identifying the features that are most influential to those predictions. Evidence for the effectiveness of random forest approaches to sports prediction problems is given in works by Shi et al. [8], Fernando [9], Lin et al. [10], and Lock et al. [11], where random forests were used in NFL and NBA game predictions. However, during research random forest approaches to the Kaggle competition were not found - making this approach novel in the domain. Linear regression was implemented as a baseline for comparison to the random forest implementation and to provide an alternative approach to identifying influential features. In the interest of giving unique and meaningful evaluation results, the matchup predictions generated by both models were used to create complete tournament brackets, which were scored using ESPN's Tournament Challenge bracket scoring rules.

Problem Definition and Algorithms

Task Definition

The specific goal of the models presented in this paper is to produce a value between 0 and 1 that indicates a predicted probability of "team A" winning a given NCAA tournament matchup "team A vs. team B" using game statistics for the given teams and general tournament information. These statistics are available in a dataset taken from the 2016 version of the previously mentioned Kaggle competition. It contains detailed regular season and tournament game results for more than 72,000 games over 13 seasons (2003 - 2015). These results contain various features related to a team's offensive and defensive performance including most of the statistics typically recorded for a basketball game. Six of the 34 total features are not used for prediction (date, location, etc.), while the remaining 28 features are simply 14 recorded statistics - one for each participating team.

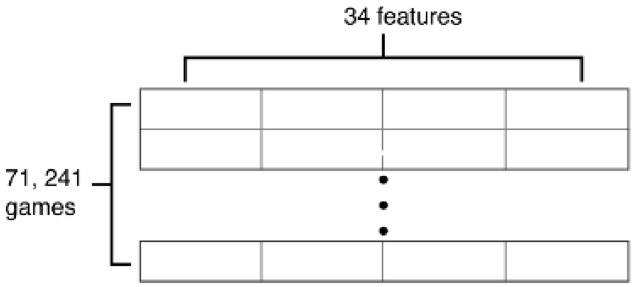


Figure 1: Dataset Structure – Detailed Regular Season Results

The steps taken to extract and modify the raw data are as follows:

1. For each team in for a given season, obtain regular season averages of all 28 features. This will result in 14 "offensive" features (points/game, etc.) and 14 "defensive" features (points allowed/game, etc.).

2. Take the differences between the corresponding offensive and defensive features to obtain 14 stat differential features for each team.

3. Normalize all 14 stat differentials with respect to those of all other teams in the given season.

4. For each tournament matchup, concatenate the 14 normalized stat differentials for both participating teams giving a total of 28 features.

The random forest regression model will take the entire list of 28 features described above for a given tournament matchup as input and will output a predicted probability (between 0 and 1) that the first team will win the matchup.

The linear regression model will use a subset of the 28 features described above found using simple forward feature selection. The selected features are the differentials for score, 3-pt field goals made, and 3-pt field goals attempted. Taking this set of 6 features as input, the linear regression model will output a predicted probability (between 0 and 1) that the first team will win the matchup (this is the same output as the random forest regression model).

Algorithm Definitions

Random forests utilize bootstrap aggregation to reduce the variance in the model by randomly sampling the dataset with replacement. The random samples are then used to train many regression trees (hence the "forest"). What makes the forest random is the method in which each regression tree is split. At each split, the best feature from a random subset of all features is chosen as the split attribute. Once the random trees are fitted, predictions can be made using the forest by making a prediction with each tree in the forest and averaging those results. This allows trends in the data to be learned by most of the trees in the forest. Pseudocode for the random forest Regression implementation is given below:

```
For 1 … NUM_TREES:
    1. Sample, with replacement, n (input training set
       size) samples from input training set.
    2. Train a regression tree on the sample, but at each
       split select from a random subset of all features of
       size n/3.
    3. Add the trained regression tree to the forest.
```

