Predicting College Basketball Games Using Machine Learning

Merkovitz, Sam R
merkovitz.s@husky.neu.edu

Abstract

This paper aims to determine which types/groups of college basketball statistics are predictive of game outcomes. This paper used a plethora of different statistics and statistical categories ranging from raw statistics to calculated efficiency ratings to statistics based on strength of schedule. Using Logistic Regression, Naïve Bayes, Random Forest Classifier, and Neural Network models, different feature sets were evaluated in terms of how well the predict basketball game results. Logistic Regression and Neural Network provided the best results with over 75% accuracy.

Introduction

College basketball is a phenomenon in the United States, especially come March with the annual NCAA March Madness National Tournament, which brings together college basketball's sixty-four best teams every year. Kaggle ran a competition earlier this year calling for people to submit their machine learning algorithm predictions for the 2020 NCAA March Madness Tournament. Unfortunately, due to the COVID-19 outbreak, the 2020 March Madness Tournament was cancelled, but the algorithms and data used are not specific to the tournament and can be used for predicting all college basketball games.

It is important to make a few initial distinctions as these guided many decisions made throughout the process of deciding what and which data to use in the machine learning algorithms. First, college basketball is much different than professional basketball as college basketball teams have players on their team for a maximum of four years and minimum of one year. This means that statistics aggregated over more than any one season are not necessarily predictive of how that may team may perform in the next season, and any season after that. While many college basketball teams perform similarly year to year, it was safer to look at the data irrespective of the actual team it belongs to. By doing this, we can look for patterns in performance to determine how teams will fair in a matchup.

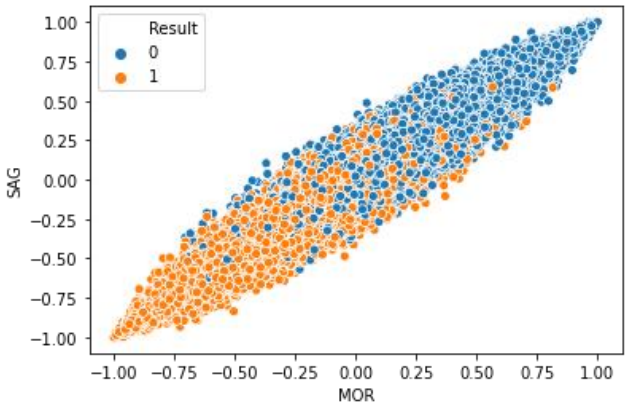
Second, college basketball has a wide range of performance tiers. There are over 300 division 1 basketball teams, recruiting from all over the World. Some college teams have many players that will get drafted into the NBA and some college basketball teams will never see a player from their school get drafted into the NBA. This wide range of competitive ability was an important aspect when predicting the outcomes of games. A quick example will help illustrate this.

Florida State University has a consistently top tier college basketball team. Year to year, they perform about the same due to their recruiting abilities and their college basketball program. Competing in one of the most competitive conferences in all of college basketball, the SEC, their statistics and performance show them to be one of the best. In 2019, they finished the season with a 29-8 record and ranked 11 amongst all college basketball teams. Murray State University also has a D1 basketball team. In 2019, they won the Ohio Valley Conference Championship with a record of 23-9 and ranked at number 37 amongst all college basketball teams. While their records and performance were very similar on paper, their numbers are dictated by the strength of the opponents that they play. Murray State was the only team from their conference to make the national tournament. When Florida State and Murray State played later that year, Florida State won the game 90-62.

This example is designed to illustrate how difficult it is to compare teams with certain metrics. Murray State and Florida State in 2019 were very similar across the board, in fact, Murray State average more points and assists per game that season than Florida State. However, if you asked any college basketball expert who they would've picked to win, they would've overwhelmingly supported Florida State. In order to produce a thorough prediction model, the algorithm would need to be able to understand the difference in competitive levels between teams. If Murray State played the same competition as Florida State, their season's statistics would surely be much different than they were after playing teams in the Ohio Valley Conference. A model looking at just their numbers and ignoring the competition each team played against *may* predict Murray State to win that game. Was there a chance that Murray State could've

beaten Florida State in that matchup? Absolutely, it is impossible to be certain how any given game will turn out but there was a higher probability of Florida State winning then there was of Murray State winning. For future reference, we will call this "the competition problem."

To further illustrate "the competition problem," the graph below shows that picking differences in ranks, while linear, does not provide a clear break between what difference in ranks always result in a win or a loss.



Difference in rankings using the SAG and MOR systems

This graph is meant to show that basketball games do not follow a clear-cut trend. There will always be upsets and games that defy common trends, thus making predictions, and the confidence in those predictions, much more difficult.

