# NCAA MARCH MADNESS MACHINE LEARNING CHALLENGE

**SUBMITTED BY:**
Erik Kahnke
Gabriel Urrutia
Kartik Sharma
Sanchit Shaleen
Suraj Tata Prasad

## Table of Contents

## 1. Abstract:

The Division I NCAA Men's Basketball Tournament is a popular sporting event held annually to determine the leagues National Champion. Over the past several years the betting scene surrounding the tournament has become arguably more popular than the tournament itself, drawing in fans who bet billions overall on its outcome. The aim of this project was to learn which learning model, feature selection technique, ordering process, and distance criteria provided the best accuracy in predicting the outcome of any NCAA Men's Basketball Tournament match. In this paper, we discuss the statistical challenges in correctly predicting winners in the tournament and present a machine learning strategy for predicting the games. These machine learning techniques are trained on historical team, regular season and tournament data.

The Kaggle Machine Learning March Mania Competition was used to test the effectiveness of the model by comparing it against other machine-learning-based models submitted to the competition and the performance of predictions based on seed. We were able to combine the different modeling techniques using the ensemble method which produces Log loss score of around 0.58, which exceeds Seed-based benchmark method (0.63). Furthermore, we analyze the pitfalls that were encountered during our analysis.

## 2. Introduction:

Every March the NCAA Basketball Championship Tournament attracts millions of fans who passionately follow March Madness until a champion is crowned. Fans typically fill out a bracket by selecting a winner for every matchup in every round to compete in pools for bragging rights amongst peers and cash prizes with friends, family or co-workers.

The tournament's format was recently changed to accommodate 68 teams instead of the usual 64. 32 teams receive automatic invitations for winning their conference's championship tournament at the end of the regular season. The remaining 36 teams receive at-large bids that are determined by the selection committee who is also responsible for tournament seeding. Teams are seeded 1 through 16 in one of four regions (East, Midwest, South, West) based on their regular season performance, strength of schedule and quality of wins. One of the most exciting characteristics of the tournament is its penchant for upsets. Seemingly every year a "Cinderella" school defies the odds and makes a deep run into the later rounds. This uncertainty makes predicting the outcomes of each matchup very difficult. Oftentimes, the people with the least amount of knowledge of the teams or college basketball in general that do just as well as experts who spend days analyzing the matchups.

An estimated 60 million Americans filled out tournament brackets in 2015, but this does come at a productivity cost. According to Challenger, Gray & Christmas, the cost of lost wages paid to distracted and unproductive numbers was predicted to be as high as $1.9 billion for the 2015 tournament.

The NCAA Basketball Tournament has implanted itself in the fantasy sports world with Yahoo Sports' Tourney Pick'em, and has also become an exciting topic in the Sports Analytics Industry. Because of the tournament's popularity among Americans, competitions have emerged that offer the winners substantial monetary rewards. Working with Yahoo Sports Tourney Pick'em, Warren Buffet offered $1 billion to anyone who completed a perfect bracket by successfully predicting the winner of every tournament matchup. Facing very steep odds (1 in 9.2 quintillion), nobody was able to claim Buffet's prize, but $100,000 was awarded to the top twenty performing brackets.

We have developed a model to predict the outcome of each matchup in every round of the NCAA Tournament given the tournament teams, seeding, historical data on individual games and aggregate measures for that season. Our dataset provides data spanning from 2010 to 2015. We plan to use data from 2010 to 2014 as our training set, and to predict on data from 2015. We chose to focus on these years because the college game has moved to a more perimeter oriented game recently with the shortening of the Three Point Line. To keep consistency with our historical data, we will only predict the outcomes from the round of 64 going forward. The original dataset contains features on individual game statistics and some categorical features. We want to boost our predictive ability by incorporating other measures such as strength of schedule, wins versus the Top 25 and RPI (Rating Percentage Index).

We hope to offer a statistical, machine learning tool to assist college basketball fans, bettors and any other interested parties in predicting the outcomes of every tournament game or, at the very least, determine how well our model performs relative to the baseline of always selecting the higher seed to advance in each round, which produced the log loss value of 0.63.

In general, predicting the outcomes of college basketball games is more challenging than for professional games due to the greater variability in team performance from season to season. New players are introduced on a yearly basis, and team dynamics change much more frequently. Measuring strength in team over various seasons will also depend on the strength of schedule in each season.

Dean Oliver identified the "Four Factors"—shooting, turnovers, rebounding, and free throws—that best quantify success in college basketball and optimize predictive modeling accuracy. It is clear that an effective prediction strategy for NCAA games requires a blend of informative data inputs and machine learning techniques that perform best given the training data. What may be considered a favorite machine learning technique may have poor performance if trained on irrelevant data. Prediction becomes even more difficult as one must both optimize the variables to train the model on and the model itself.

## 3. Objectives:

Our goal was to identify key factors in predicting NCAA tournament wins and to find a model that would perform well in the Kaggle competition. The aim was to create a list $\hat{y}$ of 2278 prediction probabilities (values between 0 and 1) that each team in the tournament would defeat every other team in the tournament, regardless of whether this match-up actually occurs. We use log-loss function to quantify the effectiveness of our model and this scoring method heavily penalizes being simultaneously confident and wrong. A, it is important to balance between being too conservative and too confident. If we are too conservative (e.g. $\hat{y} = 0.5$), then we will never accrue points, but if we are too confident, we make ourselves vulnerable to huge losses.

## 4. Exploratory Data Analysis:

We looked at the problem from the perspective of someone who would actually fill out the March Madness bracket. Although we may have some intuition as to which effects are generally good or bad objectively, for example personal fouls are bad and three point shots attempted are good, we would like to know exactly how strongly they are correlated with winning games. We started by checking assumptions like **1)** The favorite teams usually win **2)** Upsets generally happen when certain seeds match-up **3)** Shot accuracy matters more than the number of attempts etc. We then moved onto assessing the direction and strength of relationships between explanatory and outcome variables.

