March Machine Learning Mania 2022 - Men's

Ted Toohill, ttoohill[1]*

Abstract
This reports sole purpose is to find a way to predict the outcomes of the NCAA® tournament based on statistics and data they have gathered. The difficultly in predicting these games is affected in so many ways, so the ultimate goal is to have better predictions as to the winners rather than a 50/50 coin flip. It was interesting to see how certain teams fared in certain outcome when compared to others. There were instances where my predictions of certain games favored lower seeded teams, whereas others had very clear winners. I found that certain team stats were extremely relevant while others were a lot less crucial in determining an outcome. Ultimately I was able to finalize my algorithm using gradient boosting decision trees to record a binary log loss of 0.685, according Kaggle's log loss measurement. This means that based on the outcomes of the 2022 tournament I was able to predict roughly 63.5% of the games correctly.

[1] *Computer Science, School of Informatics , Computing and Engineering, Indiana University, Bloomington, IN, USA*

Contents

1. Problem and Data Description

1.1 Understanding the Problem

1.1.1 Tournament Description

The NCAA® Men's Basketball Tournament, otherwise known as March Madness [1]. It is a single elimination tournament where 64 Division-I teams compete for six rounds to determine who has the best team. Each of the 64 lucky teams are placed into one of four different divisions (West, South, East, Midwest). The division a team is placed in is random, but the division itself determines where the teams will compete. As of the 2017 season, 32 teams receive automatic bids to the tournament by winning their conference tournament. The remaining 32 teams are chosen by the Selection Committee. After being selected, the Committee then decides the seed of each team (1st-16th). The seed of each team determines who they will play in the first round of the tournament, with the lowest seed playing the highest and so on. The rounds of the tournament are, in order, the "Round of 64", the "Round of 32", the "Sweet 16", the "Elite Eight", the "Final Four", and the "Championship". The winner of the "Championship" is the winner of the whole tournament.

With the start of the 2022 NCAA® tournament there was a rule change allowing for 68 teams to be selected (instead of 64). This change added four more games (known as the First Four) to be played before the "Round of 64" creating a seventh round in the tournament. However, since most of the data is based on a 64 team selection, that is what I'll refer to.

1.1.2 Problem Description

The goal of this report is to create a model that is able to predict the likelihood of each teams chance of winning in every possible match-up within the tournament.

From a statistical perspective, the odds of determining a perfect bracket are 1 in 9.2 quintillion. This means that there are 9.2 quintillion different combinations of possible outcomes. To put that number into perspective, a group of researchers from the University of Hawaii estimated there to be 7.5 quintillion grains of sand on the Earth. Meaning that you would have higher odds of selecting a specific grain of sand somewhere in the world than creating a perfect bracket.

Outside of the magnitude of possible outcomes, the process of predicting games becomes very difficult because some games come down to a lucky half-court shot or a bad call from a referee. In a perfect world the better seeded team wins every time, however, as those who watch and participate in March Madness know, there are many upsets. Unfortunately

accounting for every possible influence on a game is simply unquantifiable. In this project we will mainly consider factors that have a high importance on the outcome of a game through the analysis of historical data.

1.2 Understanding the Data

Most of the data that will be used is referenced at later points in this report, however it is worth mentioning that all data in this report is from 1985-2021.

The training data sets will consist of the more important season averages along with personal calculation. Although we are given data from 1985 season; the latest tournament data is from 2003, therefore the training set will contain the results of the 2003-2021 NCAA® tournaments (excluding 2022 since there was no tournament due to Covid-19). The training set will also include a unique classifier column that determines whether team 1 won the game or not.

After creating a training set in our model we will use the same training set to then create predictions for the 2022 tournament. The results of these predictions will be formatted within two columns of a data set. The first columns is a string containing the year of the tournament followed by the team ID's of both teams playing, with the lower of the two team IDs being listed first. The second column will be the calculated win percentage of the lower ID'd team.

