Predictive Models for March Madness Bracket

Tyler Kurpanek
tyler.kurpanek@gmail.com
University of California, San Diego
San Diego, California, USA

Abstract

In this paper, I evaluate the effectiveness of four different regression models in predicting the outcomes of past NCAA March Madness tournaments. By analyzing historical tournament data, I assess the predictive accuracy of each model and compare their strengths and weaknesses. Finally, I apply these models to forecast the March Madness 2025 bracket, providing insight into potential tournament outcomes.

1 Introduction

March Madness, the annual NCAA Division I Men's Basketball Tournament, is one of the most unpredictable and widely followed sporting events in the United States. Every year, 68 college basketball teams compete in a single-elimination format, leading to a national champion. Fans, analysts, and sports bettors attempt to predict the tournament's results, but the chaotic nature of the event, characterized by frequent upsets and Cinderella stories, makesit nearly impossible to construct a perfect bracket.

Traditionally, March Madness predictions have relied on expert analysis, historical performance, and basic statistical measures such as win-loss records and team rankings. However, as data science and machine learning continue to evolve, predictive modeling has emerged as a powerful tool for forecasting sports outcomes. Using historical tournament data and advanced statistical techniques, machine learning models can analyze patterns, identify key predictive factors, and generate data-driven predictions.

In this study, I evaluated four different regression-based models: linear regression, random forest regression, elastic net regression, and XGBoost regression to determine their effectiveness in predicting the results of the March Madness game. These models represent a mix of traditional statistical approaches and more complex machine learning techniques, allowing us to compare how different methods handle tournament prediction. I train each model on historical tournament data and assess their predictive accuracy by testing them in past seasons.

By analyzing the strengths and weaknesses of each model, I aim to determine which approach is best suited for predicting NCAA tournament outcomes. Finally, I apply these models to the March Madness 2025 tournament, using their predictions to forecast potential results. This research not only explores the effectiveness of different regression techniques in sports analytics but also demonstrates the increasing role of machine learning in enhancing predictive accuracy for large-scale sporting events.

2 Data

2.1 Gathering Data

I gather data from two different sources. I used historical NCAA team and tournament data from Kaggle [2] and 2024-25 team data from Teamrankings [3].

The Kaggle Dataset, stored as cbb.csv contains college basketball regular season and bracket stats from 2013 to 2024, totaling 3523 columns. Each row of the dataset corresponds to the stats of a given team during a given year. There were 24 rows in the original dataset, but after cleaning, I ended up using 18 variables as predictors and 1 target variable for our analysis.

To assemble the dataset that I are going to use as our predictors for this years tournament, I first needed the teams in the tournament and the records for this year. I used ESPN [1] and manually transferred the data over. For this years stats, I webscraped TeamRankings and covnerted each of the categories I wanted into its own dataframe. I then merged all of these dataframes onto the dataframe I created with the current bracket teams found from ESPN.

2.2 Variables

The variable I are trying to predict is:

POSTSEASON: How far the team advanced in the NCAA tournament

And the 18 Variables Iused to predict the March Madness Bracket are:

W: Wins

L: Losses

ADJOE: Adjusted offensive efficiency (points scored per 100 possessions, adjusted for opponent strength)

ADJDE: Adjusted defensive efficiency (points allowed per 100 possessions, adjusted for opponent strength)

EFG%: Effective field goal percentage, accounts for three-pointers being worth more than two-pointers

TOR: Turnover percentage, the percentage of offensive possessions that result in a turnover

ORB: Offensive rebounding percentage, the percentage of available offensive rebounds a team grabs

FTR: Free throw rate, the ratio of free throws attempted to field goals attempted

2P%: Two-point shooting percentage, the percentage of two-point field goal attempts made

3P%: Three-point shooting percentage, the percentage of three-point field goal attempts made

2P_D: Opponents' two-point shooting percentage

3P_D: Opponents' three-point shooting percentage

EFG_D%: Opponents' effective field goal percentage

TOR_D: Opponents' turnover percentage, the percentage of defensive possessions that result in a turnover

DRB: Defensive rebounding percentage, the percentage of available defensive rebounds a team grabs