### 4.1 Team Seeds and Upsets:

**For Round of 64:** Here we wanted to check with what level of confidence can someone bet on a high ranked team, and as far seeds 1, 2 are concerned it's safe to bet on them to win almost every single time. And, with the exception of the performance of 5-seeded teams, the gradually decreasing results indicate how well the tournament selection committee does at seeding these school teams.

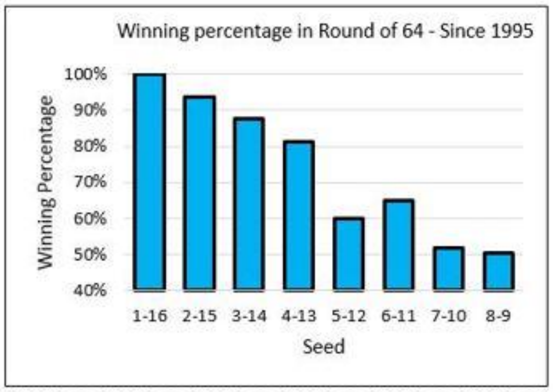


Figure EDA_1: There is a straightforward winning % trend indicating that seed-based betting is safe in this round

**Insights:**

1) All 1-seeds are virtually guaranteed to win and easy picks to advance to the next round.

2) The 2, 3 and 4-seeds are all above a .750 winning percentage. But some will lose. You will want to choose 2 schools among these 12 teams to be an unexpected miss – a possible sign of experienced coaches looking ahead.

3) Two 5-seeds will likely be upset. Selecting two 12-seeds wouldn't be unreasonable.

**For Round of 32:** But when we move on to the second round, the trend is not so straightforward anymore. Historically the 8th and 9th seeded teams win fewer games than even the 12th and 13th seeded team because 8 and 9 face 1st seeded team but 12 and 13 face 5th or 4th depending on the previous round results.

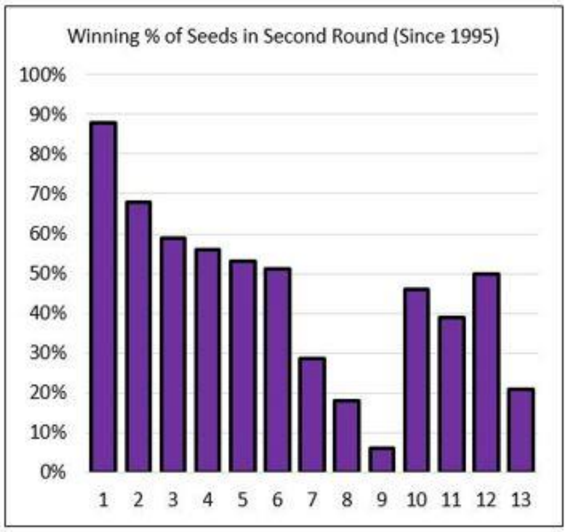


Figure EDA_2: Trend is not so straightforward anymore. Still safe to bet on 1 and 2-seeds. Be vary of 8 and 9-seed.

**Insights:**

1) Wouldn't be giving up on any 1-seeds yet but you can start crossing off any seeds 13 and higher.

2) In the Round of 32, 2-seeds have won between 2/3 and 3/4 of their games. The numbers seem to lean towards advancing a 10-seed (not a 7-seed) if you are going to take a chance with an upset of a 2-seed in this round.

3) 3-seeds win about 2/3 of their games versus 6-seeds; whereas, 4 and 5-seeds generally split their matchups.

4) Do not bet on 8 and 9 – seed unless you have some strong reason to believe they are going to win it.

**For Round of 16:**

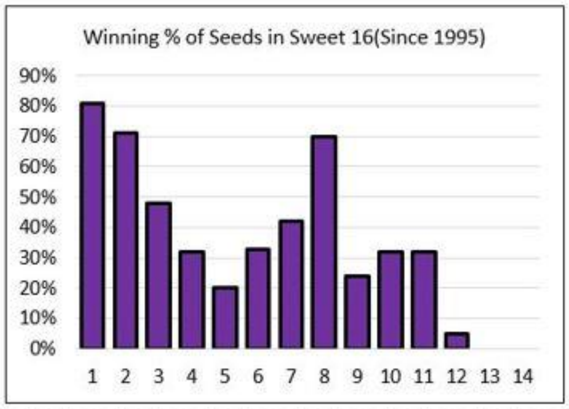


Figure EDA_3: Tough to make a safe bet beyond 1 and 2-seeds. Can expect about six 1 through 3-seeds to advance

**Insights:**

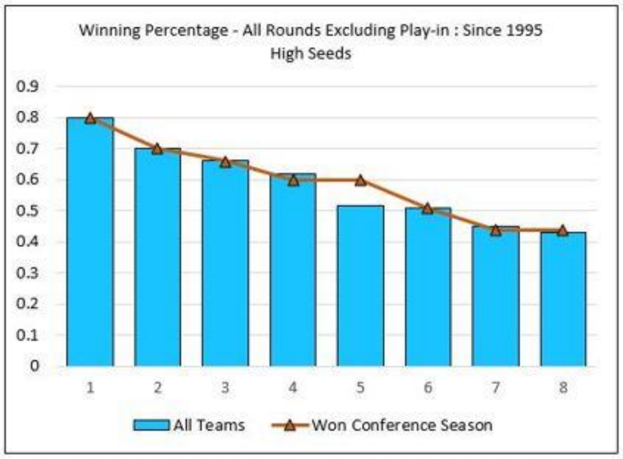
1) In Round of 16, historically, 1-seeds beat 4-seeds and 2-seeds beat 3-seeds in approximately 60% of tournament games.

2) You could expect about six 1 through 3-seeds to advance and the remaining be filled out by the field.