2. Data Preprocessing & Exploratory Data Analysis

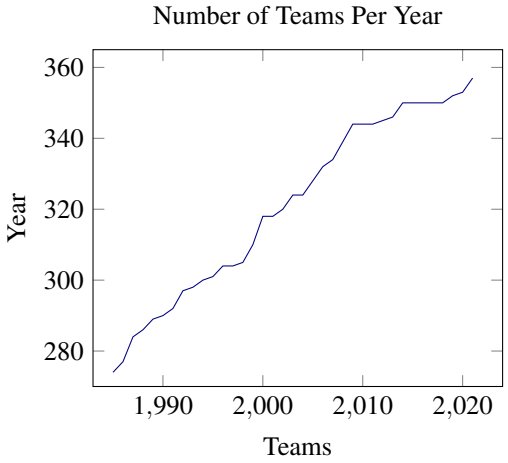
2.1 Data Preprocessing

2.1.1 Handling Missing Values

This report will use complete-case analysis when encountering missing values. Although this method will slightly reduce the precision of certain estimates, after looking over the data there seems to be very little amounts of missing data. Additionally, there is an easy implementation of complete-case analysis, and using this method of handling missing values will remove potential bias and/or skewed results with respect to the completed data. In order to properly implement this method, we will assume entries with missing data will be considered as a random sample of the data that was intended to be observed. This will at the least ensure that the likelihood of data being missing is independent of the outcome and situation, therefore removing any more potential bias in the collection of the data.

2.1.2 Removing Unwanted Data

The hardest part about predicting games is deciding what data is important or not. Here since we are only given the results of the 2003 and on tournaments, we will only use data since the 2003 season. Another important thing to note is that the number of Division-1 teams fluctuates each year so certain teams may have more data than others.



As a result only the data regarding the current teams will be accounted for, even though there is data for colleges that are no longer eligible for the NCAA® tournament (e.i. no longer Division-I colleges). As a way to differentiate historically good teams and teams that are simply having a great season, each teams stats will be based on a season average. This eliminates the possibility of a historically great team having high averages during a bad season. Additionally, as a way to not exclude information regarding teams who could make future tournaments, all teams that have become Division-I as of 2021 will be accounted for based on the data acquired since they've become Division-I.

Data containing cities, game location, and staff names will be ignored. Most of this data is won't contribute to the prediction of each outcome in a game. However, information regarding a teams winning percentage at home versus on the road will be accounted for.

2.2 Exploratory Data Analysis

The main data sets being used are "MRegularSeasonDetailedResults.csv", "MNCAATourneySeeds.csv", "MMasseryOrdinals.csv", "MNCAATourneyDetailedResults.csv", and "Teams.csv". These data sets come from Kaggle.com within their March Mania competition. The data sets with "detailed results" contain the box scores of every match-up and will be used to analyze each teams performance on a game to game and season to season basis. The TourneySeeds data set will be used to identify which teams make the tournament each year and how they've been seeded. Finally the Teams data will solely be used as a way to identify teams in other data sets using a unique "TeamID" variable held in the first column. One important thing to note is that there are no missing values in these data sets so there should not be any unintentional bias.

To start off, the teams data set has four columns: two for team identification(team name and team ID) and two for how long a team has been a Division-I school(first and last D1 season). It is worth noting that the team ID's will only range from 1000-1999. This is so there is no confusion with other data sets that or given. For teams that were Division-1 before 1985 their first D1 season will be 1985 and teams that are currently Division-1 will have a last D1 season of 2022. As

said before this data set will only be used to identify teams in other data sets.

TeamID	TeamName	FirstD1Season	LastD1Season
1101	Abilene Chr	2014	2022
1102	Air Force	1985	2022
1128	Birmingham So	2003	2006

Table 1. Teams Data Set.

The detailed results data sets contain the most information as each one holds the box score for each game every team has played since the start of the 1985 season. This data set, of 34 columns, tells us who won, which team they played, the scores of both teams, and both teams in game stats. Most of the data processing will originate from these two data sets.