FTR_D: Opponents' free throw rate, the ratio of free throws attempted to field goals attempted

ADJ_T: Adjusted tempo, an estimate of possessions per 40 minutes adjusted for opponent strength

SEED: Seed in the NCAA tournament

2.3 Data Cleaning

After loading in the Kaggle data, I needed to first create a loss column since they did not have one. I then dropped the columns that I are not going to use in our analysis: Year, Conference, Barthag, Games, and Wab. All teams played around the same amount of games and I already have wins and losses so it was a redundant variable. The Barthag and Wab are both rankings based off certain stats and would've been nice to include, however, were not available for this season on TeamRankings so I did not use them. The year would not have made any impact because I are trying to predict for 2025 for all teams. Laslty, the conference variable was removed because teams have chaned conferences lots of times in the years from our data set so it would introduce bias to use conference data from past years when the conferences for this year are different.

Next, I mapped the Postseason column to be numeric instead of categorical so I can use our regression models. I assigned a point value to how far each team makes it in the tournament: "R68": 0, "R64": 0, "R32": 1, "S16": 2, "E8": 4, "F4": 8, "2ND": 16, "Champions": 32 Lastly, I split our data into a train and test split in order to see the performance metrics from past year.

3 Models

3.1 Linear Regression

Linear regression is a linear approach to modeling the relationship between a dependent variable y and one or more independent variables x. It is based on the assumption that the relationship between the variables can be described by a linear equation. The general form of a simple linear regression model is given by:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where: - y_i is the dependent variable, - x_i is the independent variable, - β_0 is the intercept or constant term, - β_1 is the slope coefficient, and - ϵ_i is the error term.

In matrix form, this can be represented as:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

where: - $\mathbf{Y}$ is the vector of dependent variables, - $\mathbf{X}$ is the design matrix including a column of ones for the intercept, - β is the vector of coefficients, and - ϵ is the vector of error terms.

The coefficients β are typically estimated using Ordinary Least Squares (OLS), which minimizes the sum of the squared errors:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

3.2 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of predictions. It works by training each decision tree on a random subset of the data and features, and then aggregating their predictions. The Random Forest model can be described as a collection of randomized base regression trees $\{r_n(x, \Theta_m, D_n), m \geq 1\}$, where Θ_m are i.i.d. outputs of a randomizing variable Θ, and D_n is the training dataset.

The aggregated regression estimate $\bar{r}_n(x, D_n)$ is given by:

$$\bar{r}_n(x, D_n) = E_{\Theta}[r_n(x, \Theta, D_n)]$$

where E_{Θ} denotes expectation with respect to the random parameter Θ.

Random Forests are particularly effective because they reduce overfitting by averaging the predictions of multiple trees, each trained on a different subset of the data.

3.3 Elastic Net

Elastic Net is a regularization technique that combines both L1 and L2 regularization. It is used in linear regression to prevent overfitting by adding a penalty term to the loss function. The Elastic Net cost function is given by:

$$E(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2 + \alpha \left(\lambda \sum_{j=1}^{p} |\beta_j| + \frac{1 - \lambda}{2} \sum_{j=1}^{p} \beta_j^2 \right)$$

where: - α is the regularization strength, - λ is the mixing parameter between L1 and L2 regularization, with $\lambda = 1$ corresponding to Lasso (L1) and $\lambda = 0$ corresponding to Ridge (L2) regression.

The Elastic Net model is particularly useful when there are many correlated features, as it can set some coefficients to zero (similar to Lasso) while also shrinking others (similar to Ridge).

3.4 XGBoost

XGBoost is an ensemble additive model composed of several base learners, typically decision trees. It is designed to be highly efficient and scalable. The core idea behind XGBoost is to iteratively add models that correct the errors of previous models. At each iteration, XGBoost fits a base learner to the negative gradient of the loss function, similar to gradient boosting.