3) Elite 8 onwards it is no longer safe rely on schools with more tournament victories over the past three years to have a distinct advantage.

**Do Conference Season wins matter?**

All teams take part in the Conference Season in the lead up to the March Madness tournament and we wanted to check if the performance here was any indicator of how same-seeded teams perform.



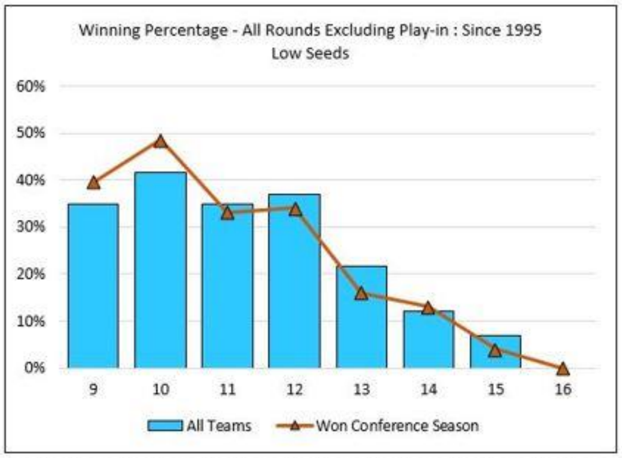


Figure EDA_4: With the exception on 5th, 9th and 10th seeds, the Conference Season wins does not make a difference to the winning %.

Can be seen that with the exception on 5th, 9th and 10th seeds, the Conference Season wins does not make a difference to the winning %.

## 4.2 Features most affecting Winning Performance

In order to understand the dynamics of what types of teams are most likely to win games in the March Madness tournament, we first need to gain a sense of what typically is linked to a win on an individual game level.

In addition to the features already provided, we did some modifications like taking a difference of game statistics to further interpret the significance of each feature. Since the winning team had to have scored more points, we explore to see if on average the winning team is also attempting more baskets but, surprisingly, we find that in 60% of the games the winning team actually took less shots than the losing team. We then see that in over 80% of the games the winning team had a higher field goal accuracy, displayed on the right. One might have guessed that teams that shoot more are the ones score more points. But, clearly the important factor in winning a game is how effective a team is in terms of shooting accuracy.

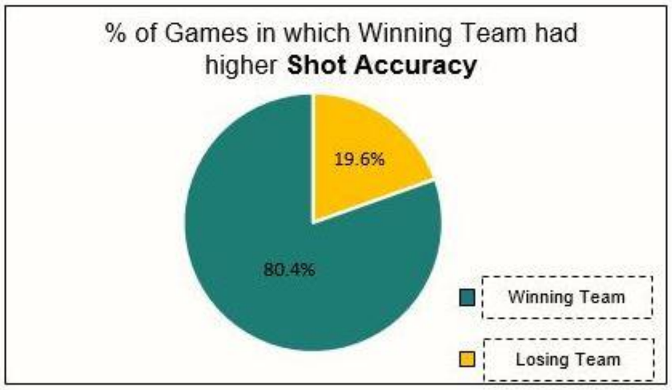


Figure EDA_5: 80% of the time the winning team had a higher shot accuracy

Another important predictor is total number of free throws attempted. On the right, we display the ratio of games in which the winning team also had a higher number of free throws attempted than the losing team. As we can see, in about ⅔ of all games, the winning team was taking more free throws. This is likely because free throws are pretty easy points and getting to the line is advantageous. Also, there is more likelihood that when a team is losing a game they will foul more and this results in more free throws.

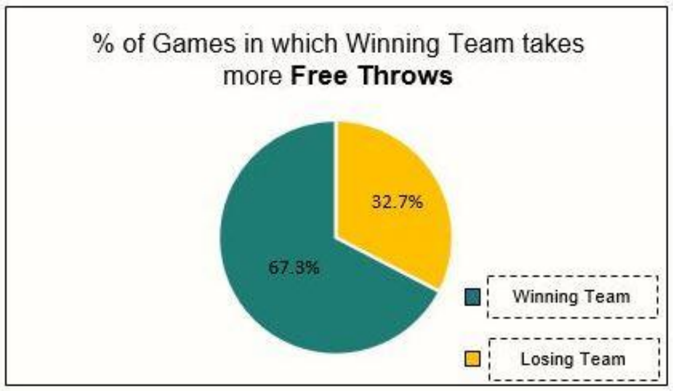


Figure EDA_6: 68% of the time the winning team took more Free Throws

Further, we see that the number of turnovers is also an important factor in determining whether or not a team will win the game. We observe that in only about ⅓ of games did the winning team lose the ball more.

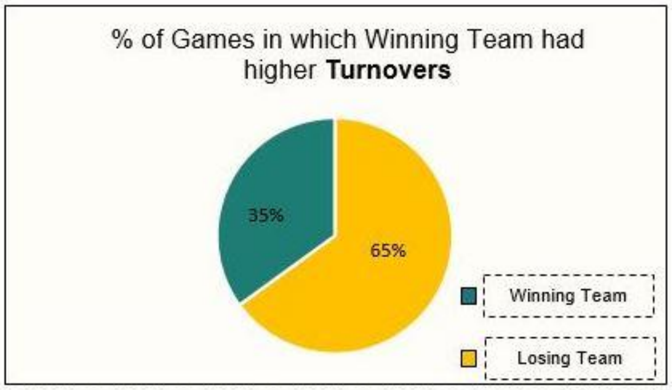


Figure EDA_7: 65% of the time the Losing team had a higher Turnover

Finally, we have the % of games in which the winning team also had more defensive rebounds than the losing team. However, when testing for total number of offensive rebounds, we find that it is not very significant in predicting which team will be victorious.

