Predicting the Men's March Madness 2023 Bracket

Hayden Johnson and Nic Carpino, *UCLA MQE*

***Abstract*—Every year since the NCAAB (National Collegiate Athletic Association) Men's Basketball Tournament expanded to 68 teams, chosen colleges around the United States participate in a three-week tournament with desire to be the one crowned champion. This bracket is known as "March Madness". Gamblers have the potential of winning large amounts of money for predicting the winners of games, with millions of aiming for a perfect bracket or attempt to pick the underdog winners in each round before any game has been played. In our project, we utilize decision trees and random forests to determine the most effective statistics one should use to predict the winners of the Round of 64, and beyond in March Madness.**

I. Introduction

The National Collegiate Athletic Association has hosted a Division-I Men's Basketball tournament every year since 1939, which started with a total of eight teams. The field expanded to 64 teams in 1985, while simultaneously gaining popularity in the United States (US) amongst sports fans. In 2019, a total of 17.3 million brackets were submitted to ESPN's Tournament Challenge, and zero were perfect. The tournament takes place in March and spans through the beginning of April each year. Every Division-I Men's Basketball team in the US is given a rank by the *Associated Press* based on their performance throughout their season. From these rankings – along with other metrics, 12 members of the basketball selection committee subjectively pick 68 teams to participate in the tournament. Four teams must compete in what is called a play-in. The play-in games involve eight teams competing for the final four spots before the bracket is finalized. The winner of each game advances to March Madness, forming out the field of 64. Each team is given a ranking, ranging from 1 to 16, along with a region: East, West, Midwest or South. The number one seeds each play the 16th seeded teams. The second ranked seeds play the 15 seeds, and so on. The winner of the round of 64 moves on to play an opponent in a round of 32. The winners of this round move on to a round of 16, then round of 8, the "Final Four, and finally a title game to crown a school the National Champion. March Madness brings in millions of dollars in gamblers' money, with the opportunity of being *the first* to have a perfect bracket simply too good to pass up. Our project employs numerous machine learning models to aid gamblers, analysts, and the average fan win their Bracket challenges and accurately predict winning teams for the 2023 March Madness tournament.

II. Task Description

The machine learning models included in this report are decision trees and random forests. We chose decision trees for their powerful visualizations; we wanted to see what decisions the model was making – and when. While decision trees are effective for classification problems, a major drawback is that decision trees tend to overfit their data. We used the ensemble method of random forests to maintain the decision trees visuals, while the issue of overfitting is mitigated. Accuracy needs to be estimated at a high-level for potential sports bettors to trust our model, and for this reason – we only predict the round of 64 games. Predicting further rounds without certainty of the teams playing lowered our confidence in what we were outputting. K-Nearest Neighbor, along with other instance-based learning algorithms, did not perform well considering the low amount of data (less teams) in further rounds, and predicting the winner was inaccurate.

Our data involves tournament data from 2008-2022. Round 64 has the highest amount of data points, making it an ideal choice for training and testing data due to the large number of games. Additionally, the fine tuning of hyperparameters did not give us an even split of teams no matter how fine the tuning, furthermore, pushing us to stick with the round of 64. With 32 games being played in this round, we believe there are enough discrepancies within the betting market to help you predict March Madness.

III. Major Challenges and Solution

The small amount of data that the Championship and Final Four games provided us led to imprecise predictions. K-Nearest Neighbor did not resolve our issue. While more data points were reproduced, there was a small sample size to begin with, and our predictions for the teams that would qualify for those rounds were also inaccurate. A solution we tried was to iteratively go through the brackets just as the tournaments are setup. Whichever team was deemed the winner would move on to the round of 32, and so on. However, after fine tuning the hyperparameters and threshold, we did not get an even split of teams regardless of the combinations we tried. Consequently, we stuck with the round of 64. A person can exploit the market by correctly predicting the round of 64, also since more games bring in more action on betting market. Since we did not get an even split of teams, our threshold will predict less than 32 teams

to move on to the round of 32 in the 2023 dataset. We believed it was imperative to accurately predict the winners and avoid any false positives, to limit risk for individuals utilizing our model. A final challenge is the limited number of metrics to capture the human element of the game. While team statistics are very good at making predictions, the parameters do not include confidence of players, injuries, moment going into a game and other metrics that may identify an upset or a "Cinderella Story" where a 16 seed makes it all the way to the championship and wins.

