Technical Report TR-20240429-1

Comparative Analysis of Artificial Intelligence Predicting the Outcome of March Madness

Pete Goldstein
Indiana University South Bend
Advisor: Dr. Dana Vrajitoru

I. Introduction

Every March, the National College Athletic Association (NCAA) hosts a single-elimination end of season tournament for Men's College Basketball called March Madness where the tournament winner is crowned as the National Champion. Using a selection committee, the NCAA selects 68 teams to receive bids (selected to participate) in this end of season tournament. There are two types of bids the committee uses. The first is an automatic bid. 32 teams will receive an automatic bid by winning their respective conference's tournament[1] [1]. The remaining 36 bids are considered at-large [2], where the selection committee chooses the remaining best 36 teams across all conferences after all conference tournaments have concluded. These 68 teams are then ranked and seeded from 1 to 16 by the selection committee.

Once seeded, teams are placed in four geographical regions, namely East, West, South, and Midwest. Each of the regions will have a single play-in game [3]. The winner of each play-in game will then enter the round of 64 where the win or go home format continues. At the end of each round, exactly half of the teams are eliminated. This pattern continues until the last two teams play, where the winner of the last game is deemed the National Champion.

While the tournament is used to crown the champion of college basketball, the tournament offers the public an opportunity to compete in the bracket pool challenge where it is estimated that around 50 million people participate [4]. In a bracket pool, every entrant must completely fill-out their bracket from the Round of 64 to the National Championship [2] of their predicted outcomes prior to the start of the round of 64. This lends us the question, can Machine Learning successfully predict the outcome of tournament matches and win a bracket pool? In this paper, we explore the use of three different Machine Learning algorithms to answer this question.

[1]A single elimination tournament comprised of teams in the conference which takes place at the conclusion of regular season play

II. Background information

In this section, we will provide details to help understand the subject of our objective.

A. NCAA Tournament

The Division 1 NCAA basketball tournament, often referred to as "March Madness," is a single-elimination tournament featuring 68 college basketball teams from universities and colleges with the following format and structure:

- Selection Process: The tournament field consists of 68 teams that are selected through a combination of automatic bids and at-large selections. The conference champions are award automatic bids, while the NCAA Selection Committee chooses at-large selections based on a team's performances throughout the season, and various other factors.
- Seeding: Once the field is determined, the Selection Committee seeds the teams

[2]Play-in games are typically omitted.

from 1 to 16 within each of the four regions. The Selection Committee determines the team's seeding based on the team's strength, record, and other criteria.

- Bracket: Each tournament in the bracket format is organized by placing each team in a specific spot within their respective region. The bracket consists of four regions: East, West, Midwest, and South.
- Neutral site: Each tournament match-up takes place at a neutral venue.
- First Four: The tournament begins with the First Four, a series of play-in games between the lowest-seeded teams (typically seeds 65-68). These games determine which teams advance to the first round of the tournament.
- First Round: Following the First Four, the tournament moves into the first round, where the remaining 64 teams compete. This round features 32 games, with each winner advancing to the next round.
- Second Round (Round of 64): The winners of the first-round games advance to the second round, also known as the Round of 64. In this round, 32 games are played, with the winners moving on to the third round.
- Third Round (Round of 32): The third round consists of 16 games, with the winners advancing to the Sweet 16.
- Regional Semifinals and Finals (Sweet 16 and Elite 8): The tournament then moves to the regional semifinals (Sweet 16) and finals (Elite 8). The remaining teams compete in their respective regions, with the winners advancing to the Final 4.
- Final 4: The winners of the four regional finals advance to the Final 4, which is usually held at a predetermined neutral site. Here, the remaining four teams compete in two semifinal games.
- National Championship Game: The winners of the semifinal games face off in the National Championship Game to determine the overall champion of the NCAA Division 1 men's basketball tournament.

B. Bracket Pool

In a bracket pool, a group of participants create and fill out their brackets by selecting the teams they predict will win each match-up throughout the tournament, from the Round of 64 through to the National Championship Game. To determine a winner of a pool, participants establish a scoring system before the tournament begins. Points are awarded for each correct prediction, with the number of points typically increasing as the tournament progresses. For example, correct predictions in the early rounds might be worth fewer points than correct predictions in later rounds.

III. Literature Review

The literature on Men's College Basketball has primarily concentrated on two areas. The first of the two is the development of a more defined seeding system, since the selection committee is the sole decider on at-large bids. The second is the predicted outcomes of NCAA Men's basketball games. For this one, we focused on the latter.

Kvam and Sokol [5] used a Markov chain with transition probability derived using logistic regression with the goal of determining the probability that Team A is better than Team B. In their paper, the transition probabilities for every team were created using only match-ups between two teams in a given season where the teams had home and home series (i.e., Team A plays Team B twice in a season, once at home and once at Team B); using the margin of victory with respect to the home team, a factor was derived to represent a home court advantage to account for neutral site match-ups.

Beal, Norman, and Ramchurn [6] compared 9 different machine learning algorithms to predict the outcome of NFL games. A data set of 1280 games containing 85 input variables, their comparison yielded two algorithms that outperformed predictions from odds makers, namely the AdaBoost and Naïve Bayes algorithm. Although both algorithms would successfully beat the odds makers, the Naïve Bayes algorithm

produced a better accuracy, recall and precision over the AdaBoost algorithm.

Schwertman et al. [7] tested 11 different ordinary least squares and logistic regression models. They found that an ordinary least-squares regression model to determine a teams probability of winning the tournament would be best. Also using linear regression, N.E.O and Uzoma [8] proposed an hyrbrid model, feeding the results of a linear regression model with 21 features to a K-Nearest Neighbor Algorithm.

IV. Data Collection and Data Set Creation

Prior to the initiation of our project, our research led us to several potential data sets. There, we found Lopez and Matthews[9] who utilized efficiency statistics and Las Vegas spread data to train a Machine learning algorithm to predict the outcome of the round of 64 most promising. Further backing this approach, Kubatko et. al.[10] says efficiency statistics can be used as a starting point in comparing the two teams. While we originally set out to replicate Lopez and Matthews' experiment, the parameters of our experiment differed slightly. That is, during the time frame one would predict their outcomes, Las Vegas will only have spread data published for the round of 64 as match-ups in succeeding rounds are yet to be determined. Using spread data, we create models that are incompatible with the constraints of the questioned posed. Thus, our data set will be composed of only efficiency statistics.

A. Data Collection: Efficiency statistics

Once a subscription was purchased to Kenpom[11], we were able to obtain pre-tournament[3] efficiency statistics from 2008-2021[4] through a csv download for each year. Each of these files contains

- Year: The year for which the statistics were calculated
- TeamName: Team for which the statistics are calculated for
- Tempo: The team's expected number of possessions per 40 minutes
- Tempo Rank: A ranking of a team's tempo (out of approximately 360 teams)
- Adjusted Tempo: The team's expected number of possessions per 40 minutes against an average team
- Adjusted Tempo Ranking: A ranking of a team's adjusted tempo
- Offensive Efficiency: A team's expected number of points scored per 100 possessions
- Offensive Efficiency Ranking: A ranking of a team's Offensive Efficiency
- Adjusted Offensive Efficiency: A team's expected number of points scored per 100 possessions against an average team
- Defensive Efficiency: A team's expected points allowed per 100 possessions
- Defensive Efficiency Rank: A ranking of a team's Defensive Efficiency
- Adjusted Defensive Efficiency: A team's expected points allowed per 100 possessions
- Adjusted Defensive Efficiency Rank: A ranking of a team's Adjusted Defensive Efficiency
- Adjusted Efficiency Margin: The difference of a team's Adjusted Offensive Efficiency and Adjusted Defensive Efficiency
- Adjusted Efficiency Margin Rank: A ranking of a team's Adjusted Efficiency Margin
- Seed: A team's seeding in the NCAA tournament

B. Data Collection: Historical match-ups

Next, we looked for historical match-up data. Using SportsBookReviewOnline [12] we were able to freely download an.xlsx file for each season from 2008-2019 and 2021 containing every match-up of a season. Here, the fields that comprise the files are:

- Date: Date of the match-up
- rot: No information pertaining to this field
- VH: Indicates if the team found in a given row is the visiting or home team
- Team: Team name of the given record

[3]Efficiency Statistics without prior to the start of the tournament

[4]2020 did not have a tournament due to Covid-19

- 1st: Total points score in the first half for the team in a given match-up
- 2nd: Total points score in the second half for the team in a given match-up
- Final: The total points score, sum of 1st and 2nd, for the team in a given match-up
- Open: Gambling odds first publish, Over/under total points if VH is visitor, spread if VH is home
- Close: Gambling odds at beginning of match-up, Over/under total points if VH is visitor, spread if VH is home
- ML: Gambling odds, The odds of a team winning the match-up
- 2h: Gambling odds for the second half of a match-up, Over/under total points if VH is visitor, spread if VH is home

C. Data Set Creation: Historical Match-ups

With both datasets, we first set out to create an indicator for each historical match-up. Although the field *rot* was not critical to the analysis, we identified that it was incremental. Increasing by a value of one in each row. Leveraging this property, we created a new field using modular arithmetic to find the mod2 value of a *rot* of a given row, multiplying the remainder by *rot*. As a result, a given row would have a new field containing either the *rot* or zero (0). We then review the *rot* of the first row, of each file, to determine a cadence. Where if the initial row != 0, the calculated value is assigned to the succeeding row. On the contrary, if the initial row = 0, we can assign the value calculated in the subsequent row to the current row. In doing so, we created a indicator, pairing rows to identify the two teams of a given match-up.