Background

The data used in this paper are from the outcomes of all Division 1 basketball games beginning with the 2003 season through the end of last year's 2019 season. For each game, there are the two teams involved, the season and day, the location of the game (denoted as H if it was at the winner's home, away if at the loser's home, and N if it was in a neutral location), and raw statistics (shots attempted, assists, steals, fouls, etc.). Using this dataset, many other statistics could be calculated from there. For example, Field Goal Percentage is simply shots made divided by shots attempted. From this, three separate groups of features were created and tested.

The first group is named "Fundamentals". The features included here are formed from the idea of hustle stats and team chemistry; this aimed to measure how well a team works together. The features for this set were: Assists, Steals, Defensive Rebounds, Blocks, Turnovers, Offensive Efficiency (average points scored per possession), Defensive Efficiency (average points allowed per defensive possession, and Net Efficiency (Offensive Efficiency minus Defensive Efficiency. These statistics were aggregated over each season for each team.

The second group is named "Percentage". The features include here aimed to measure a team's overall efficiency across several stat lines. The features for this set were: Offensive Efficiency, Defensive Efficiency, Net Efficiency, True Shooting Percentage (points per field goal attempt including a weight of 0.44 per free throw attempt), Effective Field Goal Percentage (points per field goal attempt with a weight of 0.5 for three point attempts), Pace (how fast a team moves up and down the court per game), Free Throw Percentage, Three Point Attempt Rate, Rebound Percentage (rebounds by a team divided by all available rebounds), Offensive Rebound Percentage (percent of rebounds when on offense), and Assist Percentage (how many points were scored off an assist). These statistics were also aggregated over each season for each team.

The third group is named "Adjusted Ratings". This feature set is aimed at understanding a team's performance in relation to the strength of their opponents. The features for this set were: Pace, Offensive Efficiency, Defensive Efficiency, Net Efficiency, Effective Field Goal Percentage, Turnover Percentage (how many turnovers per offensive possession), Offensive Rebound Percentage, and Free Throw Percentage. These statistics were a mix of averaging over each season and each game.

Each feature set was analyzed using a Random Forest Regressor, a Naïve Bayes model, and a TensorFlow Neural Network

Related Work

While each feature set was not directly based off of any preexisting models, the ideas behind them were based off basic ideas in the game of basketball. The first feature set, "Fundamentals", was derived from an idea of what makes a *good basketball team*. Teams that play together generally perform well whereas teams that may depend on a high recruit may not have the same performance year to year.

The second feature set, "Percentage", was based on the idea of efficiency. Hypothetically, a more efficient team will produce better outcomes game to game. For example, Effective Field Goal Percentage is a measure of how effectively a team shoots the ball. It weights three-point attempts so that if a team makes more points with less shots, they have a higher effective field goal percentage.

The third feature set, "Adjusted Ratings", is based off the idea of accounting for strength of schedule and is meant to solve "the competition problem." The algorithm to calculate adjusted ratings by strength of schedule is based on work by Alok Pattani and Elissa Lerner, two software engineers at Google Cloud (Pattani and Lerner 2019). The ratings were then transformed and used in my own machine learning models.

Project Description

As a general note, it is important to understand that each game yields two perspectives, the perspective of the winning team and the perspective of the losing team. For one game, there is the winning team, which we will call Team A, and the losing team, which we will call Team B. Each game has statistics for each team. The easy way to understand this is that Team A has their stats in comparison to Team's B stats. Conversely, Team B has their stats in comparison to Team A. The data was engineered to load into the algorithms as Team A's stats and then Team B's stats, assigning a value to the target label column, "Result", with a 1, noting that the first team loaded in had won the game. And then we have another row with Team B's stats followed by Team A's stats and gave a value of 0 to the target label column, noting that the first team loaded in had lost the game.

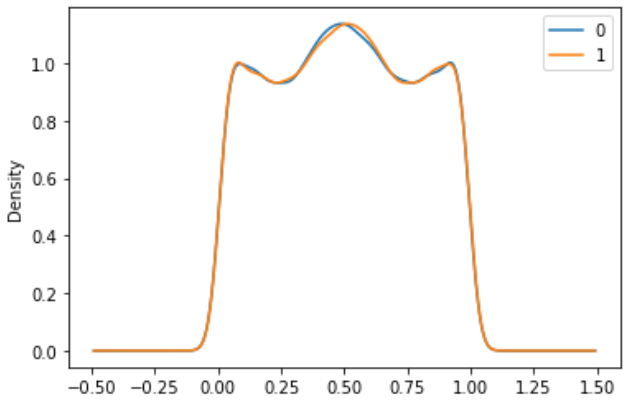
Feature sets either loaded in the statistics for both teams, as mentioned above, or took the difference in stats between the two teams. Those sets still worked on the same premise where the data would be loaded in as Team A's stats minus Team B's stats with 1 as the target label and Team B's stats minus Team A's stats with 0 as the target label.

