March Madness Estimations

Presented by: Anthony Ngumah, Ben Yoon, Enzo Grimaud

ABSTRACT

March Madness, an annual NCAA basketball tournament, attracts millions of fans as they attempt to predict one of the most challenging creation of brackets ever. The tournament's drama and unpredictability draw watchers to test their game knowledge in guessing who will win the tournament.

In this project, we leveraged pre-tournament statistics to build a machine-learning model to predict teams' strength as a numeric value, a power ranking (BARTHAG). With this data, we ultimately simulate a final matchup to determine the winner of the tournament.

The most accurate model produced the winner for 7 out of the 10 years we pulled data from. When running the model on 2024 data, it predicted the winner within the final 2 and the finalist within the top 3 seeds.

INTRODUCTION

Predicting March Madness results is a historically difficult task to accomplish. More than 22 million people this year submitted a bracket predicting the exact wins and losses in each round of the tournament. 9.3 million of them were invalidated in the first round due to Michigan State's win over Mississippi State. There exists an element of seemingly unpredictable randomness in the final seedings of teams as well as the team that ultimately wins March Madness. Our machine learning models aim to solve this problem by predicting the final seedings of teams by using pre-season data. We determined that if we are even able to predict the winner of the tournament as a finalist, then our model has accomplished our objective.

Obtaining Data: Gathering as many stats of teams as possible to assess a team's strength.

EDA: Preparing our data to be able to split it and train machine learning models. Dataset was very clean with no nulls or invalid numbers, removed useless columns.

Model Assessment: Exploring the application of 3 different ML models to discover the one that most accurately get our NCAA winner.

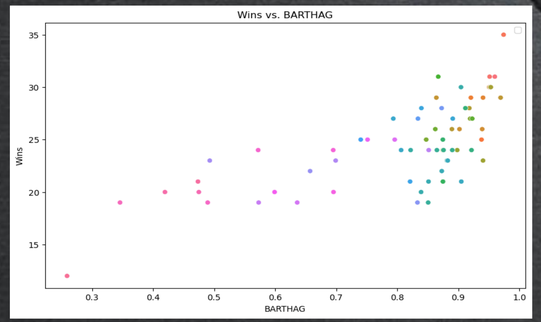
Model Training and Evaluation: Splitting to train and test data. Evaluation of models using Mean Squared Error (MSE) and tuning hyperparameters.

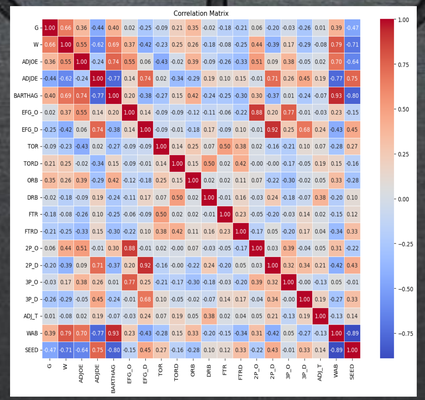
METHODOLOGY

To design our March Madness Estimator, we first had to acquire pre-existing data on historical March Madness tournaments. After scouring the internet for potential candidates, we settled on a particular data set from Kaggle at DATA.

This dataset included tournament data from 2013 up to 2023 providing us with approximately 3500 data points to work and build our model with. Our preliminary EDA steps involved taking action toward cleaning up our data and understanding correlations. First off, we decided to forgo any data belonging to the year 2020, as due to the COVID epidemic, the data for that year was unusable. When determining a winner for March Madness, there are a multitude of factors that may determine how likely a team is to win the tournament. Each season data frame contained the following columns: the Division I college basketball team (TEAM), The Athletic conference (CONF), number of games played (G), number of games won (W), adjusted offensive efficiency (ADJOE), adjusted defensive efficiency (ADJDE), power rating (BARTHAG), effective field goal percentage shot (EFG_O), effective field goal percentage allowed (EFG_D), turnover percentage allowed (TOR), turnover percentage committed (TORD), offensive rebound rate (ORB), offensive rebound rate allowed (DRB), free throw rate (FTR), free throw rate allowed (FTRD), 2 point shooting average (2P_O), 2 point shooting average allowed (2P_D), 3 point shooting percentage (3P_O), 3 point shooting percentage allowed (3P_D), adjusted tempo (ADJ_T), wins above bubble (WAB), round where the team was eliminated (POSTSEASON), seed in the March Madness tournament (SEED), and season (YEAR). After reading through the dataset, we decided that the column "BARTHAG" would be a good indicator of a team's likelihood to win the tournament. We created a correlation matrix to understand which other columns contribute to the "BARTHAG" metric.