Continuing with our historical match-up data set, we then took advantage of the *Final* field which contains the total points scored for the team found in the row of a given match-up. Using the created indicator field, we found the difference in the final score of each team. A negative difference would signify a loss, and a positive difference would signify a win. This allowed us to classify a given record as an 'L' for loss or 'W' for win.

Following the creation of the indicator and record classification, we turned our focus to filtering our data to games played on or after March 1st. The historical match-up data set represented dates as integers. For example, November 20th would be written as 1120 and January 24th written as 124. Using this format, we preformed two integer comparisons. First, we excluded records with a date greater than 500, as the tournament takes place in March (300) and April (400). Next, we excluded records less than or equal to 299. This leaves us with all the match-ups that take place on or after March 1st. Upon the filtering, we removed the rot, VH, 1st, 2nd, Final, Open, Close, ML, and 2H fields as they were not involved in the analysis.

D. Data Set Creation: Efficiency statistics

The data obtained from Kenpom required minimal adjustments. We merely needed to remove the ranking fields, as they were not included in the analysis.

E. Data Set Creation: File Merging

The next phase of our data set creation was the merging of our two independent data sets to associate a team and match-up classifications with their efficiency statistics for the given season. To do so we aimed to use Team and TeamName fields of the historical match-up and efficiency data sets, respectfully. This was seemingly straightforward with the caveat that team names are often abbreviated. These abbreviations can vary depending on the author, and in our case varied drastically.

For this, we opted to utilize a dictionary, creating a master key:value pair that would map team names from our efficiency to the team names found in our historical match-up data. More precisely, a key value pair of TeamName:Team. To begin iteratively adding key value pairs, we walked through each file for a given year, adding to our mapping of teams that were able to map as is. Then, we performed a string search of characters of all team names found in our efficiency data. We then reviewed the string search and identified

a potential mapping. At the conclusion of the string search, if any teams were unmapped for a given year, we would manually review and add the appropriate mapping. Prior to adding mappings, we validated the mapping by performing an internet search of a historical match-up found in our data set.

Using the mapping, we added the Team name from our historical data to our efficiency data, for each year of data. In doing so, we were able to join the data sets on the Team name, creating a data set for each year.

F. Data Set Creation: Opposition Efficiency

With a single data set for each year, we utilized the match-up indicator to pair teams in a given match-up. Adding the efficiency statistics of the given opponent.

G. Data Set Creation: Training and testing sets

Given the format of the tournament, a typical training-testing split of 80-20 could prevent the model from being trained on instances from every round. To combat this, we opted to view each year of the data as an instance and randomly select two of the thirteen years as testing sets. We selected two years by using a random number generator selecting a number between 2008 and 2021. Resulting in the selection of the 2010 and 2014 NCAA tournaments. For testing purposes, these test sets where not joined. After selecting which years would comprise the test set, we combined the remaining years into a single data set.

H. Data Set Creation: Tournament games

An important aspect of our models is that each match takes place at a neutral venue, eliminating the home court advantage of a team [13]. The timeline of the model selection and training portion of our project happened to coincide with 2024 tournament and the time frame that our data set is based on. To be one with our project, we watched a handful of match-ups. We then realized that our data set contains instances that may skew our models. That is, that not all match-ups on or after March 1st take place at neutral venues, but rather some teams are still playing in match-ups at home or as a visitor. To prevent the models from being skewed by home court advantages, we removed all non-tournament match-ups. This allowed our models to be trained on neutral site tournament match-ups only.

I. Adjusted Efficiency

Data obtained by Kenpom contained Efficiency and Adjusted Efficiency metrics. As previously mentioned, adjusted is the expected value of a metric against the average team, and normalizing the metric towards the average of a given year. Fig. 1 shows that AdjTempo has fewer outliers due to normalization. Based on this, we centered our analysis on the adjusted efficiency metrics.

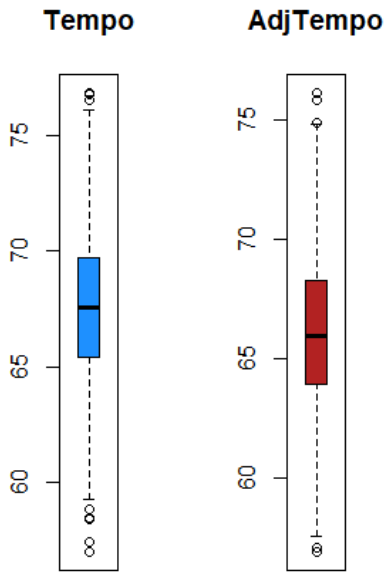

Fig. 1. Comparison of the distribution of Tempo and Adjusted Tempo

V. Data Set

The data set used to train our model consisted of 11 years worth of NCAA tournament match-ups between 2008 and 2021, excluding

2020. Omitting the play-in match-ups to mirror our objective, each tournament consists of 64 teams and a total of 63 match-ups, yielding 126 instances each year[5], where the split in classifications is even, 50% Win and 50% Lose. In total, our models were trained on 1384 instances.

A given instance consists of the following fields.

- *Adjusted Tempo*: A numerical field signifying the number of possessions per game against an average opponent. (AdjTempo)
- *Adjusted Offensive Efficiency*: A numerical field signifying the number of points scored per 100 possessions against an average opponent. (AdjOE)
- *Adjusted Defensive Efficiency*: A numerical field signifying the number of points allowed per 100 possessions against an average opponent. (AdjDE)
- *Adjusted Efficiency Margin*: A numerical field signifying the difference of a team's Adjusted Offensive Efficiency and Adjusted Defensive Efficiency. (AdjEM)
- *Opponent Adjusted Tempo*: The opposing team's Adjusted Tempo (OppAdjTempo)
- *Opponent Adjusted Offensive Efficiency*: The opposing team's Adjusted Offensive Efficiency (OppAdjOE)
- *Opponent Adjusted Defensive Efficiency*: The opposing team's Adjusted Defensive Efficiency (OppAdjDE)
- *Opponent Adjusted Efficiency Margin*: The Adjusted Efficiency Margin of the opposing team (OppAdjEM)

To prevent the models from being trained on data identifiers, each observation of our data set is centred around a singular team, and their opponent for a given match-up.

VI. Prediction Scoring

Determining the winner of a pool is based on the scoring which is determined prior to the start of the tournament. To answer our question we must also determine a scoring system to judge the bracket our algorithms predicted. While there is a number of ways to score a bracket, we used a progressive scoring system. For each correct prediction we received,

- Round of 64: 1 point
- Round of 32: 2 point
- Sweet 16: 3 point
- Elite 8: 4 point
- Final 4: 5 point
- National Championship: 6 point

VII. Model Bench Marking

While scoring helped us compare models and their predictive power, we also need a comparison of a bracket that is not predicted by machine learning algorithms. In doing so, we can gain insight into how our models may perform against humans. To do this, we employed two different control brackets. The first was a bracket where the better seed always wins. In this control, we are guaranteed a match-up between two teams with the same seed. In this event we will take the team, that is listed on top from the bracket found on NCAA.com [14] [15]. Our second control bracket can be found in the White House archives[16] [17] and will be the 2010 and 2014 NCAA brackets of President Barack Obama.

By choosing these brackets as our control, we can emulate potential brackets of competitors. A our bracket where the higher seed always wins represents someone with no knowledge of college basketball [18], while Obama's bracket represents someone with moderate knowledge of college basketball. Ideally, we would include a college basketball expert in the control group. However, college basketball often creates and publishes multiple brackets, introducing a selection bias in our control group. Furthermore, the common bracket pool in not inclusive of college basketball experts, thus an unfair comparison to our objective.

VIII. Models

For our comparison, we evaluated three different machine learning algorithms.

[5] In 2021, Virginia Commonwealth forfeited their Round of 64 match-up to Oregon. This match-up is not included

A. Logistic Regression

In 2014, Lopez and Matthews [9] combined two logistic regression to win a Kaggle competition [19] where contestants' models were evaluated on the logarithmic loss of their predicted results using probabilities of a given Team A defeating Team B.

B. Classification Decision Tree

In 2012, Delen, Cogdell, and Kasap [20] used the CRISP-DM methodology to compare three different machine learning algorithms to predict the outcome of NCAA college football games. Using a data set of 244 bowl games from 2002-2009 where each instance contained 28 dependent variables, they found that a Classification and Regression Tree best predicted the results of bowl games.

C. Support Vector Machine

In 2016, Shen, Gao, Wen, and Magel [21] used 3 different algorithms: A Bayesian Model, Support Vector Machine (SVM), and a Random Forrest. In the two tournaments used as testing data, they found that the SVM produced the best results in one, and the Random Forest performed better in the other. Given the split, we chose the SVM because it produced the highest accuracy for any given model. In the year it performed the best it predicting the winner of a given match-up with 79.4% accuracy.

IX. Exploratory Data Analysis

To enhance our models' predictive powers, we evaluated the distribution of the data. Here, we found two variables that could be transformed to provide a more approximately normal distribution.

Fig. 2 illustrates the distributions of our variables, showing the presence of outliers. Upon this discovery, we reviewed our distributions using a normal Q-Q plot.