IV. Experiments

A. Dataset Description

We extracted our data from *Kaggle,* the source contained all the regular season statistics of teams from 2008 to 2023 and how well each team did in their respective tournaments. From these regular season statistics, we made predictions if the team would win their round and move on to the next round; furthermore, columns included the current round of the team for that year, and which round they were eliminated in. A large component of the columns is the KenPom adjusted efficiency, defense, offense and tempo. These metrics were developed by Ken Pomeroy, an advanced statistician who notably adjusted offensive and defensive efficiency in the world of basketball. *Adjusted efficiency* is the KenPom adjusted offense minus the KenPom adjusted defense to determine how much the team would outscore an average opponent. The adjusted offense is an estimate of the offensive efficiency a team would have against an average Division-I defense. The adjusted defense statistic is an estimate of the defensive efficiency a team would have against an average Division-I offense. Adjusted tempo is an estimate of the tempo a team would have against a team that wants to play an average Division-I tempo. Tempo is defined as how fast a team plays play on the offensive end of the floor. There are also BartTorvik adjusted statistics in efficiency, offense, defense and tempo. Bart Torvik is a basketball statistician who used play-by-play data to develop "Shot Quality" measurements for a team's offensive possessions, along with other advanced NCAA analytics. One of Torvik's metrics that is not included in KenPom's metric is Barthag. Barthag is the power rating or the chance of beating the average Division-I team, with a higher Barthag being more desirable. Wins above bubble is an interesting metric that is the difference in the number of wins a team has compared to the expected number of wins an average "bubble" (team playing in the play-in) team would earn against a given teams' schedule. The last uncommon metric between the two is Elite SOS: measuring the strength of schedule a team plays, against the average Division-I schedule – with higher numbers being elite.

Once the data was imported into Jupyter Notebooks, we began to clean and prepare it for use. There is an extra column added to the end of each row titled 'Team.1' which repeats the name of the team already established in the row. This unnecessary column was dropped. The dataset does not include whether a team won a given round. It has the 'TEAM ROUND' which indicates how far a team made it into the tournament, and 'CURRENT ROUND' which represents the current round a team is. Since we are only predicting the round of 64, we added a column to indicate if a team made it out of the round of 64. If the 'TEAM ROUND' is less than 'CURRENT ROUND', then a 1 is inputted into a new column titled 'WIN'. If not, a 0 is inputted. Because we will eventually be using decision trees and random forests, there cannot be strings in the dataset. LabelEncoder from scikit-learn transforms the team names which are strings into a number that appears in the dataset again. Random forests and decision trees will now accept the dataset. Our decision to only predict the round of 64 led us to drop any results not from the round of 64. While data is lost, we still retain nearly 900 data points. Next, the 2023 teams are separated from all the other years since the teams have not played yet and cannot be included in the training and testing sets. Any duplicates are dropped. Finally, the dataset not containing the 2023 teams is split into X and Y data frames. The X drops the 'WIN' column as that is what we will predict against. The Y data frame only contains the 'WIN' column.