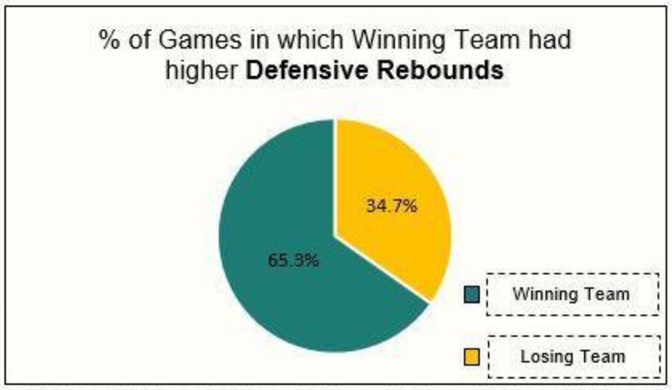


Figure EDA_8: 65% of the time the winning team had a higher number of Defensive Rebounds

## 5. Data and feature extraction:

We derived our datasets from the kaggle competition website and utilized 2010- 2014 regular season and tournament data to predict the outcome of 2015 tournament. Some additional features were also collected from NCAA and Greenfield rankings to help the analysis. The data contained following features for the winning and the losing team:

Field goals attempted, field goals made, three pointers attempted, three pointers made, free throws attempted, free throws made, offensive rebounds, defensive rebounds, assists, turnovers, steals, blocks, personal fouls, rebound margin assists per game, turnover margin assist-turnover ratio, field goal percentage, field goal percentage defense, ratings power index (RPI), difference in conference RPI

Each instance in the dataset provided contains all of this information for the winning and losing team in each NCAA regular season and tournament game since 2010. Game-by-game statistics were provided, but for prediction purposes all of the statistics for 2014-15 season were rolled into season averages for each team for use in the tournament predictor. This allows for predictive evaluation of teams, as using game-by-game predictors of outcome is not helpful, since it is impossible to know the game statistics prior to the game occurring. All factors are expected to have a positive relationship with winning apart from adjusted defense, true shooting percentage allowed and turnover differential, where in all three cases the goal is to have lower figures than opponents. The object of this competition is to minimize log loss probability so a binary win loss indicator was created, however, a continuous predictor could be created using the margin of victory for each team in each game as well.

All the features were initially utilized in the logistic regression model which resulted in poor prediction accuracy. On further investigation, we found that many variables were correlated which worsened the efficiency of the model and they were removed. Also, using step feature selection we came up with final set of features which had the greatest effect on deciding the result of the games:

Adjusted Offense: offensive efficiency relative to number of possessions each team has – this takes efficiency of scoring into play, rather than just looking at points per game, since teams play at different speeds, and get different numbers of opportunities per game

Adjusted Defense – defensive efficiency relative to number of possessions each team has

Adjusted Tempo – measures the speed at which teams' play – purpose of including in the model is to see if faster or slower teams have an advantage

SOS Difference – Strength of schedule measures the quality of competition for each team

Location – Whether the winning team was playing at home, away or on a neutral court

Offensive Rebound Differential – Average offensive rebound differential per game for a team, calculated the same way as average point differential

Defensive Rebound Differential – Average defensive rebound differential per game for a team, calculated the same way as others in this section.
Turnover Differential – Average turnover differential per game for a team, calculated the same way as other in this section
Three Point Reliance – Percentage of points scored from three pointers for a team
True Shooting Percentage – True shooting percentage for a team. True shooting percentage is a more accurate version of shooting percentage, as it weights shots based on their value (ex. Three---pointers are more valuable).
True Shooting Percentage Allowed – True shooting percentage allowed for a team. Same as above, but what the team allows rather than shoots.
Block Differential – Average difference in blocks per game for team vs. opponents.

**5.1 Page Rank as a predictor:**

We included two other factors in our model, team and conference rankings, using the PageRank algorithm. PageRank was initially used to improve the efficiency of search queries on the Internet. Webpages are represented by nodes in a graph with directed edges connecting nodes if a link to one website exists on another. Thus, if webpage A has a web link that directs the user to webpage B, the associated graph has a directed edge from node A to node B. The edges are weighted with respect to how many incoming edges a node possesses. These ideas can be further extended to other fields, in particular, to sports.
Ranking basketball teams is analogous to ranking webpages. The associated graph would have nodes representing basketball teams and directed edges between teams that have played each other with the edge directed towards the winning team. The edges are weighted based on point differentials during regular season games. Thus a team's in-degree represents the number of other NCAA tournament teams it defeated during the season, and its out-degree represents the number of other NCAA tournament teams it was defeated by during the season.
The weighted adjacency matrix is then adapted by dividing each value in the matrix by the sum of every element in the matrix (sum of point differentials for every game) to make a transition matrix. By multiplying the transition matrix and a starting vector, with the number of rows equal to the number of columns in the transition matrix and with each entry being the reciprocal of the number of columns, one can repeat the multiplication by the transition matrix to produce a stationary result that represents the steady-state value of each row (team). These values can then be ranked to create a team ranking. In this way the teams are ranked based on the total weights on incoming edges. A dominant team will have many edges entering its node, with few edges leaving. The edge weights will contribute to the magnitude of their dominance. The

resulting team and conference rankings were used as inputs in our model. This model magnifies the importance games played against teams of the quality that a team will face in the tournament.

## 6. Benchmarks:

Kaggle provided benchmarks to surpass for each stage of the competition. These relatively simple, public predictive models were informative.

**Seed Benchmark:** This benchmark uses only the teams' tournaments seeds ranging from 1 (best) to 16 (worst). For each game i, suppose that team A is stronger than team B, and let $s_i$ (A), $s_i$ (B) be the corresponding seeds. Then, the probability $\hat{y}_i$ *(A,B)* of Team A beating Team B is given by :

$$\hat{y}_i(A,B) = 0.50 + 0.03 \cdot (s_i(B) - s_i(A))$$

The seed benchmark can be derived by fitting an OLS regression model to the difference in seeds in the past tournaments.