Adjusted Offensive Efficiency (Fig. 3) displayed signs of concave downwards distribution. We began by applying data transformations to create a more linear QQ line. Although

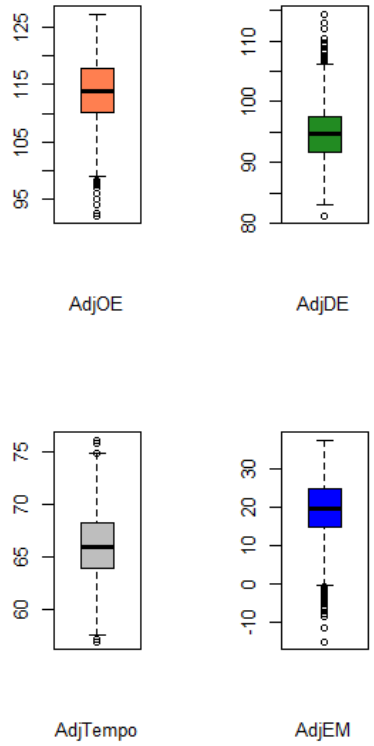

Fig. 2. Comparison of the distribution of the AdjOE,AdjDE, AdjTempo, and AdjEM

our right tail deviated from the QQ line, we were able to improve the fit of the left tail by squaring the data, creating a more normalized distribution set.

The adjusted efficiency margin presented a more complex distribution (Fig. 4). Although concave down, squaring the data resulted in a worse fit to the QQ line. When evaluating the raw data, we found that the adjusted efficiency margin contained negative values and squaring would create skewness. By nominally shifting the data by the doubling the absolute minimum, we began applying factional powers. 9/5ths was chosen as it produced the best fit to the QQ line.

Although adjusted defensive efficiency and adjusted tempo both presented outliers and did not have an approximately normal distribution, our efforts to transform these variables were unsuccessful.

Though our data set contains 8 variables, 4 of the 8 are merely 1 of the 4 adjusted metrics rearranged. For example, in a match-

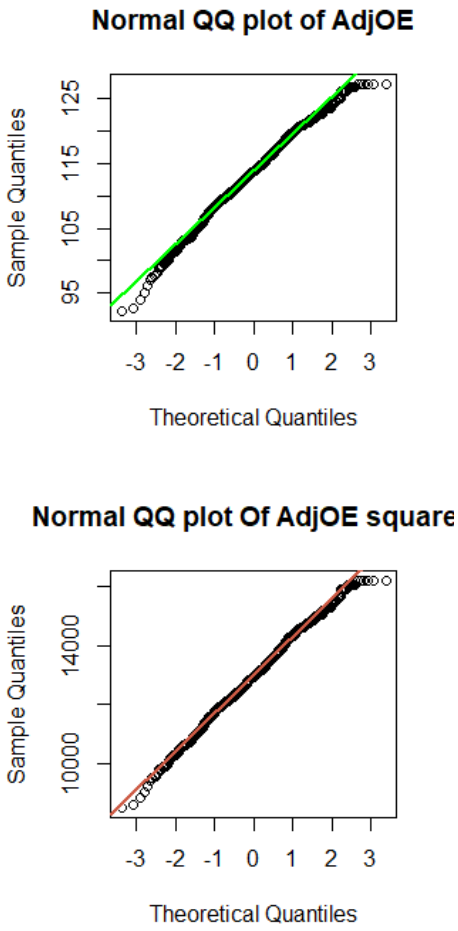

Fig. 3. Comparison of the Normal QQ plots of AdjOE and AdjOE squared

up with team A playing team B, suppose that team A has an AdjEM of 14.54 and Team B an AdjEM of 21.2. Then the OppAdjEM for team A is 21.2 and the OppAdjEM of team B is 14.54. Therefore, the distribution of the OppAdj metrics are the same distribution of the Adj metrics. This fact allowed us to ignore the distributions of the OppAdj metrics and apply the transformations found for the Adj metrics to the OppAdj.

To preserve our data set and prevent confusion of future evaluation of models, we created four new variables and remove the original data prior to training. Namely,

- TransAdjEM: Transformed AdjEM
- TransOppAdjEM: Transformed OppAdjEM
- TransAdjOE: Transformed AdjOE
- TransOppAdjOE: Transformed OppAdjOE

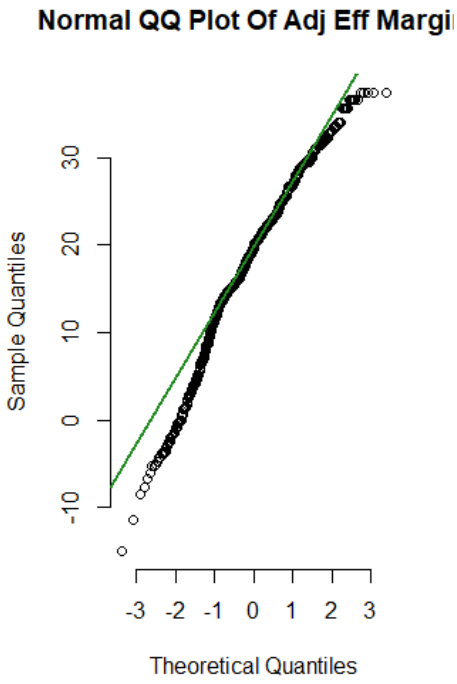

Fig. 4. Normal QQ plot of the Adjusted Efficiency Margin

Following data transformations, we performed review of the Variance Inflation Factor of our variables to test and remove any co-linearity. We found that TransAdjOE and AdjDE presented signs of co-linearity with TransAdjEM and OppAdjEM and removed them from the training set.

X. Model Selection

Prior to training our models, we needed to determine if all data points were necessary for our analysis. To perform this, we performed a 5-fold Cross Validation[22] lasso regression for logistic model and 5-fold Cross Validation recursive feature elimination for our Decision Tree and SVM. As 5-folds outperformed other common folds.

A. Lasso Regression

When applying lasso regression to our problem we found only 4 features were necessary for our logistic regression model, that TransAdjOE and OppAdjDE had zero coefficients, leaving us with the following equation:

$$\hat{y} = -3.437427e-06 + 0.0029AdjTempo + -0.0029OppAdjTempo + 0.0026AdjEM + -0.0026OppAdjEM \quad (1)$$

B. Recursive Feature Elimination

To utilize recursive feature elimination (RFE), the algorithm required us to define a metric to evaluate the data subsets and to determine the best model, given the data set. Given that we need to preform a binary classification, we chose to use Receiver Operator Characteristics (ROC).

For our decision tree (Fig. 5), RFE turned out to be the best model for only TransAdjEM and TransOppAdjEM with a ROC of 0.6864.

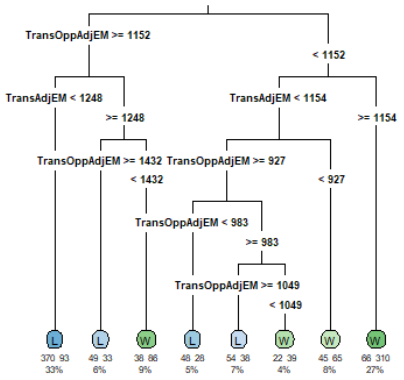

Fig. 5. Trained Classification Tree

For our SVM, RFE was found to be the best model to use the six variables. It produce an ROC of 0.7801.

XI. Testing

To effectively analyze these algorithms and their ability to predict brackets, we needed to develop a testing structure that would recreate the tournament, but in a data frame format. Using the subset of the Round of 64 games, we added a game key for each subsequent round. That is, after we predict the outcome of the round of 64 match-ups, we would be left with 32 teams and 16 plausible match-ups based on the bracket of the given year. For example, in 2014 the winner of Michigan vs. Wofford would be matched-up against the winner of Texas and Arizona State. For these four teams, we added a second round key of 16, indicating that the winner of these match-ups would have played in the 16th game of the Round of 32.

Given the format of our data set, we must mention that we may have match-ups where both teams are predicted to either win or lose. Since match-ups in the tournament cannot end in a tie, we must also implement a tie breaker. For this, we extracted the probabilities of a models predictions. In the event of a tie, the team with the highest probability of winning is chosen as the winner. And should the probabilities be the same, the lower seed will advance.

Additionally, we also had to add the results for the subsequent rounds. By manually reviewing the tournament, we added the correct result for each team for each subsequent round after the Round of 64. For example, in 2010, Georgia Tech defeated Oklahoma State in the Round of 64. In turn, the results of all possible subsequent matches for Oklahoma State would be classified as a loss.

This strategy allowed us to effectively match-up teams based on the bracket and test our models using the match-ups it had previously predicted.

XII. Results

In this section, we will simultaneously present the results of our models for each round and in order of the rounds in the tournament. They are presented in chronological order.

A. 2010 Tournament

After the initial phase of testing, our 3 models performed as expected on the basis of our training. Fig. 6 shows the confusion matrices of the results from each model for 2010 Round of 64 match-ups. Here our logistic regression model successfully predicted twenty-four of

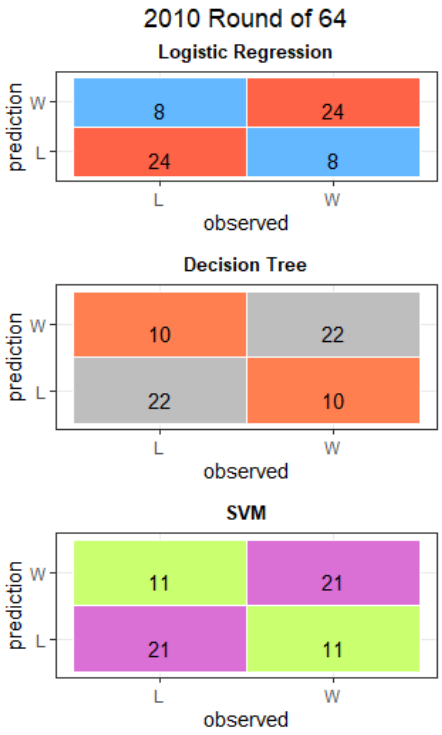

Fig. 6. Confusion Matrix of 2010 Round of 64 predictions Logistic Regression, Decision Tree and SVM classifiers.