B. Evaluation Metrics

Decision trees and random forests are used to train and test the data, as well as predicting the 2023 bracket. Decision tress and random forests are plural because we trained, tested and predicted on scaled and unscaled data to compare the results. Scikit learn is used to split our data into training and testing. Additionally, the random forests, decision trees and scaling are done using the Scikit learn packages in Python The decision trees and random forests fitting and predicting capabilities do not allow for strings. Scikit learns Label Encoder allow us to maintain the names of each team as a number, and we can transform the name of the team and the number back and forth as needed. To ensure we got the best models, we utilize GridSearchCV to fine tune hyperparameters. The decision trees' hyperparameters are criterion, max depth and class weight. The criterion is gini and entropy. Both gini and entropy are used to evaluate the quality of split in the decision trees. Our max depth ranges from 1 to 15, indicating how many branches were allowed. The class weight is balanced to mitigate the effect of class imbalances. The random forest hyperparameters involve n_estimators, max depth and max features. The n_estimators range from 50 to 150, indicating how many decision trees should be produced in this ensemble methods. The max depth indicates how many branches the decision trees can make ranging from 2 to 20. The max features could either be sqrt or log2. Max features are the maximum number of features to be considered when looking for the best split at each node of the decision tree. Sqrt relates to taking the square root of the maximum number of features in the dataset. Log2 specifies that the maximum number of features to be considered at each split is the logarithm base 2 of the total number of features in the dataset. After the hyperparameters are tuned, the models are fitted on the training set. Next, the actual probabilities of a win, no 0 or 1 but the actual probabilities, are predicted. After getting these probabilities, we use receiving operation characteristic (ROC) and area under the curve (AUC) to acquire the best threshold to indicate a win or loss. The

threshold which maximizes the AUC curve would be the best threshold for predicting the testing set. After fitting the predicting, the best accuracy, mean squared error (mse), and threshold are printed. A confusion matrix is constructed with the best metrics and the accuracy, recall and precision are recorded. For the decision trees, a decision tree figure is plotted. Next, a dataframe is constructed with the team names transformed back into strings and the predicted outcome of a win or not. The feature importance is charted using a bar chart. Feature importance shows what the model determined as the most influential input or column that will affect a team's chance of winning. We can compare what the decision trees, random forests, scaled, and non-scaled data value more. Our last form of comparison is plotting the ROC curve and printing the AUC. The ROC curve plots the true positive rate against the false positive rate to discuss how the model distinguishes between positive and negative samples at different classification thresholds. The AUC assists in evaluating the performance of the binary classifications of a win or loss.

C. Major Results

Scikit-learn's, matplotlib, numpy, and panda's libraries and packages are used to conduct all the analysis. The data is randomly split with the testing sample receiving 20% of the X and Y data and 80% to training. A random state is established to maintain the same train and test sets across different executions

1. Decision Tree without Scaled Data

Our first analysis is conducted with a decision tree without scaled data. We want to compare scaled vs unscaled data to test the differences in accuracy, mse and other metrics. After training the data, the predict_proba method is used to return the probabilities for each data point. We did not want to use the default threshold of 0.5 to predict a 0 or 1 for a win. Instead, we wanted to fine tune the hyperparameter to get the threshold with the highest accuracy. GridSearchCV, ROC curves and AUC fine tune the hyperparameters and produce the best threshold. The decision tree without scaled data found the threshold to be 0.72 with an accuracy of 81.01% and a mse of 0.1899. While the accuracy may not seem that impressive, most bracket winners every year average an accuracy of 49.8%. It also means the decision tree predictions are better than just flipping a coin with a random guess. The mse is 0.1899. The lower the mse the better, so we can compare it to the other models later. The testing sample predictions are 103 teams will lose and 76 will win. This is not perfectly balanced, but it is suitable. The confusion matrix reports 78 true negatives, 9 false negatives, 25 false positives and 67 true positives. Recall is 0.7283 and precision is 0.8816. The model chose a max depth of 3 to be the best. There are 44 columns that the model uses to predict winners. Here are the top 10 features the model values:

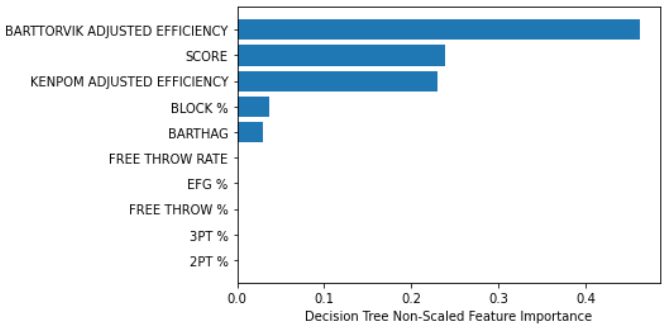

Fig 1. Feature importance chart for the non-scaled decision tree. Bart Torvik Adjusted Efficiency and Score have the greatest effect on decisions made in the tree.