**All Zeros Benchmark:**

This is the benchmark where one chooses a zero probability for each game ($\hat{y}_0$ = 0). One will not lose infinite points as Kaggle bounds the logloss function away from infinity, but one will lose the maximum number of points for each wrong prediction, and gain the maximum number of points for each correct prediction. This number turned out to be L($\hat{y}_0$) = 19.19 **-** this is the worst score that you could get in the competition.

**All 0.5 Benchmark:**

This is the "coin flip" benchmark where one chooses $\hat{y}_{0.5}$ = 0.5 for each game. One will lose the same amount of points on each game ln(0.5) = 0.693. Thus, L($\hat{y}_{0.5}$) = 0.693.

## 7. Approach:

We used an aggregate model of an ensemble of regression functions derived from the most highly-correlated statistical differentials in regard to wins. The training data included results from the 2010-2013 regular season and tournament games including season statistics for each team. For testing the accuracy of the predictions of our model we used 2015 tournament data. We aggregated the team features for the 2014 regular season and used it as the input features for our model to predict the winning probabilities for all the possible 2,278 matchups. The prediction accuracy for the ensemble model is measured by the log loss function.

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.

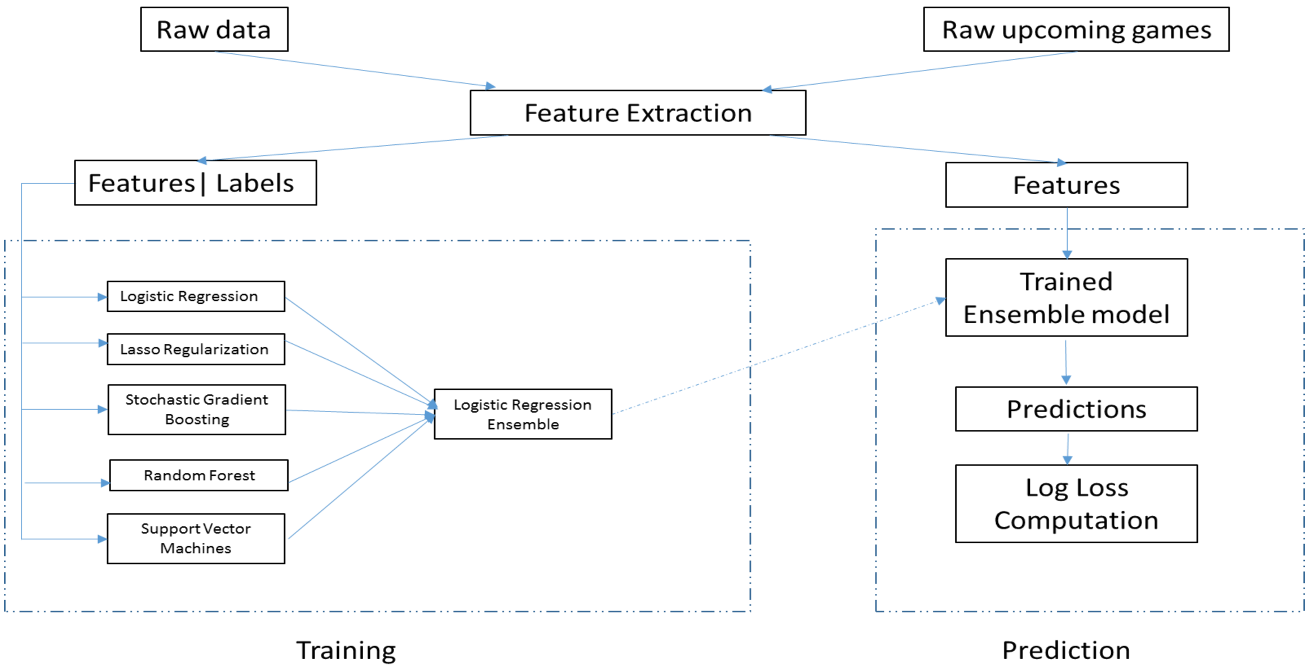


Figure Flowchart_1: Model process flow - Sequential steps involved in model building and predicting

## 7.1 Measuring Model Efficiency:

Standard systems for scoring NCAA basketball tournament pools, including those used in contests hosted online by ESPN and Yahoo, award points based on picking each tournament game winner correctly, where picks are made prior to start of the tournament. In these pools, there are no lost points for incorrect picks, but it is impossible to pick a game correctly if you had previously eliminated both participating teams in earlier rounds. The standard point allocation ranges from 1 point per game to 32 points for picking the tournament winner, or some function thereof, with successive rounds doubling in value. With the final game worth 32 times each first round game, picking the eventual tournament champion is more or less a prerequisite for a top finish. This approach generally results in a very low accuracy rate due to the single elimination aspect and therefore we will only be measuring our average accuracy on correctly predicting the outcome of any possible match up.

Systems that classify games as win or lose fail to provide probability predictions, and without probabilities there is no information provided regarding the strength of victory predictions. For example, a team predicted to win with probability 0.99 by one system and 0.51 by another would both yield a win prediction, even though these are substantially different evaluations.

An alternative structure, used in the Kaggle competition, involved submitting a probability of victory for every potential matchup among the 68 teams prior to the play-in round. This corresponded to 2278 probability predictions. A submission would then be evaluated by a measure of predictive discrepancy on the 63 games that were played in the bracket of 64 teams.

Let $\hat{y}_i$ be the predicted probability of Team 1 beating Team 2 in game i on submission j, where i = (1, … , 2278) and j = 1, … , S , where S is the total number of submissions. Let $y_i$ equal 1 if Team 1 beats Team 2 in game i and 0 otherwise. The $\text{LogLoss}_{ij}$ for entry j from game i is defined as

$$L(y|\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \cdot \log(\hat{y}_i) + (1-y_i)\cdot \log(1-\hat{y}_i)]$$

Only the 63 games which were eventually played counted towards the participants' standing i.e. we cannot judge the efficiency of the model on the games that haven't been played.