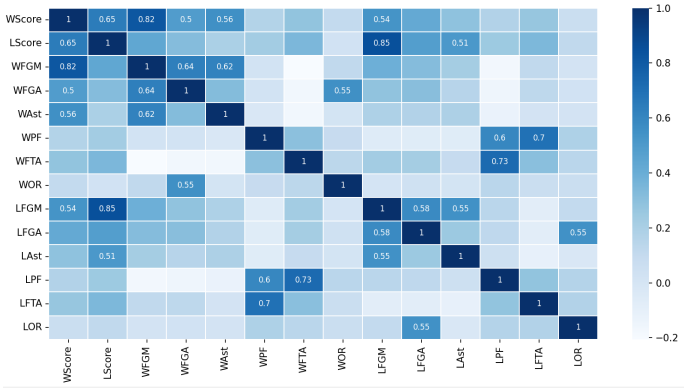


Figure 1. Heatmap of Certain Regular Season Stats

Looking at the heat map, there are some correlations between certain statistics that we will use to better our predictions. An obvious one that goes without saying is the correlation between a teams score and the number of field goals they make. Alternatively, there are some not so obvious like the positive correlation between the winning and losing teams personal fouls and that winning teams have a higher correlation between assists and points than losing teams do. This could imply that teams who win typically pass the ball better.

The tourney seeds data set will also be used as a weight on the potential outcome of each game. Although the detailed stats should speak for themselves, professional sports analysts ran the numbers and thought the seeding was most beneficial in the order the teams are placed in. Therefore the seeding of a team will have an effect on the outcome of a game. In fact there has only been one instance where a 13 seed team beat a 4 seed team, which will be referenced later in the report.

On top of the tournament seeds the Massey ordinals data set will be used in a very similar way. This data set contains the daily rankings of each team each season from multiple different ranking systems. I have no knowledge of any faults or inaccuracies in any of the rankings systems within this data set so I will take the mean of each teams rank per day based on all the different systems.

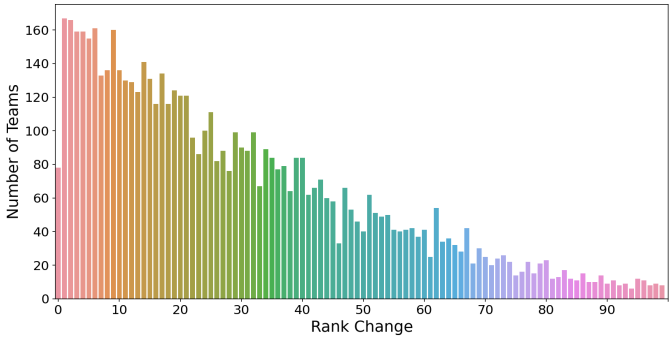


Figure 2. Difference Between First and Last Week Rankings

In the count plot above it is clear that lots of teams ranks change from the start of the season towards the end. This creates two different approaches on how to use this data set. The first approach would be to take the mean of the last weeks rank for each team and use that as the weight. This approach would show the most accurate results on the current state of that team since the last week of the season is the closest to the tournament. The second is taking the mean of every days ranking and creating an average season rank. This would show how well each team performs throughout an entire season. However as the graph above shows, teams ranks change a lot since the beginning of a season, so to reduce that variance we could take the average season rank starting from the second or third week of the season. Athletes need time to adjust the their teams and get back into the swing of things, so after a 1-2 week period their minds and body's are better prepared.

When measuring how good each team is it is important to focus on the wins. As mentioned earlier, more and more colleges are becoming Division-I colleges, so simply counting the number of wins would benefit teams who have been Division-1 for longer. This is why it will be a better metric to compare teams win percentages, rather than their number of wins.

Team	Wins	Losses	Win(%)
Kansas	949	224	80.90
Duke	958	235	80.30
Kentucky	898	281	76.17
Arizona	878	283	75.62
Gonzaga	831	270	75.48

Table 2. Best Regular Season Win Percentage.