The features suggest only five out of the 44 columns are important in determining a winner. The decision tree without scaling seems to favor a team that can outscore their opponents and who can score a lot since the top three are the adjusted efficiencies and score. Blocking shots aid in efficiency as a team is outscoring their opponent. Barthag appears to be a solid metric for determining who is an excellent team. The ROC curve can be seen below:

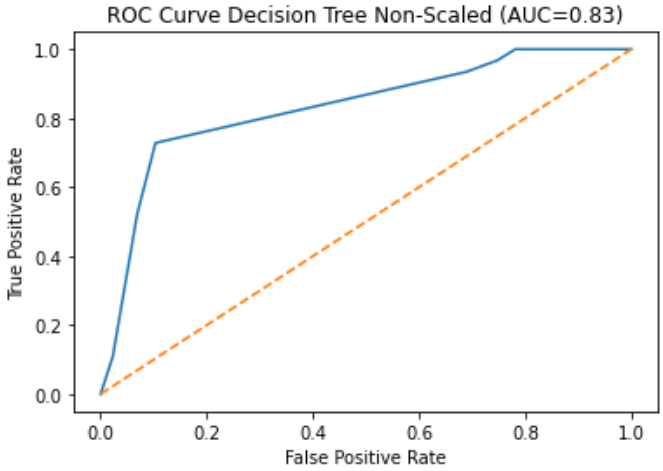

Fig. 2 ROC curve does not follow a 45-degree angle. This indicates it is better than a coin flip. However, it does not come close to the top left corner, so accuracy is not 100%

While this ROC curve does not shoot straight up to the top left indicating 100% accuracy, it by no means follows a 45-degree line. The AUC of 0.83 suggests the decision tree is good at distinguishing between wins and losses. The model can identify a large portion of true positive instances while minimizing false positives.

Predicting the round of 64 for the 2023 teams is conducted next. The best model that contains the greatest performing hyperparameters is implemented on the 2023 teams to predict the probabilities. A threshold of 0.89 to indicate a win or a loss is used. This threshold is larger than the best threshold that maximizes the AUC, but the best threshold from the training and testing set does not predict enough teams to win. The threshold is very sensitive as well. If a 0.90 threshold is used, over 40 teams are predicted to win. Any threshold less than 0.89 only predicts seven. The 0.89 threshold predicts 11 teams to win

and 57 teams to lose. While 11 is still well below the 32 mark of splitting the teams perfectly in half, we felt it would be better to try to get our predictions 100% correct instead of guessing of 40 plus teams to win.

2. *Decision Tree with Scaled Data*

The training and testing data is scaled using the StandardScaler method. After scaling the data, the decision tree is fitted and predicted in the same way as before, but this decision tree uses scaled training and testing samples. Eerily the best threshold is 0.72 with 81.01% accuracy and a mse of 0.1899. Getting the same threshold, accuracy and mse could be because the data was already scaled properly before importing it. However, when we see the scaled vs. unscaled data in the random forest models, they threshold, accuracy and mse are very different from each other. Another point could be that decision trees are not as affected by feature scaling like other machine learning models. This explanation seems the most reasonable as the random forest will demonstrate that scaling the data does make a difference in terms of performance. For the testing samples, the model predicts 103 teams will lose and 76 teams will win. Again, not perfectly balanced but the sample is randomly taken. The confusion matrix reports 78 true negatives, 9 false negatives, 25 false positives and 67 true positives. Recall is 0.7283 and precision is 0.8816. The model chose a max depth of 3 to be the best.

There are 44 columns that the model uses to predict winners. Here are the top 10 features the model values:

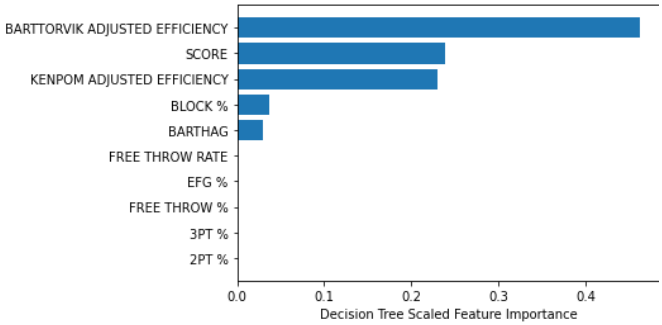

Fig 3. Feature importance chart for the non-scaled decision tree. Bart Torvik Adjusted Efficiency and Score have the greatest effect on decisions made in the tree.