The loss function is approximated using a Taylor series expansion up to the second order:

$$L(y, \hat{y}) \approx L(y, \hat{y}^{(t-1)}) + g_i \cdot f_t(x_i) + \frac{1}{2} h_i \cdot f_t^2(x_i)$$

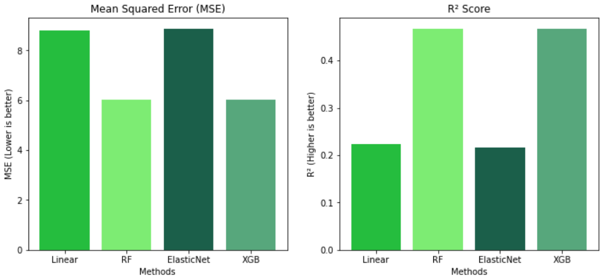
where: - g_i is the first-order derivative of the loss function with respect to the predictions at the previous iteration, - h_i is the second-order derivative of the loss function with respect to the predictions at the previous iteration.

XGBoost uses this approximation to efficiently explore different base learners and select the one that minimizes the loss. It also uses a greedy algorithm to construct decision trees, choosing splits that result in the maximum reduction in loss.

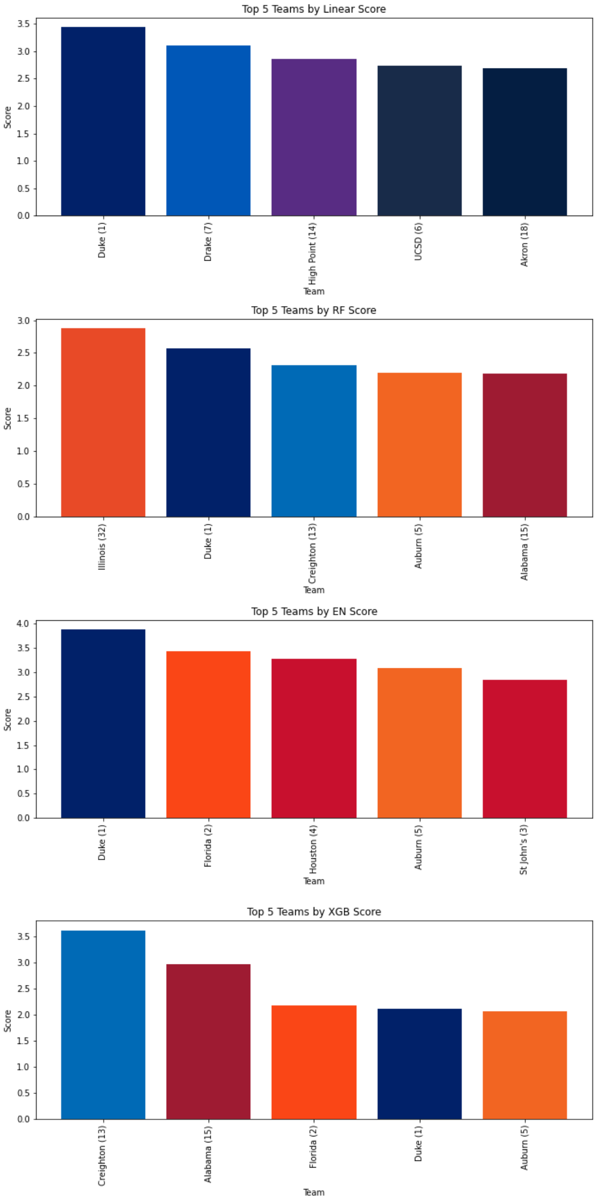
4 Results

The results of our models are presented as follows. For figures 2, 3, and 4, the numbers in parentheses represent the average rank combined across all models. This comprehensive approach allows for a thorough analysis of model performance.