Team	Wins	Losses	Win(%)
Duke	97	29	76.98
Connecticut	55	17	76.39
North Carolina	91	29	75.83
Kentucky	83	27	75.45
Loyola-Chicago	8	3	72.73

Table 3. Best Tournament Win Percentage.

One could argue that the more wins the better the team; like we see with Duke and Kentucky. Generally that may help when considering how great a team has been, but this won't help when predicting how well a team will perform in the current season. Loyola-Chicago is a good example of why total wins is an ineffective measurement. They've been in the tournament three times since 1985 (excluding the 2022 tournament), but have the fifth highest winning percentage. So calculating any statistics by the number of wins or tournament appearances won't provide accurate results.

Another important factor to consider is a teams ability to win on the road versus at home. Some teams get the benefit of playing close to home while others have to travel around the country.

Season	Team	Home Win(%)	Away Win(%)
1985	Air Force	18.18	23.08
1985	Akron	46.67	25.00
1985	Alabama	71.43	66.67
1985	Alabama St	44.44	33.33
1985	Alcorn St	66.67	84.62

Table 4. Teams Win Percentages at Home vs. Away.

After processing data on teams per year, its clear that some teams performed better on the road in certain season than others and vice versa. Knowing the number of home and away games a team plays could have a correlation between their win percentage in that current season.

After analyzing the home and away wins per team, it was interesting that most teams had a higher win percentage at home. In Figure 3, the percentage of teams that won at home were much greater than the percentage of teams that won on the road. Some games were played in a location where both teams were away and this is labeled as 'Neutral'.

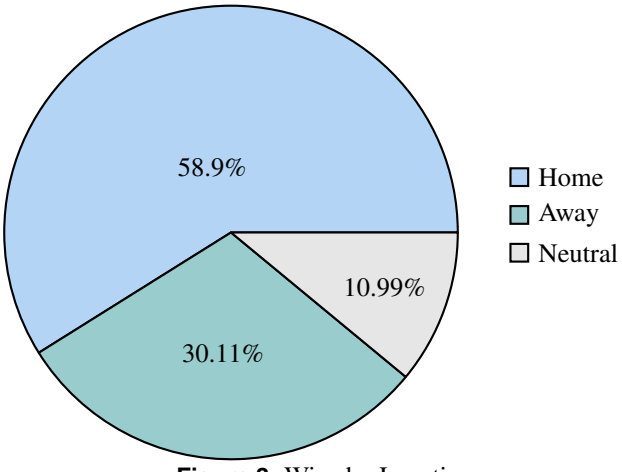


Figure 3. Wins by Location

Knowing this we must also consider how close many games can be. As mentioned earlier, the outcome of a game could be determined by a lucky shot or a bad call. Looking at the total scores in both tournament games and season games, there seems to be a large amount of games with a small differences between the score of the winning team and the losing team. Of course there are always outliers, but it is worth noting that predicting the outcomes of games may come down to a close call in the end of a game.

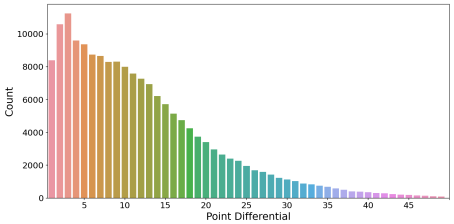


Figure 4. Season Point Differentials

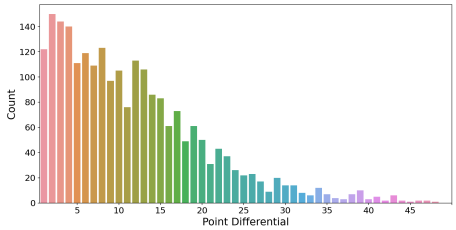


Figure 5. Tournament Point Differentials

As we can see there are many close games which is why it is important to understand the importance of every piece of data. In these graphs notice how the point differential varies slightly more in the tournament than in the regular season. Point differentials for the figures above were calculated by subtracting the score of the winning team from the score of the losing team, tied games were not accounted for in these plots.