Just as the non-scaled decision tree, Bart Torvik Adjusted Efficiency, Score, KenPom Adjusted Efficiency, Block % and Barthag makeup the top 5 and all other metrics are disregarded. The chart appears to be the same. The ROC curve and AUC have a similar story:

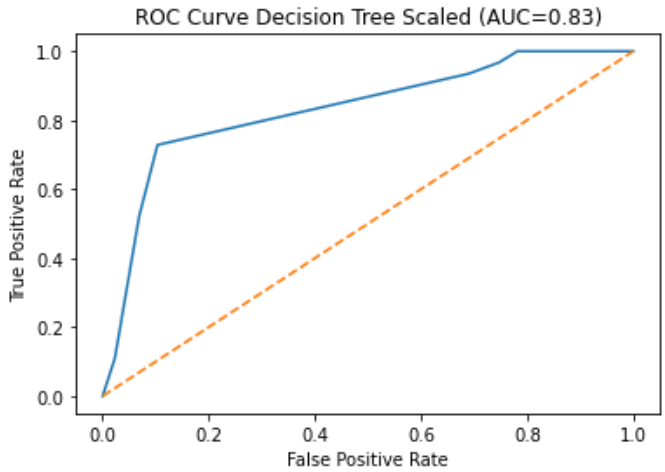

Fig 4. ROC curve does not follow a 45-degree angle. This indicates it is better than a coin flip. However, it does not come close to the top left corner, so accuracy is not 100%

The AUC is 0.83 and the ROC follows the same curvature. Using best hyperparameters, the 2023 bracket is predicted. This prediction differs from the non-scaled data. The threshold to determine winners and loser is 0.34, meaning the cutoff to win is a lot lower than the non-scaled data. Only 6 teams are chosen to win and 62 or chosen to lose. Even with this low of a cutoff, very few teams are chosen to win. Increasing the threshold to 0.35 predicts over 45 teams to win. A threshold of 0.33 omits almost all teams. Again, we feel that getting the predictions 100% correct is more important than guessing more teams than are allowed to win in the round of 64.

3. *Random Forest without Scaled Data*

The Random Forest learning method uses multiple decisions trees to make predictions, as each tree is built using a subset of features and a random subset of the training set. In this model, random forest without scaled datal, we look at ten features: in contrast to the decision tree that looked at five features. Random Forest reduces the variance and the overfitting that previously occurred in our models, and the final prediction from this model is generated by taking the majority vote predictions of all individual trees. The best threshold is 0.64 with 54.19% accuracy and a mse of 0.4581, which means it correctly predicted the outcome of the games more than half of the time. The confusion matrix revealed that the model correctly predicts 91 teams as winners (true positives) and 6 teams as losers (true negatives). However, it incorrectly predicted 81 teams as winners (false negatives) and 1 team as a winner (false positive). This large number of false negatives suggests that this model have been overly conservative with its predictions. Recall is 0.9891and precision is 0.5291

There are 44 columns that the model uses to predict winners. Here are the top 10 features the model values:

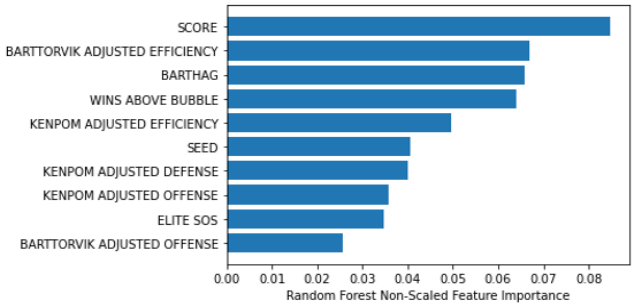

Fig 5. Feature importance chart for the non-scaled Random Forest algorithm. Besides the Score, the Bart Torvik Adjusted Efficiency, Barthag, and Wins Above Bubble have the greatest effect on predictions made by this model.