Smaller log-loss scores are better, and the minimum log-loss score (i.e., picking all games correctly with probability 1) is 0. There are several unique aspects of this scoring system. Most importantly, the probabilities that minimize the log-loss function for a submission are the same as the probabilities that maximize the likelihood of a logistic regression function. As a result, we begin our prediction modeling by focusing on logistic regression.

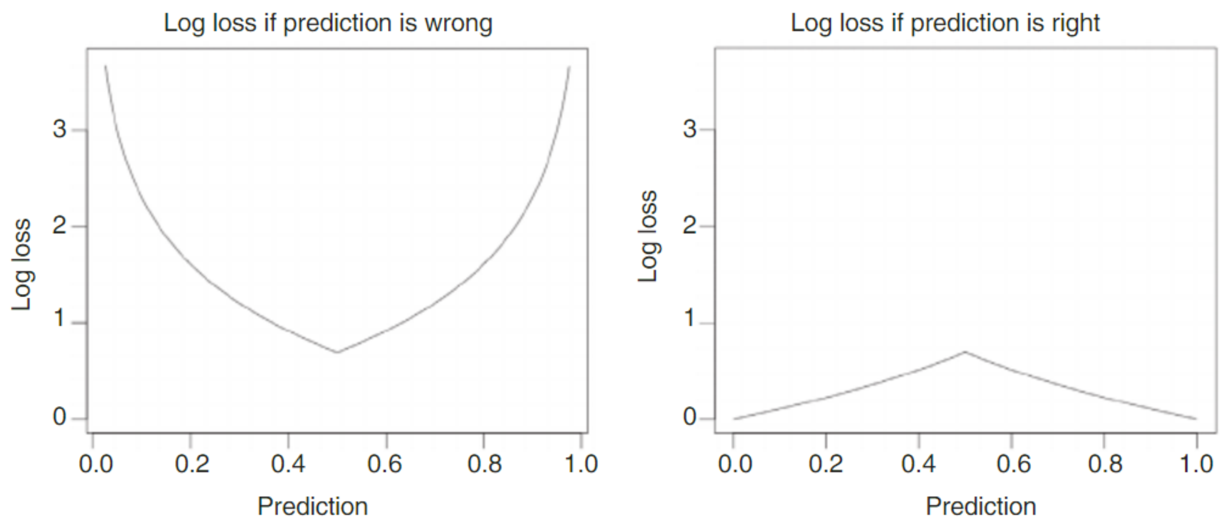


Figure Chart_1: Log loss as a function of the predicted probability. Note that when the prediction is wrong, log loss approaches infinity at both 0 and 1

Further, all games are weighted equally, meaning that the tournament's first game counts as much towards the final standings as the championship game. Finally, as each entry picks a probability associated with every possible contest, prior incorrect picks do not prevent a submission from scoring well in future games.

## 7.2 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 season and tournament games. 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 $n_i y_i \sim \mathrm{Bin}(n_i, \pi_i)$, 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

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

Considering the large number of covariates and relatively few data points, overfitting and multicollinearity were notable concerns. In the logistic regression model, we standardized all the feature values by season prior to model fitting. We also fitted a logistic regression model, including all features, to the training data. Covariates with significance levels <0.1 were then iteratively pruned to produce the final fitted model.

## 7.3 Logistic regression with Lasso regularization:

Lasso 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:

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

In Lasso regression, C controls the amount of shrinkage applied to the parameter estimates. We chose the optimal value of C by performing 10-fold cross validation over the training set, i.e., the regular season and tournament games of all preceding seasons, back to year 2010. In particular, we used the largest value of C within 1 standard error of the minimum, to prevent overfitting. The logistic regression model with lasso

regularization not only helped us identify the top features helpful in predicting the outcome of the match but also achieved lower log loss values when compared to the subsequent models.
Adjusted Offense, Adjusted Defense, Adjusted Tempo, SOS Difference, Location, Offensive Rebound Differential, Defensive Rebound Differential, Turnover Differential, Three Point Reliance, True Shooting Percentage, True Shooting Percentage Allowed and Block Differential were identified as the most important variables for match outcome prediction. All the subsequent models have been built using these selected features.

**7.4 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. The method derives its efficiency on the fact 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.
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 over-fitting, and also reduces computation time. This method was implemented using the gbm package 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 2010-11 through 2013–2014 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. Features obtained as a result of the backward feature selection were used as inputs to the model.

**7.5 Random Forest:**

A random forest is a bootstrap of the decision tree. Like in the standard bootstrap method, the available training data can be resampled to build many different decision trees. Additionally, the random forest method takes a random subset of available predictors to build its new decision tree. By resampling both the cases and the input variables, many decision trees are grown. Final classification decisions are made based on a majority rule of the many decision trees.

Given a decision tree with height k, let $P_r(r_d)$ be the probability associated with node d on branch path r. Then the probability of success for a new case is:

$$\hat{y}_i(A,B) = \frac{1}{m}\sum_j \prod_d Pr(r_{(d,j)})$$

There are two main reasons we would use Random Forests over Gradient Boosted Decision Trees:

RF are much easier to tune than GBM because RF basically has only one hyper-parameter to set i.e. the number of features to randomly select at each node and RF are harder to over-fit than GBM and they are more robust to overfitting and require less tuning to avoid it.

Random forest feature reduction proved to be highly successful, in terms of reduction of log loss. We were able to achieve fairly consistent results for training and test data with a log loss of around 0.63. Increasing the number of trees in the forest assists in a more accurate prediction of class probabilities by decreasing the effect of a non-congruous prediction of a single tree.