3. Algorithm and Methodology

To complete this prediction I had to create adjust some of the data sets containing team stats and rankings as well as add some research based basketball calculations. I performed this project in python however the same results could be replicated in other coding languages.

In the process of creating my training data set I used data frames from the regular season detailed results (all regular season game box scores since 2003), the tournament detailed results (all tournament box scores since 2003), the tournament seeds (every teams seed since 2003), the Massey ordinals (each team's per season rank among all Division 1 teams), and the sample submission file (containing all possible tournament match-ups). All of these data sets were downloaded from Kaggle in 2022 March Machine Learning Mania competition.

3.1 Creating the Training and Testing Data

Basic stats like steals (Stl), blocks (Blk), and personal fouls (PF) are averaged per season. More complex stats like field goal percentage are calculated by dividing the number of field goals made (FGM) by the number of field goals attempted (FGA).

I researched the most effective measurements to determine a teams likelihood to win and found one that combined multiple stats to create what is called a possession [2]. A possession starts when a team gets the ball and ends when they lose the ball or the quarter ends. This means that if a team shoots the ball and gets the rebound it is still considered the same possession. Dean Oliver, discusses the correlation between possessions and the outcome of the game. Calculating possessions will explain a teams pace of play and how effective they are with the ball. This calculation uses a teams field goal attempts (FGA), offensive rebounds (OR), turnovers (TO), and free throw attempts (FTA). The formula is:

$$Possessions = FGA - OR + TO + (FTA * 0.44)$$

The 0.44 multiplied by the free throw attempts is a universal number that Dean Oliver created to get better results. It assumes that most free throw shots are made but is only at 0.44 because a free throw shot that is rebounded by the offensive team still counts as the same possession. According to multiple sources this number is a very good predictor of a teams performance.

Possessions only tell part of the story, since a team who plays fast usually has more possessions than a team who plays slow. This is why the offensive and defensive ratings of a team are important. These ratings are calculated by points per 100 possessions. It may seem like a weird measurement when compared to points per possession, however a team that scores 98 points per 100 possessions is a much cleaner number than 0.98 points per possession since you can't score 0.98 points. This number accounts for pace of play and ultimately looks at the pure offensive and defensive abilities of each team.

After the box score season averages are taken, the teams seeds are taken and the difference between them is computed. The higher the difference the larger chance the lower seeded team had of winning. The same is also true for the Massey ordinal ranks for during the season rankings. This measurement is experimented with later.

The last step in preparing the data sets is to merge all the data

3.2 Predicting the Outcomes

The data set created from exploratory data analysis, contained various season averages and rankings from both teams as well as some personal computations stated earlier to provide as accurate results as possible. For this model, I used a built-in gradient boosting decision tree algorithm [3] to train and test my data. Gradient boosting uses a series of weak learners that depend on each other in a cumulative manner. These weak learners have a high bias and low variance. The built-in function that was used is called 'lightgbm' and is very similar to the 'XGBoost' machine learning algorithm. Both algorithms use advanced regularization, which improves model generalization, and are feasible with large sets of data. However, lightgbm stands out more not only because of its efficiency but also its scalability and minimized memory usage.

The LGBM or Light Gradient Boosting Model capitalizes on two techniques [4]. The first being GOSS or gradient based on side sample. This technique will exclude a relatively large portion of the data that has small gradients and use the remaining data as small gradients in the predicting process. The more important data is primarily used to estimate the overall information gain. The second technique is Exclusive Feature bundling (EFB). This technique regroups mutually exclusive data into bundles to be treated as a single feature. However, one should still be able to retrieve the original values of the features from the bundle. Combining both of these techniques drastically decrease the computation time with very minimal loss. Also due to the sheer quantity of stats collected per game in each data set, the LGB model is much more suitable.