KenPom Adjusted Efficiency, Seed, KenPom Adjusted Defense, KenPom Adjusted Offense, Elite SOS, and Bart Torvik Adjusted Offense make up the rest of the metrics. ROC curve and AUC have a similar story:

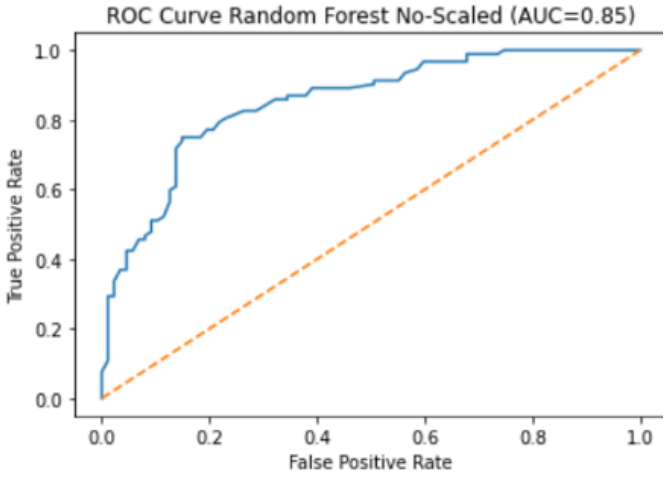

Fig 6. ROC curve does not follow a 45-degree angle, indicating it performs better than random chance. Again, it does not come left to the top left corner and has room to further improve.

The AUC is 0.85, while the ROC follows a similar shape – indicating that his model performed well distinguishing between positive and negative classes. The fact that the curve above does *not* follow a 45-degree angle, means that the model is better than random chance which would be represented by a straight diagonal line. The ROC curve for the random forest model without scaled data proves that the model is performing better than a coin flip; however, there's is still room for improvement in terms of accuracy.

4. *Random Forest with Scaled Data*

Using scaled data, we improved the performance of our Random Forest model. Again, this method looked at the top ten features, rather than merely five. The best threshold is 0.63 with 77.1% accuracy and a MSE of 0.2291, which means our model correctly predicted the outcome of March Madness games nearly four out of five times. The MSE is also a significant improvement over the non-scaled model, indicating that this model's predictions are closer to the actual outcomes. The confusion matrix revealed that the model predicts 62 teams as winners and 76 teams as losers, with only 11 false negatives and 30 false positives – suggesting the model is more balanced in its predictions, with fewer false negatives that before. The recall of 0.6739 and precision score of 0.8493 are also improved over the non-scaled model, indicating this model is doing a better job of identifying true positives and minimizing false positives. Looking at the top ten features that this model generated – Wins Above Bubble, along with KenPom Adjusted Efficiency, are important factors in determining March Madness outcomes. The inclusion of a team's seed and Bart Torvik Adjusted Efficiency are interesting, are interesting – suggesting that these factors are more important than KenPom adjusted defense and offense. Overall, the scaled data significantly improved the performance of our Random Forest model for predicting the tournament.

There are 44 columns that the model uses to predict winners. Here are the top 10 features the model values:

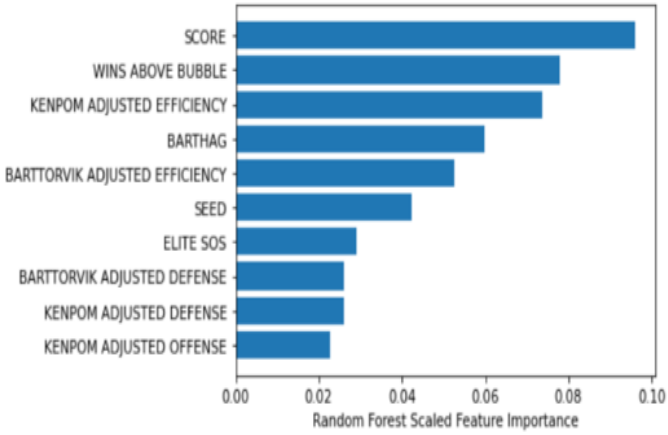

Fig 7. Feature importance chart for the scaled Random Forest. The increase importance of Wins Above the Bubble suggesting that a team's regular season wins are more important than their seeding.