Linear regression aims to find the underlying trend in the data by fitting a line through the data points. The typical linear model is

$$Y = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

where Y is the predicted value, the beta values are the weights of each of the features, and the x values are the features of the dataset. During training, linear regression updates the weights of the features by computing the error of each sample and adjusting the weight accordingly. Pseudocode for the linear regression implementation is given below:

```
1. Insert a column of 1's to all samples (bias term)
2. Initialize all weights to 0
3. For 0 … NUM_ITERATIONS:
       For each sample:
           Error = sample error
           weights = weights - alpha*error*example
```

Experimental Evaluation

Evaluation Metrics

The criteria used to evaluate the models described are LogLoss, accuracy, average bracket Score, and best bracket score. LogLoss rewards correct predictions and penalizes incorrect predictions more/less when the confidence level of the prediction is high/low. LogLoss is given by the equation

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1-y_i)\log(1-p_i)]$$

where N is the number of samples, y is the label of the sample, and p is the prediction for the sample. Note that a smaller LogLoss value is desirable.

Accuracy is simply the number of games correctly predicted divided by the number of games predicted. For accuracy calculation, values greater than or equal to 0.5 will be considered a prediction that the first team in a matchup will win and values less than 0.5 will be considered a prediction that the second team will win.

The average bracket score and best bracket score metrics are determined by using model predictions to generate 1000 "random" brackets. These random brackets are created by initializing a bracket with the correct matchups using information from our dataset and then advancing the first team in each matchup with a probability equal to the model prediction for that matchup. This is repeated until a complete bracket is generated. The completed brackets are then scored using ESPN's Tournament Challenge scoring rules, given below:

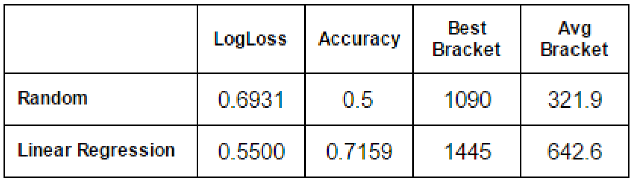
Play-in games: Contestants not required to make pick
Round 1: 10 points per correct pick
Round 2: 20 points per correct pick
Round 3: 40 points per correct pick
Round 4: 80 points per correct pick
Round 5: 160 points per correct pick
Championship: 320 points per correct pick

Experiment 1: Linear Regression

Methodology

In this experiment, it is proposed that a linear regression model will significantly outperform an agent that chooses the winners of games randomly. The model has a learning rate of 0.00001 and uses a reduced feature set found through forward feature selection of seed and differentials for score, 3-pt field goals made, and 3-pt field goals attempted. Results have been generated by taking the average of the results of thirteen train/test splits (one with each season as the test season and the rest as training seasons).

Results



	LogLoss	Accuracy	Best Bracket	Avg Bracket
Random	0.6931	0.5	1090	321.9
Linear Regression	0.5500	0.7159	1445	642.6

Figure 2: Experiment 1 Results

Discussion

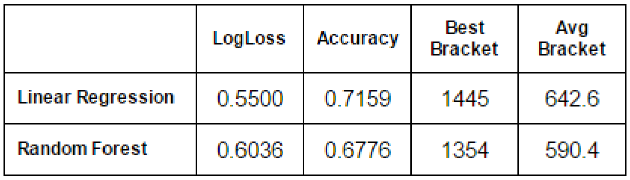
These results support the hypothesis in that every evaluation metric shows a significant improvement for the linear regression model over the random agent. These results were expected and show the general effectiveness of a simple linear regression model.

Experiment 2: Random Forest Regression

Methodology

In this experiment, it is proposed that a random forest regression model will outperform the previously mentioned linear regression model. The model uses 75 trees, no max tree depth, and 10 minimum samples at each tree leaf. Results have been generated by taking the average of the results of five train/test splits (2011-2015).