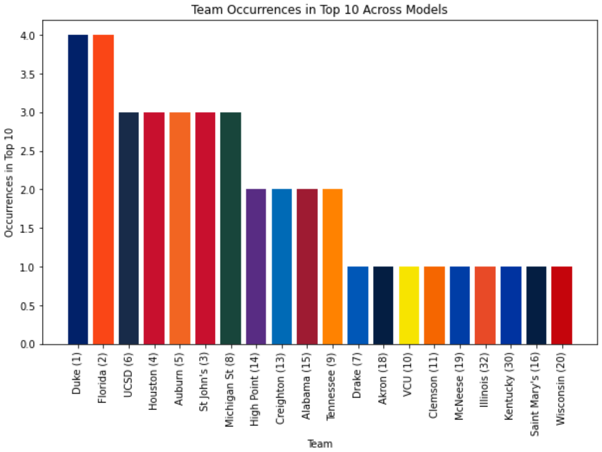
[Figure 1: Model Performance]



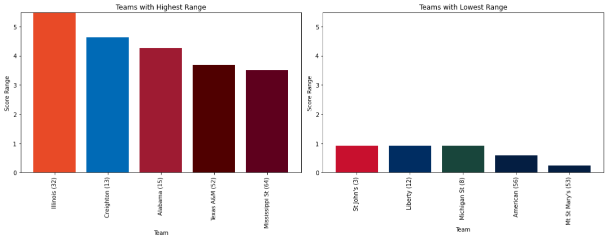
[Figure 2: Top 5 Teams for each Model]



[Figure 3: Most Common Teams in Top 10 For Each Model]



[Figure 4: Teams With the Largest and Smallest Range]



5 Discussion

5.1 Mean Squared Error

From Figure 1, I can see that the Mean Squared Error values for different models are as follows:

- **XGBoost**: 6.03
- **Random Forest**: 6.04
- **ElasticNet**: 8.88
- **Linear Regression**: 8.80

From these values, I observe that XGBoost and Random Forest have the lowest errors, suggesting they provide the most accurate predictions. Conversely, ElasticNet and Linear Regression exhibit higher errors, indicating potential difficulties in capturing the complexity of the data. This suggests that tree-based models (Random Forest and XGBoost) outperform linear models in this setting.

5.2 R-Squared

From Figure 1, I see the R-Squared values for the models are:

- **XGBoost**: 0.467
- **Random Forest**: 0.467
- **ElasticNet**: 0.216
- **Linear Regression**: 0.223

XGBoost and Random Forest have the highest R-squared values, indicating that they explain more of the variance in the data compared to the other models. This suggests that they provide better fits to the data, though they may still be sensitive to specific data patterns. On the other hand, ElasticNet and Linear Regression have lower R-squared values, meaning they explain less of the variance, which aligns with their higher error values and potential underfitting.

This observation aligns with expectations: linear models generally generalize better but may not effectively capture complex relationships, whereas ensemble tree-based models are more flexible and tend to achieve better goodness-of-fit.

5.3 Top Teams by Model Scores

From figure 2, I see the rankings of the top five teams across different models reveal notable differences in how each model evaluates team performance.

5.3.1 Linear Model (Linear Regression).

- **Duke (Rank 1)** has the highest score (3.44), showing that the linear model strongly correlates with their performance.
- **Drake (Rank 7)** and **High Point (Rank 14)** are surprisingly high, suggesting that this model may favor certain statistical patterns that benefit these teams.
- **UCSD (Rank 6)** and **Akron (Rank 18)** making the top five indicates that the linear model may not fully align with the overall rankings, potentially overemphasizing certain features.

5.3.2 Random Forest (RF).

- **Illinois (Rank 32)** is the highest-scoring team, which is unexpected given its overall ranking. This suggests that RF prioritizes different features than other models.
- **Duke (Rank 1)** remains strong, which supports its overall performance consistency.
- **Creighton (Rank 13)** and **Auburn (Rank 5)** also perform well, showing that RF still aligns with general rankings but introduces variance in predictions.

5.3.3 Elastic Net (EN).

- **Duke (Rank 1)** is again the top team, reinforcing its consistent performance across models.
- **Florida (Rank 2)** and **Houston (Rank 4)** appear near the top, indicating EN's ability to balance different variables effectively.
- **Auburn (Rank 5)** and **St. John's (Rank 3)** rounding out the top five suggests that EN favors teams with strong balanced performances.

5.3.4 XGBoost (XGB).

- **Alabama (Rank 15)** takes the top spot, suggesting that XGBoost captures unique features that other models may not emphasize.
- **Florida (Rank 2)** and **Duke (Rank 1)** remain strong, showing that XGBoost aligns well with general rankings but introduces some variation.
- **Auburn (Rank 5)** is once again in the top five, making it one of the more consistently high-performing teams across models.