Going forward, different parameters could have been incorporated into the training of the model. Controlling the number of times a leaf has been split, or the number of samples that is required to split a leaf could have direct impacts on the results of the decision tree. Implementing these features would yield better results if the training size incorporated a greater number of years, which is the reason why these features remained untouched. Increasing the data set would thereby increase the number of parameters that could be optimized.

**7.6 Support Vector Machines:**

Next we fit an SVM classification model to our data set because it performs well for two class classification problems. For our model parameters, firstly we choose to use an RBF kernel, because we have a fairly low number of features and samples and the function can work well without overfitting. We are not really limited by this model since it is not a very high dimensional data set. We are using a low C penalty parameter value because, since a single elimination college basketball tournament is prone to a high number of upsets, we have a very high variance. To avoid overfitting, we choose to keep the default value of 1.0 for C. We set a probability output parameter, and thus our output is essentially a probability of being in each class with a corresponding confidence interval. We find this output to be more advantageous than giving a strict conclusive classification as to which class each data point will be in and will help us to get a more accurate result in our final ensemble step. Furthermore, we choose not to use any shrinking heuristics in our final SVM model. Although, it does increase the speed of optimization, it raises the probability of overfitting and similarly to our choice of a low value for c, we will prefer to be more conservative in our optimization path. As for our choice of weights, we choose to assign standard values of 1 to each of the two classes, since, due to the nature of our data set, a win for one team will automatically lead to another team losing.

Therefore, we should be equally cautious about misclassifying either wins or losses. We set the maximum number of iterations to 10,000.

For Support Vector Machine, we utilized the library LIBSVM to run the algorithm on our data. The primary advantage of SVM is that working in an infinite dimensional feature space and using the kernel trick allows to capture the interaction between features. This is particularly valuable for our dataset.

One disadvantage of SVM is that it is susceptible to overfitting because in working in an infinite dimensional feature space it is extremely likely that the extracted data will become linearly separable and hence the model will learn hidden attributes of our data rather than of the problem more generally.

Each game in the training set X has a series of characteristics based on the difference in team statistics and a label $y_i$ (win or loss). These can be plotted in a p-dimensional Euclidean space. The support vector machine then finds the hyperplanes w.X - b = 1 and w.X = 0 with the greatest distance between them that partitions the space such that wins and losses are separated (where w is the normal vector to the plane and $\frac{||w||}{b}$ is the offset of the hyperplane from the origin along w.

Let the output of an SVM be f (X). Then

$$\hat{y}_i^{(1)}(A,B) = \Pr(y_i = 1 | f(X)) = \frac{1}{1+\exp(Q \cdot f(X) + V)},$$

where Q and V are parameters chosen through maximum likelihood estimation from a training set made up of coordinates (f $(X)_i$, $y_i$).

We were pleased with our results for SVM. It did not perform as well as logistic regression in terms of average score, an artifact most likely due to an inadequate amount of data. However, SVM still correctly predicted the eventual tournament winners and performed well for the latter stages of tournament. This suggests that when the tournament results are more predictable, our current features are sufficient for learning a reasonably good model. We anticipate that if we were able to acquire more years of data our SVM would continue to improve.

**7.7 K- Nearest Neighbor:**

This method uses the same beginning framework as the support vector machine. We still have each training set game labeled according to the difference in team statistics. For each new game lacking a label, we can obtain all of the same difference statistics. Then, the labeled points closest in Euclidean space to the new point, “vote” to determine the label of the new game. For example, if 3 out of 5 nearest neighbors are winners, then the new point will be assigned to be a winner, or given a 0.6 probability of winning.

Given a candidate game $x_i \in \mathbb{R}^p$, find the k observations in X that are closest to $x_i$ in Euclidean distance. Let $(z_1,\ldots,z_k)$ be this set of observations, and $y_1,\ldots,y_k$ the corresponding outcomes. Then the k-Nearest Neighbor prediction is:

$$\hat{y}_i^{(3)}(A,B) = \frac{1}{k} \cdot \sum_{j=1}^{k} y_j$$

K- nearest neighbor classifier performed very similar to the seed benchmark understating the heavy reliance of the seeds in this classifier.

**7.8 Ensembling:**

Our final step used ensembling, in which individually produced classifiers are merged via a logistic regression. Ensembling methods allow predictions based on different models to be combined, often using a type of voting schema, to determine one final prediction. The benefit to using an ensemble method is that we can leverage the information gleaned from different methods, combining the benefits from each method while hopefully negating or minimizing the flaws in each. Ensemble methods have been used to combine predictions from different machine learning techniques in order to increase stability and to more accurately classify test cases. The combination of methods can also reduce the variance of the predictions as they are less sensitive to irregularities in a training set.

The most basic ensemble method is to average the results from a set of predictions. In this way, an ensemble method can avoid extreme predictions. Ideally, we want to assign optimal weights to each group of predictions. In order to ensure that our predictions were in range (0, 1), we fit a logistic regression model where each explanatory variable is a prediction from one machine learning technique, and the response is the binary outcome.

Our ensemble model takes the following form:

$$y_i(A,B) = \frac{1}{1+e^{-\left(\beta_0+\beta_1\cdot\hat{y}_i^{(1)}(A,B)+\cdots+\beta_6\cdot\hat{y}_i^{(6)}(A,B)+\varepsilon_i\right)}},$$

where each $\hat{y}_i^{(j)}(A,B)$ is a prediction from machine learning technique j for the probability that team A beats team B in game i. The fitted coefficients $\beta_j$'s are the optimal weights yielded by the logistic regression model. Table_1 below shows the output of the ensemble logistic regression for which the below mentioned model outputs serve as the inputs.

| j | Model | $\beta_j$ | 95% CI for $\beta_j$ |
|---|---|---|---|
| 0 | Intercept | -4.32 | [-5.11,-3.12] |
| 1 | Random Forest | 1.35 | [0.61,2.21] |
| 2 | KNN | 1.25 | [.42,2.21] |
| 3 | Gradient Boosting | 1.1 | [0.54,2.21] |
| 4 | SVM | 0.41 | [-0.21,1.11] |
| 5 | Logistic Regression | 0.32 | [0.23,1.01] |

Figure Table_1: Outputs of RF, KNN, Gradient Boosting, SVM and Logistic Regression serve as inputs for the Ensemble logit model.