Results



	LogLoss	Accuracy	Best Bracket	Avg Bracket
Linear Regression	0.5500	0.7159	1445	642.6
Random Forest	0.6036	0.6776	1354	590.4

Figure 3: Experiment 2 Results

Discussion

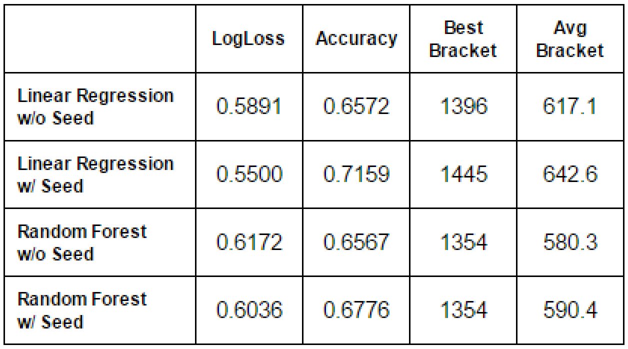
These results do not support the hypothesis in that every evaluation metric shows a significant decrease in performance for the random forest regression model. These results were not expected, but could be explained by the fact that the time-complexity of the random forest model hindered extensive parameter tuning. It is possible that unseen combinations of learning parameters (more trees, etc.) would improve the performance of the model.

Experiment 3: Seed Feature

Methodology

In this experiment, it is proposed that the "seed" feature is the most influential feature in the dataset. To test this hypothesis, results for both the linear regression and random forest regression models will be obtained with and without the seed feature included in the feature set. The train/test splits for the linear regression and random forest model results are the same used in experiments 1 and 2, respectively.

Results



	LogLoss	Accuracy	Best Bracket	Avg Bracket
Linear Regression w/o Seed	0.5891	0.6572	1396	617.1
Linear Regression w/ Seed	0.5500	0.7159	1445	642.6
Random Forest w/o Seed	0.6172	0.6567	1354	580.3
Random Forest w/ Seed	0.6036	0.6776	1354	590.4

Figure 4: Experiment 3 Results

Discussion

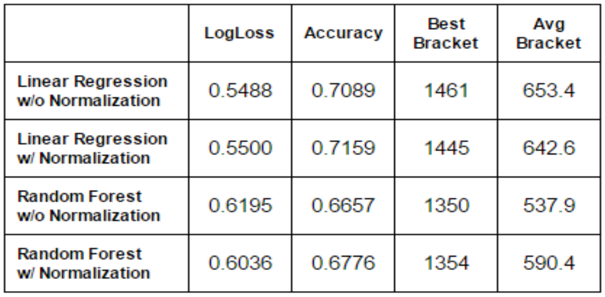
The results support the hypothesis in that, for both models, every evaluation metric shows improvement when the seed feature is included in the feature set. Additionally, when performing forward feature selection for the linear regression model, the seed feature is the first feature selected. This provides further support that the seed feature is the most influential feature in the dataset.

Experiment 4: Data Normalization

Methodology

In this experiment, it is proposed that normalizing the stat differentials prior to training both the linear regression and random forest regression models will result in improved performance. To test this hypothesis, results for both the linear regression and random forest regression models will be obtained with and without normalizing the stat differential features. The train/test splits for the linear regression and random forest model results are the same used in experiments 1 and 2, respectively.

Results



	LogLoss	Accuracy	Best Bracket	Avg Bracket
Linear Regression w/o Normalization	0.5488	0.7089	1461	653.4
Linear Regression w/ Normalization	0.5500	0.7159	1445	642.6
Random Forest w/o Normalization	0.6195	0.6657	1350	537.9
Random Forest w/ Normalization	0.6036	0.6776	1354	590.4

Figure 5: Experiment 4 Results

Discussion

The results shown do not strongly support or reject the given hypothesis. Though some metrics show some improvement (specifically those for random forest regression model), the improvements are not statistically significant or consistent across all metrics. Therefore, it cannot be said that normalizing the data improves the performance of either model.

Related Work

Because most of the related approaches were entered into previous versions of the aforementioned Kaggle competition, their results are typically only given for a single test season. In order to provide more accurate and generalized results, the results presented in this paper have been averaged over test seasons ranging from 2003 to 2015. This means that it is very difficult to directly compare the results obtained to those of others.

Predicting the winners of March Madness tournament games using machine learning has been tackled by others before using interesting approaches. Tran et al. [3], Chanen et al. [4], and Franklin [5] framed the problem as a classification problem and used Support Vector Machines (SVMs) and logistic regression on varying datasets. All 3 works ran into issues with their SVMs overfitting to the training data and reducing their test accuracies. However, since random forests average predictions over all trees they are not as susceptible to overfitting as SVMs.