When implementing this built-in algorithm I included many parameters, to try and minimize the run-time without drastically impacting the accuracy of the model. I adjusted to bagging and feature fractions, moving them from 1 to 0.9, to allow for a little more variance or more-so chance for a lower preforming team to still be predicted as a winner, since there are many upsets throughout the tournament. I also changed the learning rate [5] from 1 to 0.01 to prevent the model from converging too quickly over the large amount of data. Although a high learning rate decreases the risk of overfitting, if it is too high at any point it will level out and begin to ignore further inputs. Having too low of a learning rate is also bad since it will get stuck and never converge, so finding a middle-ground will provide the best results.

Another helpful feature in LGBM is the early stopping round parameter. Using this parameter if the validation score doesn't increase after a certain number of rounds the algorithm will stop training. This will decrease run-time while not affecting the accuracy at all since the validation score won't be changing for an extended period of time. An early stopping round parameter that is too low will however affect the accuracy. Finding an attribute that isn't too low is just another benefit to the LGBM algorithm.

The other built-in algorithm I used in modeling is the KFold algorithm which will further split my training data into more training and testing data. I used this to be able to test multiple folds in order to find the most optimal iteration which will then be used in the process of predicting the outcome.

The score of this project is based on the log loss of each prediction outcome and is calculated by this formula:

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

4. Experiments and Results

In this project different statistics had different affects on the outcome. More specifically there were certain features that had minimal importance in determining the outcome of a game. Looking at Figure 6, the most important factors were the teams rankings and the seed they were place in. Outside of those the most important stat is the offensive and defensive calculations we determined earlier from Dean Oliver [2].

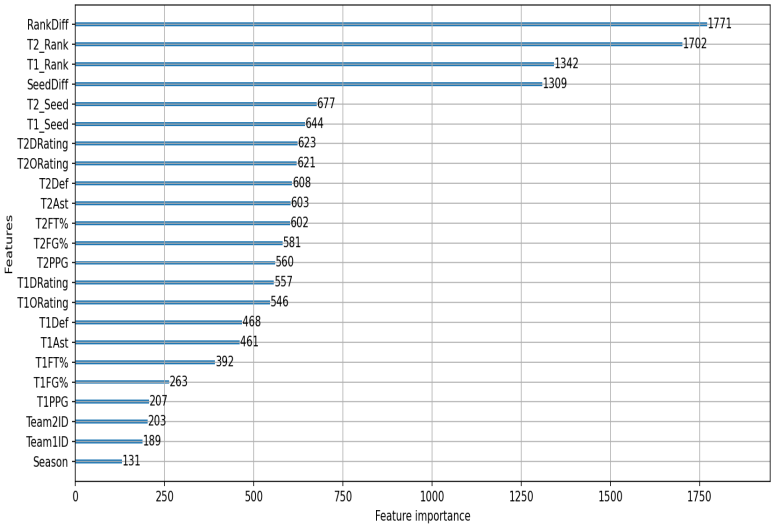


Figure 6. Final Data Set Feature Importance

At the bottom of this importance graph we see that points per game and field goal percentage have less of an impact to the outcome as do pure defensive stats (Stl + Blk). This only provides more evidence that the old saying "defense wins games" is true.

In the experimentation of choosing the right machine learning algorithm I also considered using boosters like random foresting(rf) and dropouts(dart). Unfortunately dart drastically increased the running time for very little accuracy boosting making it very difficult to test different boosting parameters. The random foresting boosting method simply provided bad boosting, meaning that the accuracy was not quite as good at the other boosters. For this reason I decided to just use a gradient boosting decision tree (gbdt) for all boosting.

Using a bracket log loss visualizer [6] I was able to see the best and worst possible cases for my predictions. I found that the better the best case was the worse the worst case got. Which is why I only decided to drop a few statistics for my final predictions. This may seem a little unconventional, however after using possession to calculate offensive and defensive ratings other stats like personal fouls(PF) fell in importance.