Another important aspect is how the outputs are evaluated. While accuracy is certainly important when predicting the outcomes of basketball games, there are always going to be games that defy common trends, and thus, 100% accuracy is extremely unlikely. The main metrics for this paper are accuracy in conjunction with LogLoss:

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

LogLoss will measure the average confidence in the probability predictions. Ideally, this log loss is extremely minimal, indicating high confidence in the outputs of the model. High confidence equates to being sure that a team will win or lose. Even then, there are some games that are going to be a toss-up and their outcomes cannot be predicted with high confidence. The goal of this paper was to produce a reasonably accurate model with a negative LogLoss of 0.2 or less. The evaluation of the models in regard to LogLoss will be discussed throughout this paper.

One more important note on evaluation. An initial Logistic Regression was run by evaluating teams using three of the most respected ranking systems: POM, SAG, and MOR. This would establish a baseline as the model would, presumably, decide outcomes based on which team is ranked higher (higher ranks equate to lower numbers). The Logistic Regression model had an accuracy of 74.7% and a LogLoss of 0.506. The assumption made here is that if the higher ranked team was picked every game, you would get 74.7% of the games right. The LogLoss equated to an average prediction probability of 0.5.



Kernel Density Plot for our prediction probabilities using average rank

As can be seen by the kernel density plot above, the predictions are relatively even spread out between 0 and 1 but the majority of games cannot be predicted any better than a coin toss. This is important as we want the models to have strong confidence in the predictions they make. So, a baseline of 74.7% accuracy and 0.506 LogLoss is how we will judge the models.

Each feature set was evaluated with a Logistic Regression, Naïve Bayes, Random Forest and a Neural Network model. The Logistic Regression model was meant to serve as a baseline for each feature set as it is the simplest model. A Random Forest Regressor was run on each set as well to provide insight into the importance of the different features in each set.

Before each feature set was evaluated, the data was randomly shuffled and a train-test split of approximately 80:20 was made, with a train-validation split of 80:20 as well. This equated to 140,000 rows of data for the train set and around 35,000 rows of data for the test set. Each feature set was then scaled on a range of (-1, 1).

Feature Sets

"Fundamentals"

The "Fundamentals" feature set used raw statistical numbers for Assists, Steals, Defensive Rebounds, Blocks, and Turnovers, summed over a season. Again, as the turnover rate for college basketball players is so high, it was better to understand performance based on a season. Offensive, Defensive, and Net Efficiency were averaged over each season and are represented on a scale between 0 and 1. The "Fundamentals" feature set was meant to establish a baseline as it contained the most basic features of all the sets.

The Logistic Regression model for the "Fundamentals" set had an accuracy of 73.1% and a LogLoss of 0.531. This

was fairly accurate. However, this performed worse than our baseline by both metrics.

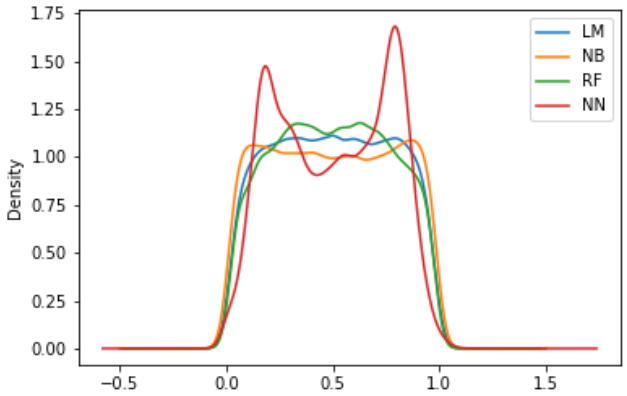
The Naïve Bayes model had an initial accuracy of 71.9% and a LogLoss of 0.683. However, as a Naïve Bayes model is creating its own probability distributions based on the data it receives, it may be diluted by including *all* the features. The Random Forest Regressor output Assists, Defensive Rebounds, and Turnovers as the four most important features in this dataset. Running the Naïve Bayes model on just those features resulted in an accuracy of 70.9% and a LogLoss of 0.56, a decrease in accuracy but an improvement in confidence. However, this still performed worse than our baseline.

The Random Forest model had an accuracy of 72.7% and a LogLoss of 0.537, both still under our baseline.

Finally, a Neural Network was used to evaluate the features. The Neural Network had a feature layer, 4 hidden layers, two Gaussian Dropout layers with values of 0.15, an L1L2 regularization layer with values of 0.01 and 0.01, a learning rate of 0.0001, a batch size of 1000, and a validation split that used 20% of the training data. The model had an accuracy of 72.73% and a LogLoss of 0.5291 when evaluated against the test set. There was a negligible difference between the evaluations of the train and test set.