The ROC curve and AUC share a similar story:

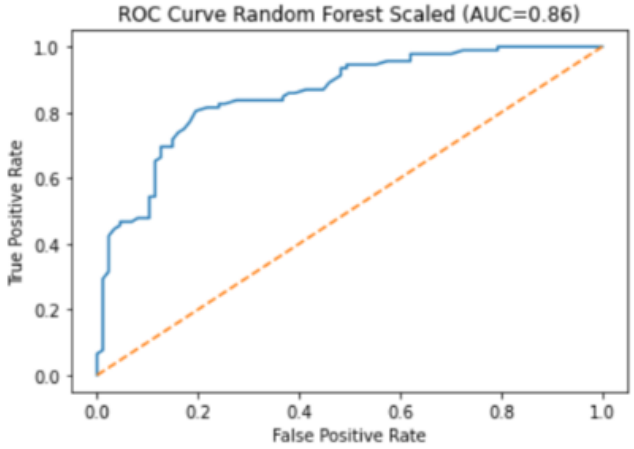

Fig 8. Similar to the previous model, the ROC curve does not follow a 45-degree angle, indicating the model is consistent across different thresholds. This indicates it is better than random chance and performs well distinguishing between positive and negative classes.

The AUC is 0.86, a slight improvement from the non-scaled Random Forest model. While the ROC follows a similar curvature not following a 45-degree angle, it again means it performed better than random chance. As with the previous model, the ROC curve does not come close to the top left corner, indicating its accuracy is not perfect. Although this is expected, our ROC curve suggests that our Random Forest model with scaled data is making accurate predictions.

D. Analysis

The accuracy between the models is quite shocking. The decision trees with both scaled and unscaled data outperformed both random forests. The scaled and unscaled decision tree accuracy is 81.01%, while the unscaled random forest accuracy is 54.19% and the scaled random forest accuracy is 77.09%. With the ensemble method of random forests, it takes the best of all the decision trees to classify a win or not, while the sole decision tree models only use one decision tree to classify. However, the ROC and AUC favor the random forest models. The scaled data random forest model has an AUC of 0.86. This AUC is greater than the unscaled data random forest model AUC: 0.85 and the scaled and unscaled data of the decision trees' AUC: 0.83. Accuracy measures the overall correctness of the model's predictions, correctly classifying true positives and true negatives. The AUC measures the ability of the model to distinguish between the positive and negative classes by plotting the true positive rate against the false positive rate at difference classification thresholds. The random forest non-scaled data has such a high AUC and low accuracy because it appears to be a serial predictor of wins. 172 teams in the testing sample are predicted to win and 7 are predicted to lose. The non-scaled data modeled with a random forest is not very reliable. This anomaly does not explain why the decision trees' accuracy outperformed the random forest with scaled data's accuracy, while the random forest with scaled data had a better AUC. The answer lies in the ROC curves. Looking at the ROC curves for the scaled and non-scaled data for the decision trees, the ROC curve starts off steep but plateaus around the 0.1 false positive rate and the 0.7 true positive rate. The scaled data for the random forest model does not plateau but gradually increases along the x-axis. This ROC curve does not start of very steep either. The decision trees' ROC curve indicates that the model is making errors at higher thresholds. These two models have the highest threshold of any of the models. A higher threshold makes it harder to upsets to happen, or a lower seed team beating a higher seed team. The decision tree models are not able to accurately predict these upsets. Referring to the important feature charts for each of the models, the random forests were able to utilize all the features in determining whether a team would win. Conversely, the decision trees chose five features and eliminated the rest. Both types of methods value efficiency, scoring and a high-powered offense. College basketball games are relatively short compared to NBA games, so teams need all the points they can get because there is less time to come back and win the game from behind; however, the random forests also put emphasis on the defensive metrics for teams. KenPom adjusted defense, BartTorvik adjusted defense and elite SOS have major roles in the random forests' models of predicting a winner. If college teams have had a tough regular season schedule, having seen many elite teams, and can play defense well, they have a better chance of winning. This makes intuitive sense. If a college team consistently plays against challenging opponents throughout a season, then the March Madness Bracket should be virtually the same as the regular season, because that team has already played against the best. A team that can prevent the other team from scoring, when scoring is one of the best and sometimes the best indicator of a team winning or not, then they will have a much better time of winning. The only defensive metric the decision trees include is the block % metric, which is not even in the top 10 for the random forests. The mse for the non-scaled random forest is horrendous at 0.4581, making clearer that the model is not very good. The random forest with scaled data does have a higher mse at 0.2291 compared to the 0.1899 of the decision trees. The tree-based models, both decision trees and random forests, have some level of randomness in their training process. The different initializations of the two may lead to different results, thus the small difference in mses. The final metric to look at is how well the model's predicted their 2023 teams. The round of 64 is over and the predictions can be tested. The non-scaled decision tree predicted 11 teams to move on to the round of 32. It predicted 9/11 correct. Arizona and the biggest upset in tournament history Purdue did not make it out of the round of 64. The scaled data predicted 6 teams to make it out of the round of 64. 4/6 teams made it out with Arizona and Purdue spoiling the brackets of many fans. Recall, the thresholds for the decision trees are very anal, and decreasing the threshold slightly to allow more teams to be predicted as a win causes well over 40 teams to be in the win column. We want to be as accurate as possible in predicting the teams, and we did not want to make incorrect predictions, so we decided to limit the number of available teams to win at 32 or less. The non-scaled random forest predicted 30 teams to win their first game. 21/30 teams were correctly predicted. Finally, the scaled random forest model chose 32 teams to win. 24/32 moved on to the next round.