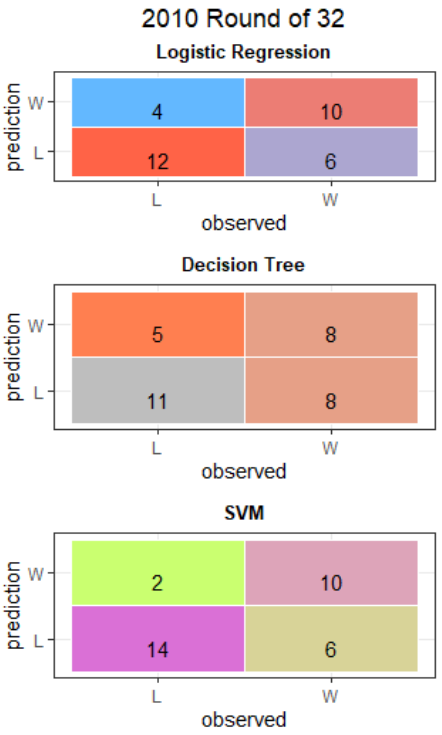

Fig. 7. Confusion Matrix of 2010 Round of 32 predictions Logistic Regression, Decision Tree and SVM classifiers.

thirty-two match-ups in the 2010 first round. The SVM model successfully predicted twenty-one of thirty-two match-ups in the 2010 first round. For our decision tree, we successfully predicted 22 out of 32 match-ups in the 2010 first round.

TABLE I
2010 Round of 64 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	75%	75%	25%	25%
Decision Tree	65.6%	65.6%	34.4%	34.4%
SVM	68.75%	68.75%	31.25%	31.25%

In addition to the accuracy of our models, we want to keep track of how our bracket performed. After the first round, the logistic regression model produced a total of 24 points, our decision tree had 22 points, and our SVM had a score of 21 points.

In the Round of 32, we see that the logistic regression model and the SVM successfully predict 10 out of 16 winners correctly. However, the decision tree only predicted 8 of the 16 correct winners.

For scoring in this round, all correct predictions provide two points. Thus, the logistic regression and SVM models receive 20 points, and the decision tree receives 16. This gave us the running total of logistic regression having 44, Decision Tree having 38 and the SVM having 41 points.

TABLE II
2010 2nd Round Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	68.75%	62.5%	25%	37.5%
Decision Tree	59.3%	50%	32.25%	50%
SVM	75%	62.5%	12.4%	37.5%

In the 2010 Sweet 16, we found that the SVM had the best predicting powers of this round, predicting five of the eight correct winners. Meanwhile, our Logistic Regression and Decision Tree model both correctly predicted four of eight winners correctly.

TABLE III
2010 Sweet 16 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	62.5%	50%	25%	50%
Decision Tree	68.75%	50%	12.5%	50%
SVM	75%	62.5%	12.5%	37.5%

At the end of the round, our brackets will

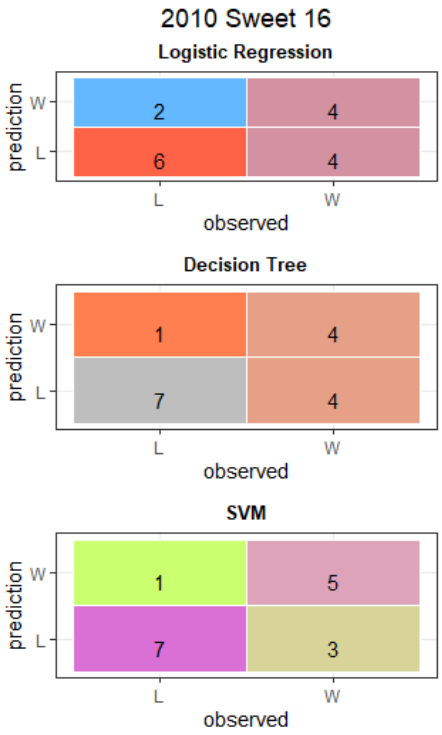

Fig. 8. Confusion Matrix of 2010 Sweet 16 predictions Logistic Regression, Decision Tree and SVM classifiers.

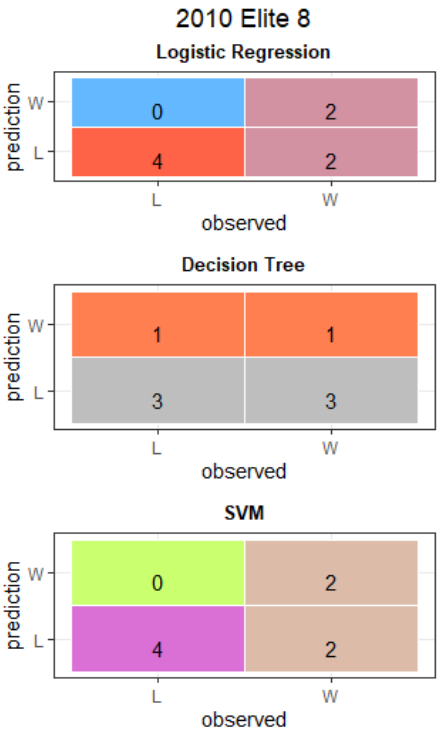

Fig. 9. Confusion Matrix of 2010 Elite 8 predictions Logistic Regression, Decision Tree and SVM classifiers.

have yielded 15 points for our SVM, bringing their total to 56 points. 12 points were for both our logistic regression and decision tree, giving each a total of 56 and 50, respectively.

With the next round as the Elite 8, we again see that our logistic regression and SVM produce the same confusion matrix, correctly predicting two out of four winners. The decision tree only predicted one of the four correct winners.

TABLE IV
2010 Elite 8 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	75%	50%	0%	50%
Decision Tree	50%	25%	25%	75%
SVM	75%	50%	0%	50%

For the round, the logistic regression and the SVM model earned eight points, giving them both a total of 64 points. The decision tree earned 4 points and has a total of 54 points.

The results of the 2010 Final 4 were the same for all three models, where each correctly predicted one of the two winners (Fig. 10).

In the Final 4, yielding 5 points for each correct prediction, each of the three models

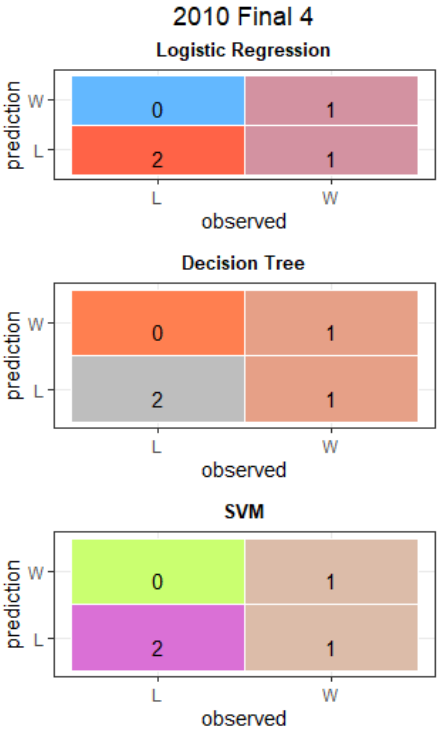

Fig. 10. Confusion Matrix of 2010 Final 4 predictions Logistic Regression, Decision Tree and SVM classifiers.

received 5 points, bringing their totals to 69, 69, and 59 for the Logistic Regression, SVM, and Decision Tree, respectively.

In the final round of the 2010 tournament,

TABLE V
2010 Final 4 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	75%	50%	0%	50%
Decision Tree	75%	50%	0%	50%
SVM	75%	50%	0%	50%

we again see all three models perform the same way. None of them predicted the correct winner (Fig. 11).

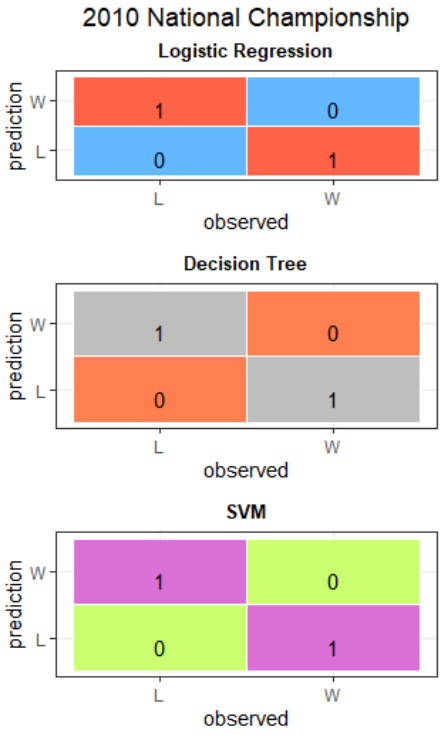

Fig. 11. Confusion Matrix of 2010 National Championship predictions Logistic Regression, Decision Tree and SVM classifiers.

TABLE VI
2010 National Championship Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	0%	0%	100%	100%
Decision Tree	0%	0%	100%	100%
SVM	0%	0%	100%	100%

As a result, none of the models received points for this round, leaving the Logistic Regression and SVM model tied at 69 points, and the decision tree with 59 points.