Here I took a small portion of the best possible outcome for my prediction model. In the figure the darker a box is the more favor or higher percentage there is for one team to win over the other. The scores show the percentage out of 100 that each team has to win based on my prediction. As you

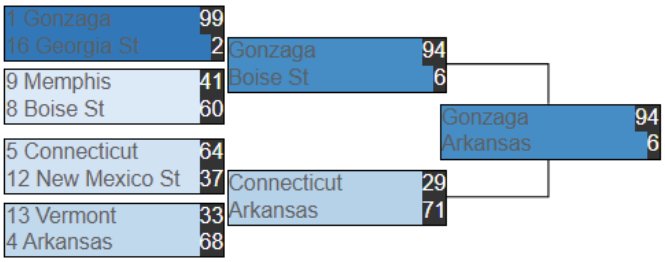


Figure 7. Best Case Bracket Visualization

can see most times the best case is when the lower seeded team wins. However as anyone that has made a bracket before knows, this is rarely the case. The beauty of statistics is that good models can predict some of the upsets purely based on the comparison between the season averages of two teams. Below you can see that although Colgate is a 13 seed, they are predicted to beat every team they face just because they have better offensive and defensive ratings than the teams they played. This topped with some of the variance parameters and similar season averages was enough to predict the upset.

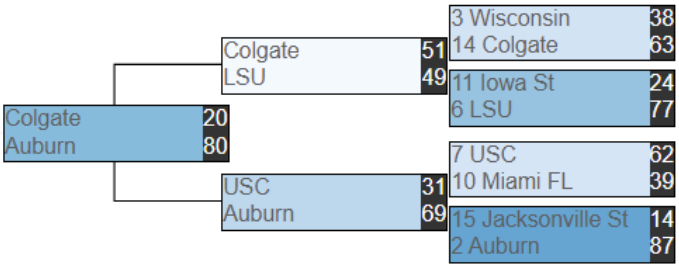


Figure 8. Model Predicting Upset Visualization

I researched what other people had added to their models for perfecting their predictions, and found that there has only been one instance in march madness history where a number 4 seed lost to a number 13 [7]. With this knowledge I adjusted the training data to where if the seed difference between two teams is greater than or equal to 9 (13 seed vs. 4 seed) then I reset the difference to 15 (equal to the distance of 16 seed to 1 seed). This method decreased the log loss of my model.

5. Conclusion

As this project comes to an end, I was able to learn a lot about machine learning and how important certain statistics are in predicting the outcome of basketball games. Below is the average log loss from running my data based on the results of the 2022 march madness tournament (stage 2 of the Kaggle competition). In the figure, green boxes resemble correct predictions whereas red boxes resemble wrong ones. The darkness of the box is determined by how favored a team was to win over the other. For example, a dark green box means that I my prediction of the winning team was correct by a large margin.