5.3.5 Takeaways.

- **Duke is the most consistent performer**, ranking in the top five across all models, which validates its strong overall performance.
- **Random Forest (RF) introduces the most variation**, placing Illinois (Rank 32) at the top, suggesting it captures different aspects of team performance.
- **XGBoost and Elastic Net highlight Alabama and Florida**, showing that they may consider factors that better reflect hidden strengths in these teams.
- **Teams like Auburn and Florida appear frequently in the top five**, indicating they are well-rounded performers under different model criteria.

5.4 Analysis of Team Ranges

From Figure 4, I see significant variability in the rankings of certain teams and also some teams that exhibit very small changes in rankings across the different models

5.4.1 Highest Range Teams. Among the teams with the highest range, `Illinois` stands out with a range of 5.48, reflecting substantial inconsistency across different models. The range measures the difference between the highest and lowest model predictions, and Illinois' wide range suggests that different models produce notably differing outcomes for this team. Other teams, such as `Creighton` and `Alabama`, show similarly large ranges, indicating that their rankings are also influenced by model selection. The presence of teams with such high ranges points to the importance of considering multiple models when evaluating a team's performance, as the predictions can be sensitive to the choice of algorithm.

5.4.2 Lowest Range Teams. On the other end of the spectrum, the teams with the lowest range, such as `Mt St Mary's` and `American`, exhibit far smaller variability. With ranges close to 0.25 and 0.58, these teams show much more consistency across the models, indicating that their performance predictions are less susceptible to the nuances of different algorithms. This could suggest that the models are more aligned in their assessments for these teams, making them more predictable or less affected by the particularities of each model.

5.4.3 Implications of Range. The range serves as an important metric for assessing the reliability of model predictions. Teams with higher ranges are more unpredictable, meaning their ranking can vary significantly depending on the model used. Conversely, teams with lower ranges tend to have more stable and reliable predictions, suggesting consensus across different models.

Conclusion

This study offers a comprehensive approach to predicting the outcomes of the March Madness tournament by employing multiple machine learning models, including Random Forest and XGBoost. One of the key findings of this analysis is the identification of `Duke` as the obvious favorite. The model predictions align with the broader consensus from March Madness experts, further validating the strength of `Duke` as a top contender. This consistency between the study's results and expert opinions provides confidence in the predictive power of the chosen models, especially for teams with well-established reputations.

However, the true value of this study lies in its potential to uncover "Cinderella" teams—underdog teams that may outperform expectations. Through the complexities of the models, particularly with Random Forest and XGBoost, this study highlights the subtle differences in team dynamics that may not be immediately apparent. These models account for interactions and non-linear relationships between features, offering a more nuanced prediction that could reveal teams capable of making unexpected runs in the tournament. While these "Cinderella" teams are often difficult to identify through traditional methods, the model's ability to capture complex patterns could offer a unique edge in predicting potential upsets.

To generate the final bracket, I combined the results from the Random Forest and XGBoost models, as these two models exhibited the lowest errors and, therefore, the highest accuracy in the predictions. By merging the insights from these models, I aimed to create a more robust and reliable bracket that reflects the most consistent outcomes based on the study's results.

In conclusion, this study not only reinforces the predictions of established favorites like Duke, but also provides a framework for identifying potential surprises in the tournament. The use of advanced machine learning models allows for a deeper understanding of the factors influencing team performance, and this approach could be invaluable for future tournament predictions.

References

[1] ESPN. 2025. Men's NCAA Tournament Bracket 2024-25. https://www.espn.com/mens-college-basketball/bracket Accessed: 16 March 2025.

[2] Andrew Sundberg. 2024. College Basketball Dataset. https://www.kaggle.com/datasets/andrewsundberg/college-basketball-dataset Accessed: 16 March 2025.

[3] TeamRankings. 2025. NCAA Basketball Team Stats. https://www.teamrankings.com/ncb/team-stats/ Accessed: 16 March 2025.