Tables VIII and IX provide a breakdown of each model on how they fared in each round. Additionally, after testing each round, we rolled up the results for each model to present a singular confusion matrix (Fig. 12) for each model.

TABLE VII
2010 Correctly Predicted Wins by Round

Model	Rd64	Rd32	S16	E8	F4	Final
Logistic	24	10	4	2	1	0
Decision Tree	22	8	4	1	1	0
SVM	21	10	5	2	1	0

TABLE VIII
2010 Score by Round

Model	Rd64	Rd32	S16	E8	F4	Final
Logistic	24	20	12	8	5	0
Decision Tree	22	16	12	4	5	0
SVM	21	20	15	8	5	0

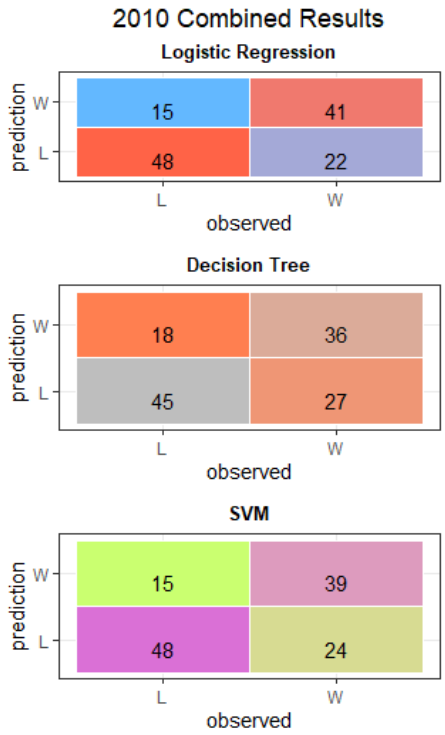

Fig. 12. Confusion Matrix of the combined 2010 predictions Logistic Regression, Decision Tree and SVM classifiers.

B. 2014 Tournament

We will now present the results of the testing from the 2014 Test set.

For the first round (Fig. 13), we see that the Decision tree performed best, correctly predicting twenty-five of the thirty-two winners. The logistic regression and SVM models both

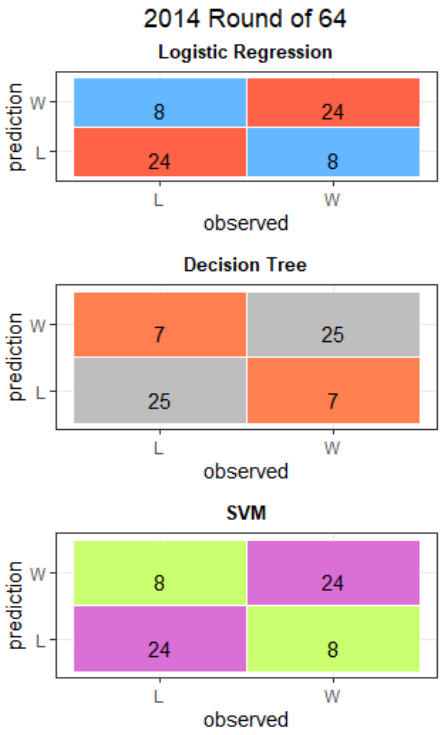

Fig. 13. Confusion Matrix of 2014 Round of 64 predictions Logistic Regression, Decision Tree and SVM classifiers.

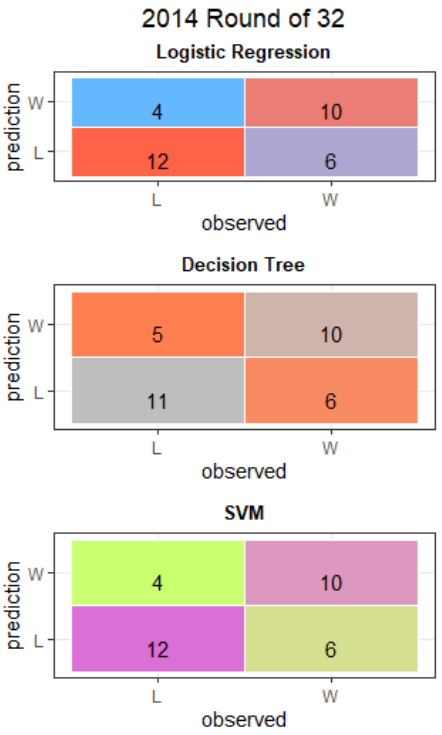

Fig. 14. Confusion Matrix of 2014 Round of 32 predictions Logistic Regression, Decision Tree and SVM classifiers.

correctly predicted twenty-four of thirty-two winners.

TABLE IX
2014 Round of 64 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	75%	75%	25%	25%
Decision Tree	78.1%	78.1%	21.9%	21.9%
SVM	75%	75%	25%	25%

This gives the decision tree a total of 25 points, and the logistic regression and SVM models a total of 24 points.

The 2014 Round of 32 (Fig. 14 and Table X) saw a tie in performance, where all three models successfully predicted the same number of correct winners, ten out of sixteen.

TABLE X
2014 Round of 32 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	68.75%	62.5%	25%	37.5%
Decision Tree	65.6%	62.5%	31.25%	37.5%
SVM	68.75%	62.5%	25%	37.5%

With each correct prediction receiving two points, there were no changes in the standings of our brackets. Updating the points, the Decision Tree had 35 points, and the Logistic Regression and SVM had 34 points.

As a result, all three models received 20 points for the round. We got a total of 45, 44, and 44 points for the Decision Tree, SVM, and Logistic Regression models, respectively.

Unlike 2010 where all three models saw a TPR of at least 50%, the 2014 Sweet 16 (Fig. 15 and Table XI) the decision tree was the only model with exactly 50% and our logistic regression and SVM had a TPR of 25%. This means that only two of the eight winning teams were correctly predicted.

TABLE XI
2014 Sweet 16 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	43.75%	25%	37.5%	75%
Decision Tree	68.75%	50%	12.5%	50%
SVM	43.75%	25%	37.5%	75%

Following the Sweet 16, the decision tree widened the gap between itself and the other two models, earning a total of 12 points in the round. Our decision tree has a total of

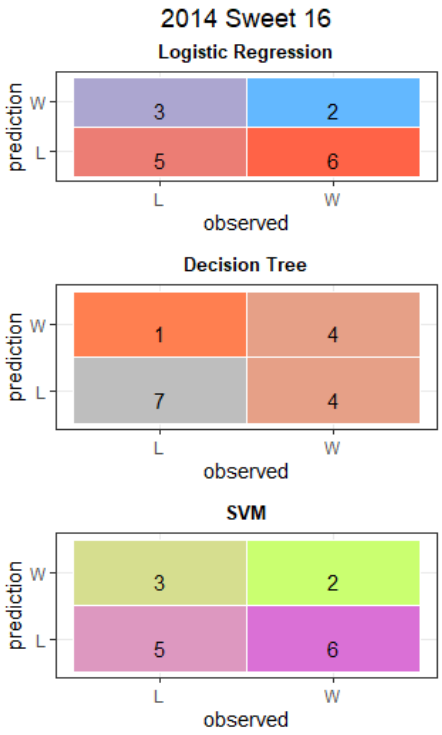

Fig. 15. Confusion Matrix of 2014 Sweet 16 predictions Logistic Regression, Decision Tree and SVM classifiers.

57 points, and the logistic regression and SVM totaling 50.

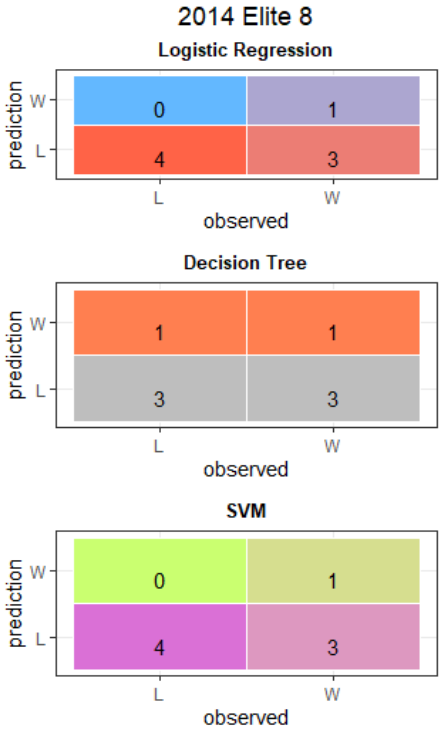

Fig. 16. Confusion Matrix of 2014 Elite 8 predictions Logistic Regression, Decision Tree and SVM classifiers.

From Fig. 16, each of our 3 models correctly predicted only one of the four actual winners. This means that each model received 4 points each, making our totals 61, 54, and 54 for the Decision Tree, Logistic Regression and SVM models, respectively.

TABLE XII
2014 Elite 8 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	62.5%	25%	0%	75%
Decision Tree	50%	25%	25%	75%
SVM	62.5%	25%	0%	75%

In Fig. 17, we find that none of the models produced any correct predictions. This means that there are no points awarded. Furthermore, since we did not have any correct predictions, our bracket can no longer earn any points.

Fig. 17. Confusion Matrix of 2014 Final 4 predictions Logistic Regression, Decision Tree and SVM classifiers.

TABLE XIII
2014 Final 4 Confusion Matrix Metric

Model	Acc	TPR	FPR	FNR
Logistic	50%	0%	0%	100%
Decision Tree	50%	0%	0%	100%
SVM	50%	0%	0%	100%

For all intents and purposes, we will present the testing results for the final round in Fig. 18.