5. Conclusion and Future Works

This report explored the use of decision trees and random forests in predicting the winners of March Madness games. The decision tree model was chosen for its powerful visualizations, but it was found to have the drawback of overfitting the data. To address this issue, the ensemble method of random forests was used. The analysis focused on predicting the round of 64 games, which had the highest amount of data points and therefore provided an ideal choice for training and testing data.

The experiments conducted in this report involved comparing the performance of decision trees and random forests with and without scaled data. The results showed that both decision tree models had similar accuracy and mean squared error values. The best threshold for predicting a win or a loss was found to be 0.72. When it came time to predict the

2023 March Madness games, the threshold for the scaled and unscaled decision tree models were quite different. They also predicted different teams to win. The random forest model with unscaled data struggled to make accurate predictions and maintain a low mean squared error. The model almost always predicts that a team will win; however, the scaled random forest model performed very well. With only a slightly higher mse than the decision tree models and lower accuracy, this model is adept at distinguishing winners and including upsets. It can also produce 32 teams to win in the 2023 tournament.

Overall, the decision tree models and scaled data random forest model performed well in distinguishing between wins and losses. For the future, our team hopes to be able to predict further into the tournament and choose a single winner. Furthermore, we strive to make every win to be as highly likely as possible. In order to do this new metrics, need to be made. Considering for injuries of players, moment teams have going into the tournament and games, and some way to measure overall morale and competitiveness can help to improve model accuracy.

References

Amin, N. (2023, March 1). March Madness Data. Kaggle. Retrieved March 3, 2023, from https://www.kaggle.com/datasets/nishaanamin/march-madness-data

Benz, L. (2023, March 17). YUSAG Bracketology | Yale Undergraduate Sports Analytics Group. Retrieved March 17, 2023, from https://sports.sites.yale.edu/yusag-bracketology

Broome, A. (2019, October 23). Kenpom 101: What the college basketball metric system is and how it ranks Michigan. Maize n Brew. Retrieved March 2, 2023, from https://www.maizenbrew.com/2019/10/23/20928669/kenpom-explained-what-it-means-michigan-basketball-ranking

Liu, X. (2023). Econ 425T: Machine Learning. Los Angeles; UCLA.