## 8. Results:

Our bracket based on the ensemble model has a Logloss value of 0.58 for the training data while 0.61 for the testing data. We successfully predict the win/lose results on 50 out of 67 tournament games, and the model did better than the seed benchmark particularly for the latter stages of the tournament. We were badly hurt by a few games in which we had false confidence in the losing team. Some of these games were close (e.g. Arizona State lost to Texas by two points on a buzzer beater) and may be attributed to bad luck, but others were not (e.g. we thought UMass had a 97% chance to beat Tennessee, but they lost by 19 points). Unfortunately, we were not able to detect any obvious trends in these large deviance games. We were able to produce predictions that had a smaller log-loss than the three benchmarks set by Kaggle.

Table_2 shows the log loss value for the different benchmarks, individual models and the ensemble model.

|  | All Zeroes | All 0.5 Benchmark | Seed Benchmark | Logistic Regression | Random Forest | Gradient Boosting | SVM | KNN | Ensemble |
|---|---|---|---|---|---|---|---|---|---|
| Train | 19.19 | 0.693 | 0.63 | 0.625 | 0.631 | 0.65 | 0.621 | 0.63 | 0.58 |
| Test | 19.19 | 0.693 | 0.64 | 0.659 | 0.642 | 0.68 | 0.69 | 0.66 | 0.61 |

Figure Table_2: Log loss value for the test dataset is the lowest for the Ensemble model

SVM performed well on the training data but due to possible overfitting its performance on the test dataset dropped drastically. The limited size of data is one of the major challenges that led to overfitting of SVM. Logistic regression performed above expectation and one of the lessons that we learnt while doing this project is that it is not always the more complex model that gives better results.

We also came across many studies which evaluated the luck involved in winning a tournament pool with probability entries. These researchers performed simulation studies, assigning each entry a LogLoss score at many different realizations of the 2014 NCAA basketball tournament. The contest organizer provided each of the 433 submissions to the 2014 Kaggle contest for this evaluation. These entries were submitted by 248 unique teams; each team was allowed up to 2 entries, with only the team's best score used in the overall standings. The simulations indicated that the amount of luck required to win the Kaggle contest was enormous; even when the predictions were the true underlying game probabilities, those submissions only won roughly 1 in 8 times, respectively. Further, if our submissions were correct, we only finished in the top 10 about 49% and 45% of the time. A large number of unaccounted factors such as injury to key players, coach experience, and team dynamics can't be captured by the data which might have improved the model performance.

## 9. Conclusion:

In this paper, we presented a suite of models for predicting the outcome of the 2015 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 using log loss as our cost function. Across a wide range of models, we found that our best results came from an ensemble of all models tested using logistic regression to assign weights to each of the model outputs. Unfortunately, our models did not perform significantly better than our baseline predictions. This was disappointing but not surprising, as there is plenty of uncertainty involved in March Madness and chances of predicting a perfect bracket are near impossible. Also, March Madness games, with the single elimination format and amateur athletics, are more likely to end up as upsets over NBA playoff matchups where the better team typically wins over a series of matchups. Furthermore, our models used regular season game outcomes and statistics to predict 2015 Tournament results, which may have hurt us because predicting regular season games are fundamentally different from predicting tournament games. Some of our models like Decision Trees encountered significant overfitting issues due to this issue.

Also, we realized domain knowledge plays very high influence in improving model efficiency. In fact, most of the top 10 finishes relied highly on domain knowledge rather than model performance. In conclusion, we were not able to significantly increase our prediction accuracy over our baseline measure due to the complexity and uncertainty of the problem but came away with the following lessons for those about approaching this problem in the future: Pay careful attention to the issue of feature selection and deriving features, choose models that are appropriate for the quality and quantity of available training data, select model based on target cost function – log loss, and finally don't ignore simple models, as they may outperform complex models that are sensitive to many parameters.

## References:

1. Boulier, Bryan L. and Herman O. Stekler. 1999. "Are Sports Seedings Good Predictors?: An Evaluation." International Journal of Forecasting 15(1):83–91.
2. Brown, Mark and Joel Sokol. 2010. "An Improved LRMC Method for NCAA Basketball Prediction." Journal of Quantitative Analysis in Sports 6(3):1–23.
3. Bryan, Kevin, Michael Steinke, and Nick Wilkins. 2006. Upset Special: Are March Madness Upsets Predictable? Available at SSRN 899702.
4. Carlin, Bradley P. 1996. "Improved NCAA Basketball Tournament Modeling via Point Spread and Team Strength Information." The American Statistician 50(1):39–43.
5. Cesa-Bianchi, Nicolo and Gabor Lugosi. 2001. "Worst-Case Bounds for the Logarithmic Loss of Predictors." Machine Learning 43(3):247–264.
6. Cover, Thomas M. and Joy A Thomas. 2012. Elements of Information Theory. John Wiley & Sons, Inc., Hoboken, New Jersey.
7. 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.
8. ESPN. 2014. NCAA Division I Men's Basketball Statistics – 2013–14, 2014.
9. Friedman, J. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." Annals of statistics 2:1189–1232.
10. 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.
11. M. Greenfield. Team Rankings, 2014. https://www.teamrankings.com/
12. ESPN. 2014. NCAA Division I Men's Basketball Statistics – 2013–14, 2014. http://kenpom.com/index.php?s=RankAdjOE

**Code for the project is available at the following github webpage:**

https://github.com/surajut/NCAA-March-madness