Ji et al. in [6] used matrix completion on the input matrix to fill in missing entries in the data. The completed matrix was used as input for an artificial neural network to output a predicted probability that a team would win. Using this method the authors were able to achieve a LogLoss of 0.56915 for the 2015 March Madness tournament, which is slightly worse than the averaged results found in this paper.

In [7], Gumm et al. took a purely statistical approach to predicting the winners of the games. The authors used an ensemble of regression methods taken from various statistical models with the 2014 version of the previously mentioned Kaggle competition dataset. Using their purely statistical approach, Gumm et al. achieved an accuracy of 66% for the 2014 March Madness tournament. This paper makes uses of machine learning methods rather than pure statistics; however, the results of [7] are comparable to the results of the random forest implemented.

Shi et al. used random forests in [8] to predict the winners of NCAA Division 1 men's basketball regular season games. The authors' accuracy results are comparable to the accuracy results found by the random forest implementation used in this paper. However, Shi et al. predicted regular season games rather than tournament games, which makes a direct comparison of accuracies difficult.

Random forests were also used by Fernando [9], Lin et al [10], and Lock et al [11] for sports prediction problems. Fernando and Lock et al. applied random forests to predicting NFL games, while Lin et al. applied them to predicting NBA games. Additionally, some approaches using methods other than random forests were explored. In [12], Ivanković et al. used ANNs to predict the winners of Serbian First B basketball league games. Delen et al. in [13] used ANNs, decision trees, and SVMs to predict the winners of NCAA football bowl games. Like Lin et al. in [10], Aryan et al in [14] predicted NBA game outcomes. The authors in [14] used linear regression, logistic regression, and SVMs for prediction. Unfortunately, the works mentioned above are in different problem domains and are not directly comparable to the results found in this project.

Future Work

One major shortcoming in this work is the lack of training data. While there are 71,241 regular season games recorded in the dataset, only 841 tournament games are recorded. The small sample size makes supervised learning using tournament games difficult and leads to lower quality results from the models. Future work could include using regular season games in a more meaningful way to increase the number of training samples.

Another shortcoming was the lack of features in the training set. The dataset only contains 14 features for both teams playing in a given game. Adding the team's seed gives another feature for a total of 30 usable features for prediction. Additionally, the models used in this paper did not make use of any tournament games played prior to the matchup being predicted. Future work could include deriving more advanced features from the dataset as well as using prior tournament games played to increase prediction results.

Lastly, more extensive parameter tuning can and should be performed on the random forest regression model. The results obtained were not quite as high as anticipated, and it is believed to be due to sub-optimal learning parameters. Future work could invest more time/resources to perform the time-intensive experiments required to determine more optimal learning parameters for the model.

Conclusion

In this paper, two supervised learning techniques were implemented and applied to the problem of predicting the outcome of NCAA Men's Division I basketball tournament games. The models were trained on a feature set derived from a dataset provided in a related Kaggle competition. Once trained, the models were evaluated using LogLoss, accuracy, and by creating/scoring brackets using ESPN's Tournament Challenge scoring rules.

The primary result of this paper is the performance of the implemented models, which show that a simple linear regression model significantly out-performs a random agent. Though its results were less than expected, the random forest regression model also performed well above the random agent. Practical results for both models were given in the form of average/best bracket scores, which are directly applicable to real-world competitions hosted by companies such as ESPN and CBS. Finally, the identification of seed, scoring differential, 3-pt scoring differential, and 3-pt attempts differential as influential factors in predicting the outcome of college basketball games was shown.