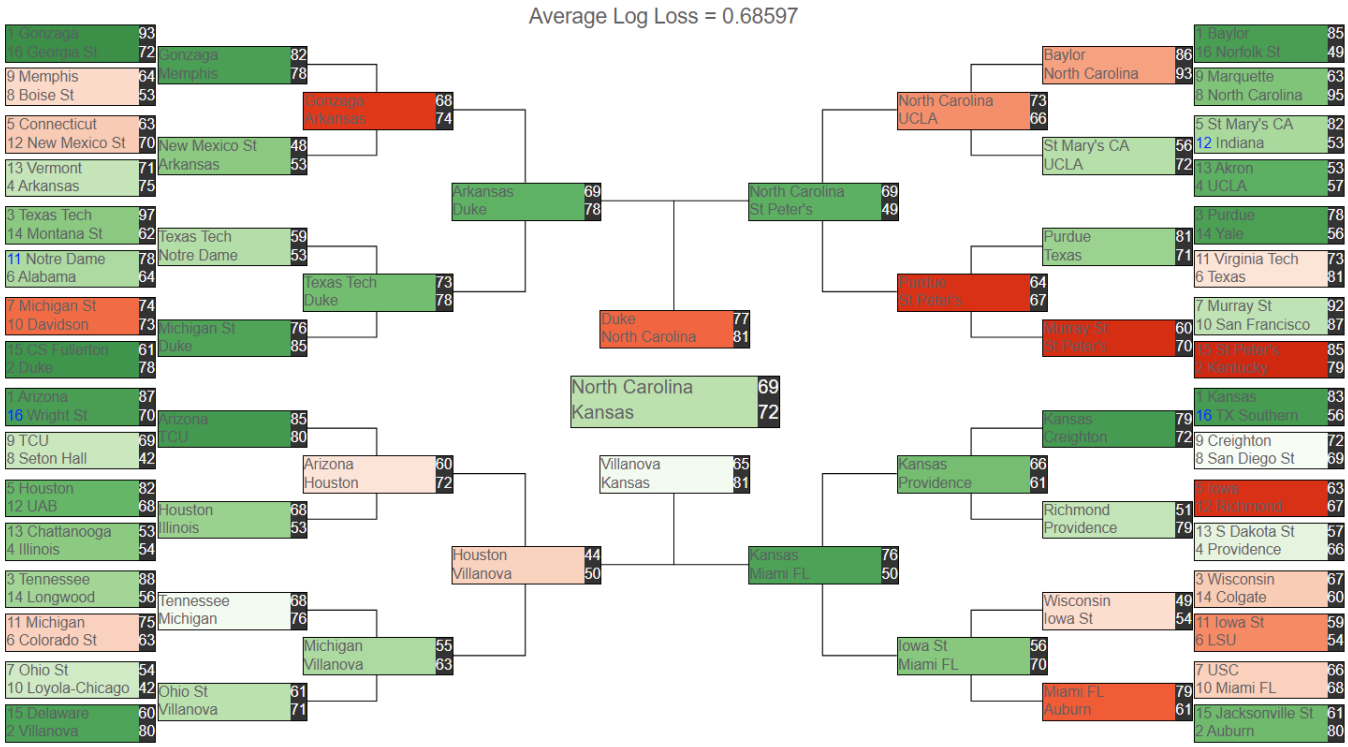


Figure 9. Final 2022 Bracket Prediction

Acknowledgments

I want first off thank my Data Analysis and Mining(B365) Professor, Hasan Kurban. He was very encouraging throughout the semester and anytime I needed help he was able to more than happy to help and encourage me. His teaching methods were different to me, since I had never had a professor that only cared about how much we learned rather than our grade.

I also want to that Kaggle.com for all the data they provided. Without this data this project would not have been possible.

I want personally thank all the Kaggle discussion posts by name as well as any other resource I used to complete this report. This list underneath is what each reference was utilized for during the creation of this report. Please note that some are referenced throughout the report, but I wanted to make sure I gave credit where credit is due.

- Understanding how brackets are made is [1]
- Understanding Exploratory Data Analysis is [8]
- Calculations for Personal Measurements is [2]
- Useful Feature Engineering is [9], [10], [7]
- Useful for modeling is [3], [4]
- Model Tuning is [5]
- Visualizing Prediction is [6]

References

[1] Daniel Wilco — NCAA.com. What is march madness: The ncaa tournament explained, Mar 2022.

[2] Dean Oliver. *Basketball on paper: Rules and tools for performance analysis*. Potomac Books, Inc., 2011.

[3] Soner Yıldırım. Gradient boosted decision trees-explained, Feb 2020.

[4] Lightgbm in python: Complete guide on how to use lightgbm in python, Aug 2021.

[5] Jason Brownlee. Tune learning rate for gradient boosting with xgboost in python, Aug 2020.

[6] Kaggle brackets for march madness.

[7] Amirghazi. Raddarr, Mar 2022.

[8] Malnedellec. mania 2022 - eda, Mar 2022.

[9] Shinkoshi. 2nd place solution, Apr 2022.

[10] Bearcater. March machine learning mania 2022 - men's, Apr 2022.