Given that BARTHAG is a numeric value, we knew to utilize a linear regression model for our predictor. In order to build the most accurate model we played around with different coefficient thresholds but ultimately, we ended up using any column with a correlation >= 0.05 as a feature (G, W, ADJOE, EFG_O, ORB, 2P_O, WAB), and BARTHAG as our target. We then built our first linear regression model, a hyper tuned SVR model. For our next two models, we kept down the route of linear regression and decided to use a Random Forest and a Gradient Boosting Regression model as our final model. Both models offered a unique perspective predicting a team's strength score.





Comparing this to the effectiveness of our hyper tuned SVR model we observed a significant decrease in the mean squared error, proving that the Random Forest model was indeed more accurate. Additionally, Gradient Boosting is less sensitive to noisy data and outliers and given the fact that we are dealing with such a large sample and messy data, we expected this to have some benefits when making the model, and it turned out that the Gradient Boosted model boasted the lowest mean squared error. All three models were hyper tuned to provide the best possible outcome.

RESULTS AND EVALUATION

Regression Model	Best Hyperparameters	Best R^2 Score	Lowest Mean Squared Error
SVR	'C': 1 'epsilon': 0.01 'gamma':'scale' 'kernel':'rbf'	0.9032	0.00293
Random Forest	'max_depth': 10 'min_samples_leaf': 4 'min_samples_split': 10 'n_estimators': 150	0.9055	0.00286
Gradient Boost	'learning_rate': 0.05 'max_depth': 3 'n_estimators': 100	0.9138	0.00261

Upon training the three regression models shown in the figure above it was found that all three models exhibited high accuracy with the lowest correlation coefficient being 0.9032 and the highest mean squared error being 0.00293. Upon deeper inspection into the training set, we found that despite some teams' STR_SCORE being higher than their matchups, the model placed the opponent higher in seeding, which means that the models were even able to account for upsets, the most difficult variable to predict in a March Madness tournament.

Final comparisons between the predicted final seedings and actual final seedings showed that the Gradient Boost model outperformed the rest of the models, predicting 8 out of the 10 year's actual champions were in the top 2. Followed by SVR and RF predicting the champion as a finalist in 6 out of the 10 years.

IMPACTS

Without prior knowledge of team's and all of their stats, determining the winner of a head-to-head matchup can be seemingly impossible when in the NCAA Championship. By selecting stats that we have deemed valuable when assessing a team's strength, this model will be able to process significantly more data and successfully determine a winner.

Fan: As a basketball enthusiast, our machine learning model will provide its own set of power rankings to assess a team's strength. Using these, a fan making a bracket can better predict which team will win.

Team Director: This model can also benefit the teams themselves by providing insights into what part of their game is underperforming compared to those ranked higher than them. A feature like 2pt % factors heavily into a teams strength and if it is lower than average, the team can focus on improving shooting percentage.

CONCLUSIONS

Overall, our model serves its purpose very well. We managed to train a model that will predict a winner from the pre-season statistics alone. Given how accurate our regression models were, we were able to correctly identify winners out of our 2 highest ranked teams in 8/10 years with a random forest model and gradient boosted model, and 6/10 winners with our SVR model. Given more time, we would have liked to incorporate data regarding upsets as upsets are another element to the tournament that are difficult to predict. By doing this we can train a second model which accounts more for randomness using the seeding generated from our current model.