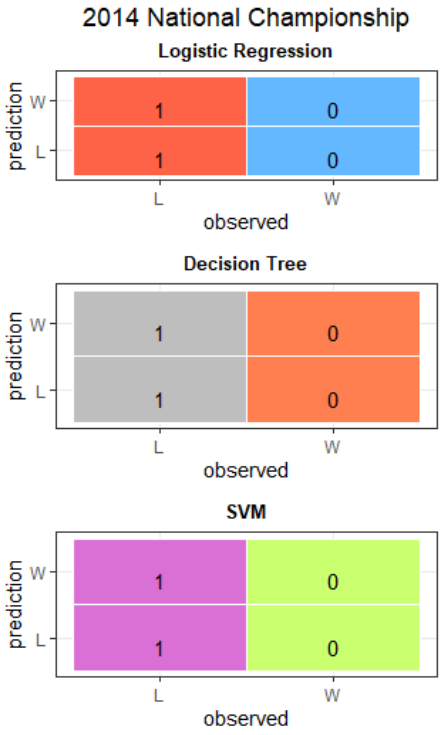

Fig. 18. Confusion Matrix of 2014 National Championship predictions Logistic Regression, Decision Tree and SVM classifiers.

At the conclusion of the 2014 final round, we found that our Decision tree performed the best in terms of scoring earning 61 points, while our Logistic Regression and SVM were tied at 54.

TABLE XIV
2014 Correctly Predicted Wins by Round

Model	Rd64	Rd32	S16	E8	F4	Final
Logistic	24	10	2	1	0	0
Decision Tree	25	10	4	1	0	0
SVM	24	10	2	1	0	0

TABLE XV
2014 Score by Round

Model	Rd64	Rd32	S16	E8	F4	Final
Logistic	24	20	6	4	0	0
Decision Tree	25	20	12	4	0	0
SVM	24	20	6	4	0	0

Table XIV and XV illustrate the number of correct predictions and points for a model in each round. Fig. 19 rolls up the confusion matrices of each round to provide a single source of results for each model for the 2014 season.

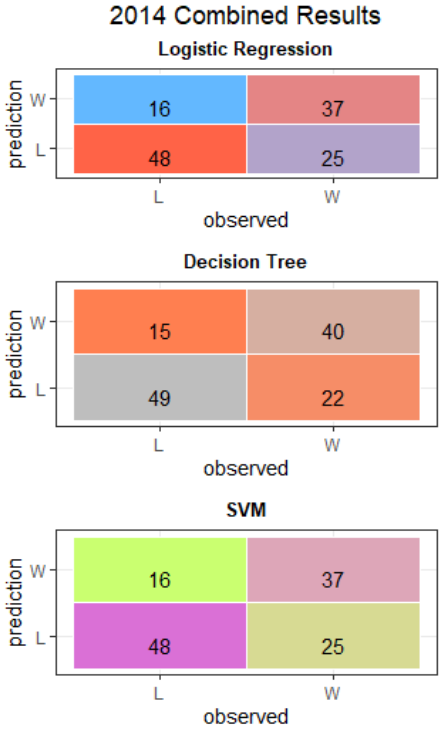

Fig. 19. Confusion Matrix of 2014 predictions for the Logistic Regression, Decision Tree and SVM classifiers.

C. Ties

As mentioned in the Testing section, ties are possible; therefore, we implemented a protocol to resolve collisions in predictions. While ties did occur, they were minimal and resolved using the classification probabilities produced by the model.

XIII. Model Comparison and Discussion

If we were to base our results on points score, the 2010 season would see the Logistic Regression and the SVM model tied (table VIII). But the 2014 season would suggest that the decision tree is the Superior model (table XV). What we see here is that there is little separation between the results of the first two rounds of the tournament. From a point perspective, these differences can be made up in later rounds, which is exactly what we see in 2010 with the logistic regression and SVM

model. In the first round, the logistic regression model correctly predicted three more teams than the SVM model. This point gap was closed when the SVM correctly predicted five of the eight winners in the Sweet 16. After that, each model correctly predicted the same teams, making them inseparable.

Even though the decision tree was unable to compete with the other two models, there is a level of optimism with regards to its predicting power. This is from the fact that, while it did not predict the correct national champion, it was able to predict the national champion would make the final round. While the other two models were able to accomplish this as well, the decision tree under performed against the other models in both the sweet 16 and elite 8.

In contrast, the 2014 data set suggested that the decision tree was the superior model. With little separation (Tables XIV and XV) in the Round of 64 and Round of 32 between the three models, the decision tree was the only model that held its performance level in the sweet 16 over the two years. After the 2014 Sweet 16, all models began to decline in performance in 2010. At first, this was alarming. However, the 2014 tournament was more of an anomaly [23] as the national champion was a 7th seed. This was a year that saw two teams seeded 7 or lower make the Final 4, and aside from the 7th seed, we also had an 8th seed. To add to the anomaly, this 8th seed also made the national finals. From the NCAA [24], the NCAA tournament has taken place, at the time of the article, 38 times. Only in three of those years has the winner not been seeded 1, 2, 3, or 4. To further add to this, 7 and 8 seeds have only each made the national championship a combined 5 times out. Having anomalous testing data makes it difficult to compare the models past the Round of 32. The decision tree is able to correctly predict half of the teams that made it to the Elite 8 in 2014.

We can see that model performance varied based on testing years. Therefore, we aggregated each round, combining the two years, as more correct predictions in later rounds hold more value than earlier.

TABLE XVI
Aggregated Correctly Predicted Wins by Round

Model	Rd64	Rd32	S16	E8	F4	Final
Logistic	48	20	6	3	1	0
Decision Tree	47	18	8	2	1	0
SVM	45	20	7	3	1	0

TABLE XVII
Aggregated Score by Round

Model	Rd64	Rd32	S16	E8	F4	Final
Logistic	48	40	18	12	5	0
Decision Tree	47	36	24	8	5	0
SVM	45	40	18	12	5	0

Tables XVI and XVII show a slight advantage in the first round for the logistic regression model. Similar to 2010, the Round of 32 shows a tie between the logistic regression and SVM model. The Sweet 16 shows the largest separation in terms of points, where the Decision Tree outperforms the other two models. However, in terms of correct predictions, the SVM outperform the logistic regression model. After the Sweet 16, we relied on the 2010 model for analysis. That said, we found that each model outperforms each other based on a given round.

Straying away from a points perspective, and focusing on the combination of both years, Fig. 20 illustrates that the logistic regression model correctly predicts more winners, but only slightly compared to the Decision Tree and SVM which have 76.

Fig. 20 shows that these three models have approximately the same predicting power in terms of predicting the correct winner. Predicting correct winners and winning a bracket pool heavily relies one which round these winners are predicted. Circling back to Fig. 12 and Fig. 19, we see that in 2010 our logistic regression model correctly predicted 41 of the 63 match-ups (65%), the decision tree predicting 57% and the SVM predicting 61% of the match-ups. Meanwhile, in 2014, the decision tree performed the best, correctly predicting 63%. However, the logistic regression

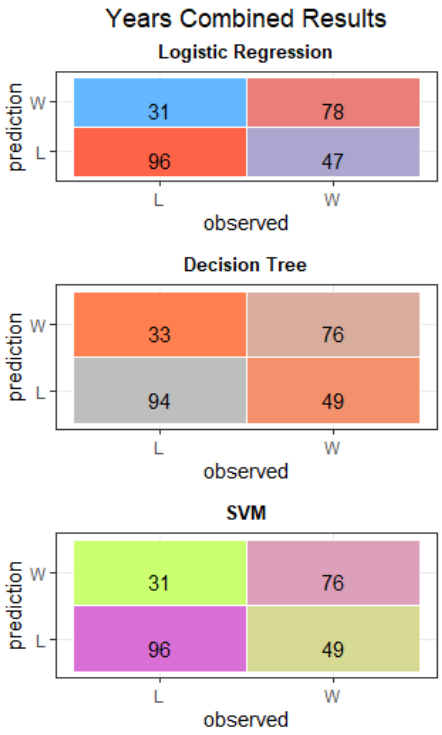

Fig. 20. Confusion Matrix of 2014 predictions for the Logistic Regression, Decision Tree and SVM classifiers.

and SVM model fell in predicting power to 58%.

A. Benchmarking

For our objective, a model's accuracy and predicting power have limited explain-ability of how a model may perform in a bracket pool. To provide more insight, we have evaluated our control brackets against, as previously mentioned, one where the higher seed always wins, and the brackets created by former President Barack Obama.

TABLE XVIII
Control Brackets Wins by Round

Ctrl	Rd64	Rd32	S16	E8	F4	Final
Seed '10	22	8	4	1	0	0
Obama '10	26	9	4	0	0	0
Seed '14	24	10	4	1	0	0
Obama '14	23	9	4	1	0	0

Table XIX shows the brakedown by round for our 2 control brackets. Similar to our models, we see that there are contrasting results between years. In 2010, Obama outperformed the Seed based selection method by 2 points,

TABLE XIX
Control Bracket Score by Round

Ctrl	Rd64	Rd32	S16	E8	F4	Final
Seed '10	22	16	12	4	0	0
Obama '10	26	18	12	0	0	0
Seed '14	24	20	12	4	0	0
Obama '14	23	18	12	4	0	0

scoring a total of 56 points in comparison to the 54 points from the seed based bracket. And in 2014, we see the seed based predictions scored a total of 60 points to the 57 points of Obama's bracket.

Furthering the contrast between years, Fig. 21 and Fig. 22 illustrate how well Obama and the seed based brackets were able to predict in their given years.