References

[1] F. Pallotta (2015). March Madness is a TV slam dunk: Highest ratings in 22 years. Retrieved from http://money.cnn.com/2015/04/07/media/march-madness-tv-ratings/ (accessed 10/29/2016)

[2] R. Lane (2015). The Only Way to Play Warren Buffett's NCAA March Madness Billion-Dollar Bracket This Year. Retrieved from http://www.forbes.com/sites/randalllane/2015/03/25/the-only-way-to-play-warren-buffetts-ncaa-march-madness-billion-dollar-bracket-this-year/#2b4c478f53f5 (accessed 10/24/2016)

[3] A. Tran and A. Ginzberg (2014). Making Sense of the Mayhem: Machine Learning and March Madness. Dept. of Computer Science, Stanford University. Retrieved from http://cs229.stanford.edu/proj2014/Adam%20Ginzberg,%20Alex%20Tran,%20Making%20Sense%20of%20the%20Mayhem-%20Machine%20Learning%20and%20March%20Madness.pdf (accessed 09/16/2016)

[4] E. Chanen and J. Gold (2014). CS229 Final Report - Machine Learning Madness. Dept. of Computer Science, Stanford University. Retrieved from http://cs229.stanford.edu/proj2014/John%20Gold,%20Elliot%20Chanen,%20Machine%20Learning%20Madness.pdf (accessed 09/17/2016)

[5] L. Franklin (2014). Predicting March Madness: Winning the Office Pool. Dept. of Computer Science, Stanford University. Retrieved from

http://cs229.stanford.edu/proj2014/Levi%20Franklin,%20Predicting%20March%20Madness.pdf (accessed 09/16/2016)

[6] H. Ji, E. O'Saben, A. Boudion, and Y. Li (2015). March Madness Prediction: A Matrix Completion Approach. Dept. of Computer Science, Old Dominion University. Retrieved from http://www.cs.odu.edu/~yaohang/publications/MarchMadness.pdf (accessed 09/16/2016)

[7] J. Gumm, A. Barrett, and G. Hu (2015). A Machine Learning Strategy for Predicting March Madness Winners. *2015 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7176206 (accessed 09/15/16)

[8] Z. Shi, S. Moorthy, and A. Zimmermann (2013). Predicting NCAAB match outcomes using ML techniques - some results and lessons learned. University of Leuven. Retrieved from: http://arxiv.org/pdf/1310.3607v1.pdf (accessed 09/17/2016)

[9] N. Fernando (2016). Predict the Winners of the Big Games with Machine Learning. Retrieved from http://data-informed.com/predict-winners-big-games-machine-learning/ (accessed 09/15/2016)

[10] J. Lin, L. Short, and V. Sundaresan (2014). Predicting National Basketball Association Winners. Dept. of Computer Science, Stanford University. Retrieved from http://cs229.stanford.edu/proj2014/Jasper%20Lin,%20Logan%20Short,%20Vishnu%20Sundaresan,%20Predicting%20National%20Basketball%20Association%20Game%20Winners.pdf (accessed 09/17/2016)

[11] D. Lock and D. Nettleton (2014). Using Random Forest to Estimate Win Probability Before Each Play of an NFL Game. *New England Symposium on Statistics in Sports*. Retrieved from http://www.nessis.org/nessis13/lock.pdf (accessed 09/18/2016)

[12] Z. Ivanković, M. Racković, B. Markoski, D. Radosav, and M. Ivković (2010, November). Analysis of basketball games using neural networks. 11th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5672237 (accessed 09/15/2016)

[13] D. Delen, D. Cogdell, and N. Kasap (2012). A comparative analysis of data mining methods in predicting NCAA bowl outcomes. International Journal of Forecasting, 28. pp. 524-522. Retrieved from http://ac.els-cdn.com/S0169207011000914/1-s2.0-S0169207011000914-main.pdf?_tid=29a4474a-7dc2-11e6-9433-00000aab0f6c&acdnat=1474218535_25fe3e827db4c5d52f90ad5a580a923c (accessed 09/17/2016)

[14] O. Aryan and A. R. Sharat (2014). A Novel Approach to Predicting the Results of NBA Matches. Dept. of Computer Science, Stanford University. Retrieved from http://cs229.stanford.edu/proj2014/Omid%20Aryan,%20Ali%20Reza%20Sharafat,%20A%20Novel%20Approach%20to%20Predicting%20the%20Results%20of%20NBA%20Matches.pdf (accessed 09/16/2016)