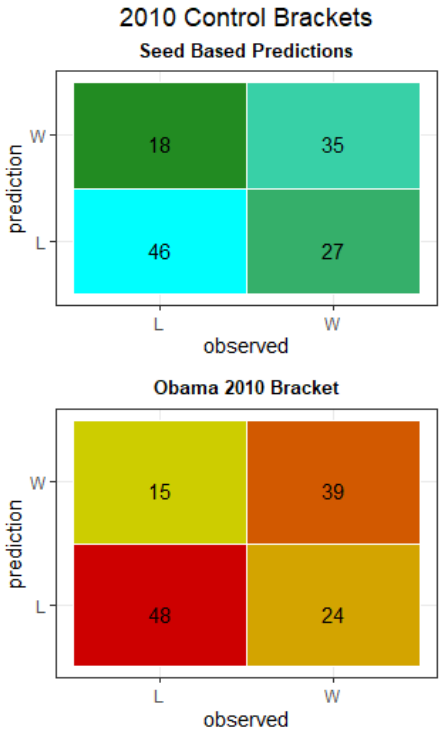

Fig. 21. Confusion Matrix of 2010 Control Brackets

While drilling down by year shows how our control brackets can perform for a given year, rolling the data up shows a different story. In Fig. 23, we see the Obama's bracket only outperformed the seed based brackets by 2 points.

In fact, any metric derived from the confusion matrices would further confirm the similarities. That is, none of the 4 quadrants vary

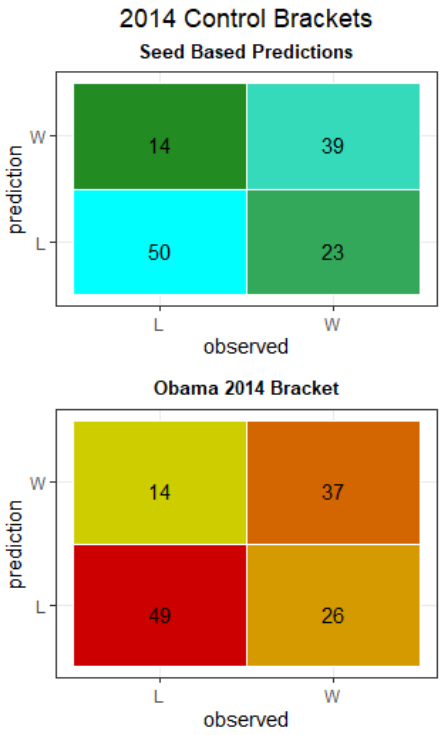

Fig. 22. Confusion Matrix of 2014 Control Brackets

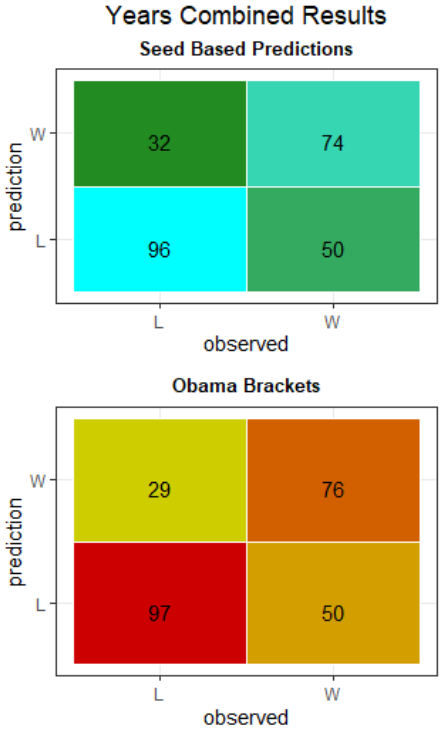

Fig. 23. Confusion Matrix of Combined Years Control Brackets

by more than 3.

B. Models Compared Benchmarks

We have previously shown how well our models have performed against each other. Now we will compare them to our control brackets.

TABLE XX
Control and Model Brackets Aggregated Wins by Round

Ctrl	Rd64	Rd32	S16	E8	F4	Final
Seed	46	17	8	2	0	0
Obama	49	18	8	1	0	0
Logistic	48	20	6	3	1	0
Decision Tree	47	18	8	2	1	0
SVM	45	20	7	3	1	0

By evaluating the correct predictions round by round, it is clear that the models have an advantage over the control groups in the later rounds of the tournaments (Table XX). This stems from the fact that in 2010, all three models were able to successfully predict one of the two national finalists, while our control brackets were not. Adding to this, the logistic regression and SVM models were more successful in predicting Elite 8 winners, as both predicted 3 winners in the two years compared to the 2 and 1 correct predictions for the seed based and Obama bracket.

Our control group, more specifically Obama, was the most successful in predicting the Round of 64. While we have seen from a points perspective that a lack of predicting power in the round of 64 can be overcome by better predicting power in the later rounds, it raises the question as to why the models did not perform better in the round of 64? To answer this, we looked at the data and found that Obama is surprisingly good at predicting upsets[6]. For example, In 2010, Obama correctly predicted 12th seeded Cornell to beat a 5th seeded Wisconsin. After reviewing our predictions, none of our models were able to correctly pick this outcome.

XIV. Conclusions

When it comes to comparing our models to our control groups, in 2010, both the logistic regression and the SVM models scored a total

[6]A lower seed beating a higher seed

of 69 points. This was largely assisted by their abilities to correctly predict multiple winners in the Elite 8 and one in the Final 4. The control brackets were unable to predict winners in these rounds. Thus, the best score produced from our control for 2010 was from Obama's bracket, which produced a score of 56. In fact, for 2010, the decision tree was also able to outperform our control group.

As for 2014, our control group outperformed their 2010 results. While the logistic regression and SVM models performed worse, totaling 54 points. On the contrary, we saw our decision tree slightly improve, producing 4 more correct predictions, and increasing its points from 59 to 61. In comparison, Obama increased the number of correct predictions from 35 to 39, while our seed-based bracket decreased from 39 to 37. Here again, Obama produced the best score from our control group with a total of 57. And while besting our prior two best models, it was unable to outperform our decision tree.

From the results of our testing, it can be difficult to determine definitively which model best serves our original question. Though our Logistic Regression and SVM models produced the most points in 2010, they were unable to reproduce similar results for 2014. Conversely, our decision tree lacked in prediction power in 2010, but was the best model in 2014. Though when we combine our testing set to get a generalization of our models (Fig. 20), our three models would perform relatively the same over the long run. For the three models, the number of correct predictions vary by 2, and ranging in accuracy from 67.46% to 69.04%. Although our models were able to produce good accuracy, to be successful in a bracket pool they should be judged on their hit rate. A higher number of correctly predicted wins will results in a higher scoring. The logistic regression model was best in this category with a 61.9% hit rate, with the SVM and decision tree having a hit rate of 60.3%

Correct predictions have thus far been the primary focus; however, we must also evaluate our misclassified observations. Not only are the number of correct predictions similar, our misclassified observations are as well. Fig. 20 shows that our models have a high rate of misclassification. The logistic regression model has the lowest number of misclassifications, incorrectly classifying 30.9% observations. Meanwhile, the SVM and Decision Tree have 31.7% and 32.5% respectively.

From a generalization stance, the logistic model presents the best traits of being a successful model. Yet so does the Decision Tree and the SVM. We speculate that this stems from the data set. The data chosen was based off of Lopez and Matthews where efficiency metrics were used to assign probabilities for the Round of 64 match-ups. Their results were successful, and won a Kaggle competition based on their methodologies. However, with the data set our SVM only tested at 67.46% falling well short of the accuracy presented by Shen, et al., which performed similar experiment as ours. The difference between our experiment extends past the quantity of data, as their data set stems from 2008 to 2014, including the variables which extend past the efficiency metrics to per game averages, for example Free Throw Attempts Per Game. These variables may be the cause of the drastic change in accuracy.

An additional factor that may have caused the stark difference in accuracy is the testing set. Shen et al. used, at the time, the two most recent tournaments, while we randomly selected the years of 2010 and 2014. As previously mentioned, 2014 was an anomalous year, and 2010 saw a 5 seed in the national championship, which has only happened 4 times in 84 tournaments, or 2.5% of the 162 teams to play in the finals. In comparison, the years 2015 and 2016 both saw two 1 seeds play in the national championship.

While the logistic regression model holds the best predictive powers of the three tested, the SVM and Decision Tree are close in comparison with the given data set, being within 2% in both accuracy and hit rate. And based on experiments tested by others, expanding upon the variables may allow better predicting pow-

ers. However, like these other experiments, we may not have a single model that consistently outperforms others.

XV. Next Steps

While our testing protocols could be refined, the next immediate step would be to collect more data. The conclusion is that our results were less than desirable. As mentioned in our closing, we have identified other research papers that uses a more verbose data set. To accomplish this, we would need to develop a web scrapper to expand upon the variables used. This would be a necessary step as the data is not readily available without purchase. We would then repeat our analysis using the expanded data before adjusting other facets of our project.

XVI. Learning Points

In this project we were able to gain hands-on experience deploying concepts learned throughout the course load. After collecting and cleaning our data, we were able to utilize concepts from Exploratory Data analysis. By visualizing the distributions, we recognized a slight skewness in the data. To resolve this, we applied transformations which normalized the data. We then took the data and applied methods learned in through our Data Mining Seminar such has creating a Decision Tree to generate a set of rules and Support Vector Machines to create hyper-planes to separate and classify our observations. From our Statistical Learning course, we applied Logistic Regression as well as recursive feature elimination to simplify our models.

To apply these concepts it required self-learning of libraries found in familiar programming languages. To clean the data, we used Python's Pandas library. Here, we were able to import the data from csv and xlsx formats into a data frame. In turn, we were able to manipulate these data types to extract the necessary data for our analysis. Once cleaned, we then explored the data in R, using qqnorm to create normal QQ plots. We also were able to remove variables that displayed co-linearity using the Variance Inflation Factor function from the car library. Prior to training our models, we utilized the caret library and its recursive feature elimination function for our decision tree and SVM models. This allowed us to eliminate variables using cross-validation. For feature selection on our Logistic Regression model, we utilized Lasso regression to minimize our coefficients.

To train, each model required a different library. For the logistic regression model, we used the glmnet. With the coefficients derived from the Lasso Regression, we performed a 5-fold cross validation to train. The Decision Tree was trained using the rpart library, allowing us to tune our hyper-parameters to maximize our training accuracy. Our SVM was trained and the hyper-parameters were tuned using the e1071 library.

Testing was accomplished using R's predict function. While we originally produced confusion matrices using the caret library, we sought the help of the ggplot library. This allowed for better readability of our plots.

XVII. Lessons Learned

Many lessons have been learned throughout this project. First, that this is a time consuming process and to expect malfunctions. We thankfully experienced this early on in our process where our computer unexpectedly crashed, erasing our R code. Thankfully, this occurred only in the Exploratory Data Analysis phase. After that, we created a repository on GitHub to store our files.

Training was also a troublesome point in our process. While we had the resources necessary to complete this analysis, we found that the libraries used did not utilize our complete computing powers. Namely, finding documentation on how to access the computing power our GPU could provide. To resolve this issue in the future, we would move our analysis away from R and fully into Python. During our search for documentation, we found that Python has libraries that contain the algorithms used in our analysis. Furthermore, Python has the ability to access our GPU.

Automation was a sticking point in our code. Since this was the first research project we have performed, our code is more script based rather than modular. Creating objects, methods and functions would have been useful to create a cleaner understanding of our code base.

XVIII. Thank you

Here, I would like to take a moment to say thank you to the faculty and staff at Indiana University South Bend. I have thoroughly enjoyed my time learning from Professionals dedicated to their craft and willingness to lift students to their full learning potential.

I would also like to give a special thanks to

- Dr. Dana Vrajitoru for her tremendous help, guidance and time as an advisor for this project.
- Dr. Liqiang Zhang for his role as my advisor in my Graduate studies.
- Dr. Peter Connor for his role as my advisor through the majority of my Graduate studies.

References

[1] NCAA.com, "Tracking all 32 NCAA Men's basketball conference tournaments, auto bids for 2023," NCAA.com, https://www.ncaa.com/news/basketball-men/article/2023-03-12/2023-ncaa-conference-tournaments-schedules-brackets-scores-auto-bids (accessed Jan. 18, 2024).

[2] D. W. — NCAA.com, "What is an at-large bid in March madness?," NCAA.com, https://www.ncaa.com/news/basketball-men/article/2019-02-05/what-large-bid-march-madness (accessed Jan. 18, 2024).

[3] D. W. — NCAA.com, "The first four of the NCAA Tournament, explained," NCAA.com, https://www.ncaa.com/news/basketball-men/bracketiq/2022-03-15/first-four-ncaa-tournament-ultimate-guide (accessed Jan. 18, 2024).

[4] M. Koba, "Your march madness pool is probably illegal," CNBC, https://www.cnbc.com/2014/03/17/march-madness-pools-breaking-the-law-ncaa-bets-are-often-illegal.html (accessed Jan. 18, 2024).

[5] P. Kvam and J. S. Sokol, "A logistic regression/markov chain model for NCAA basketball," Naval Research Logistics (NRL), vol. 53, no. 8, pp. 788–803, 2006. doi:10.1002/nav.20170

[6] R. Beal, T. J. Norman, and S. D. Ramchurn, "A critical comparison of machine learning classifiers to predict match outcomes in the NFL," International Journal of Computer Science in Sport, vol. 19, no. 2, pp. 36–50, 2020. doi:10.2478/ijcss-2020-0009

[7] N. C. Schwertman, K. L. Schenk, and B. C. Holbrook, "More probability models for the NCAA Regional Basketball Tournaments," The American Statistician, vol. 50, no. 1, p. 34, Feb. 1996. doi:10.2307/2685041

[8] N. E. O. and A. O. Uzoma, "A hybrid prediction system for American NFL results," International Journal of Computer Applications Technology and Research, vol. 4, no. 1, pp. 42–47, Jan. 2015. doi:10.7753/ijcatr0401.1008

[9] M. J. Lopez and G. J. Matthews, "Building an NCAA men's basketball predictive model and quantifying its success," Journal of Quantitative Analysis in Sports, vol. 11, no. 1, 2015. doi:10.1515/jqas-2014-0058

[10] J. Kubatko, D. Oliver, K. Pelton, and D. T. Rosenbaum, "A starting point for analyzing basketball statistics," Journal of Quantitative Analysis in Sports, vol. 3, no. 3, 2007. doi:10.2202/1559-0410.1070

[11] 2024 Pomeroy College Basketball Ratings, https://kenpom.com/ (accessed Jan. 20, 2024).

[12] historical NCAA Basketball Scores and Odds Archives," SportsBookReviewOnline, https://www.sportsbookreviewsonline.com/scoresoddsarchives/ncaabasketball/ncaabasketballoddsarchives.htm (accessed Feb. 22, 2024).

[13] "College Basketball Home Court Advantage Study," VSiN, [Online]. Available: https://vsin.com/featured/college-basketball-home-court-advantage-study/#: :text=In%20general%2C%20I%20believe%20most,about%202.8%20in%20conference%20play. [Accessed: Mar 01, 2024].

[14] "2010 NCAA tournament: Bracket, scores, stats, records," NCAA.com, [Online]. Available: https://www.ncaa.com/news/basketball-men/article/2020-05-12/2010-ncaa-tournament-bracket-scores-stats-records [Accessed: Mar 06, 2024].

[15] "2014 NCAA tournament: Bracket, scores, stats, records," NCAA.com, [Online]. Available: https://www.ncaa.com/news/basketball-men/article/2020-05-10/2014-ncaa-tournament-bracket-scores-stats-records [Accessed: Mar 06, 2024].

[16] "President Picks His Favorites in the 2010 NCAA Basketball Tournament," The White House Archives, [Online]. Available: https://obamawhitehouse.archives.gov/blog/2010/03/17/president-picks-his-favorites-2010-ncaa-basketball-tournament [Accessed: Mar 08, 2024].

[17] "President Obama's Bracket: 2014 NCAA Men's Basketball Tournament," The White House Archives, [Online]. Available: https://obamawhitehouse.archives.gov/blog/2014/03/19/president-obamas-bracket-2014-ncaa-mens-basketball-tournament [Accessed: Mar 08, 2024].

[18] "Bracket Busted: March Madness," NPR, [Online]. Available: https://www.npr.org/2024/03/23/1240296429/bracket-busted-march-madness [Accessed: Mar 29, 2024].

[19] "March machine learning mania," Kaggle, https://www.kaggle.com/competitions/march-machine-learning-mania-2014/overview (accessed Jan. 19, 2024).

[20] D. Delen, D. Cogdell, and N. Kasap, "A comparative analysis of data mining methods in predicting NCAA bowl outcomes," International Journal of Forecasting, vol. 28, no. 2, pp. 543–552, 2012. doi:https://doi.org/10.1016/j.ijforecast.2011.05.002

[21] G. Shen, D. Gao, Q. Wen, and R. Magel, "Predicting results of March madness using three different methods," Journal of Sports Research, vol. 3, no. 1, pp. 10–17, 2016. doi:10.18488/journal.90/2016.3.1/90.1.10.17

[22] J. Brownlee, "How to Configure k-Fold Cross Validation," Machine Learning Mastery, 2019. [Online]. Available: https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/. [Accessed: April 23, 2024].

[23] "What Is The Lowest Seed To Win NCAA March Madness?," Forbes, [Online]. Available: https://www.forbes.com/betting/basketball/college-basketball/what-is-the-lowest-seed-to-win-ncaa-march-madness/ [Accessed: Apr 01, 2024].

[24] "March Madness brackets: How do seeds perform in the Final Four?," NCAA.com, [Online]. Available: https://www.ncaa.com/news/basketball-men/bracketiq/2017-12-14/march-madness-brackets-how-do-seeds-perform-final-four [Accessed: Apr 02, 2024].

[25] D. Oliver, Basketball on Paper: Rules and Tools for Performance Analysis. Washington, D.C.: Potomac Books, Inc., 2011.

[26] "NCAA BB Team Posessions Per Game," NCAA Basketball Stats - NCAA BB Team Points per Game — TeamRankings.com, https://www.teamrankings.com/ncaa-basketball/stat/possessions-per-game?date=2023-04-04 (accessed Jan. 23, 2024).

[27] "NCAA BB Team Points Per Game," NCAA Basketball Stats - NCAA BB Team Points per Game — TeamRankings.com, https://www.teamrankings.com/ncaa-basketball/stat/points-per-game?date=2023-04-04 (accessed Jan. 23, 2024).

[28] M. Ellentuck, "What is the first four? and why is it a thing?," SBNation.com, https://www.sbnation.com/college-basketball/2018/3/13/17101418/ncaa-tournament-first-four-times-teams-history-march-madness (accessed Jan. 18, 2024).