Ultimately, through these four models, it showed that using fundamental statistics proved less accurate and decisive than picking the higher ranked team for every matchup.

When evaluating the distribution of prediction probabilities, the Logistic Regression, Naïve Bayes and Random Forest Classifier had a similar distribution to the baseline. The Neural Network however provided less probabilities around 50% and had more probabilities around 25% and 75%. The reasoning behind the differences in probability distributions will be presented later in this paper.



Probability distributions for the “Fundamentals” feature set

“Percentages”

The “Percentages” feature set used percentage-based statistics. This feature set was designed to measure a team’s efficiency on the basketball court and was used to standardize statistics across all teams. As some teams play many more games than others, calculating percentage and efficiency would mitigate any noise due to any raw data outliers. For instance, a team that has fewer total games in a season as opposed to a team who played extra games, due to qualifying for their conference championship, would have drastically different raw numbers, but their percentage-based statistics and efficiencies would be directly comparable as although it is aggregated over a season, it is based on their average game performance.

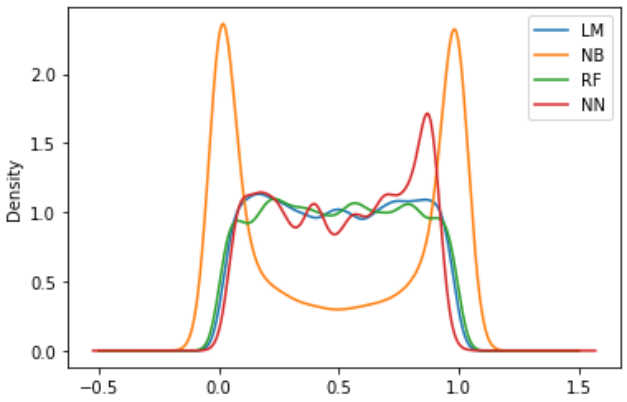
The Logistic Regression model had an accuracy of 74.2% and a LogLoss of 0.515. These were very close to our baseline scores and better than the results of the Logistic Regression model using the “Fundamentals” feature set.

The Naïve Bayes model had an initial accuracy of 72.6% and a LogLoss of 0.836 when using all the features in this set. Looking back at the Random Forest Regressor, Win Percentage has an importance value of 0.4 with the next highest being Net Efficiency with an importance value of 0.06. Running the Naïve Bayes model again using only Win Percentage, the accuracy improved to 74.01% and the LogLoss decreased to 0.519. This is more comparable to our baseline now but still doesn’t perform any better.

The Random Forest Classifier had an accuracy of 72.7% and a LogLoss of 0.534, again performing worse than our baseline but no improvement over the “Fundamentals” feature set.

The Neural Network model was composed of a feature layer, 4 hidden layers, two Gaussian Dropout layers with values of 0.15, an L2 regularization layer with value of 0.01, a learning rate of 0.0001, a batch size of 1000, and a validation split that used 20% of the training data When evaluated against the test set, the model had an accuracy of 72.60% and a LogLoss of 0.5437. There was no dropout between the training set and the test set in terms of the Neural Network’s evaluation.

When evaluating the prediction probability distribution for this feature set against the “Fundamentals” set, there is an interesting difference between the Naïve Bayes model and the other three models. The Naïve Bayes model was providing less predictions between 25% and 75%, most of the predictions were between 0-25% and 75-100%. The Neural Network evaluated differently as well, showing an uneven distribution and having the simple majority of its predictions fall between 75-100%.



Probability distributions for the "Percentages" feature set

"Adjusted Ratings"

The "Adjusted Ratings" feature set also contained feature-based numbers. All of the statistics before being scaled were on a range of 0-120. This feature set is aimed to solve "the competition problem." As discussed before, raw statistics, whether it be in the form of raw numbers or average efficiencies, do not take into account the strength of the competition one team may be playing versus another team's competition. For this feature set, a Ridge Regression model was used to calculate adjusted statistics based on the performance of the team with respect to the strength of their opponents. The algorithm for this was adapted from an article posted by software engineers at Google Cloud (Pattani and Lerner 2019) explaining how they used Ridge Regression to calculate adjusted ratings in relation to the strength of a team's opponents. The algorithm was then redesigned to work for the statistics used in this feature set and then those statistics were applied to the four machine learning models.

In the end, this feature set had two categories, season long averages and adjusted ratings. For each statistics, the team's season long average (ignoring strength of schedule) and adjusted ratings (from the Ridge Regression model) were included for both the team and its opponent and no differences in statistics were included. These features were: Pace, Offensive Efficiency, Defensive Efficiency, Net Efficiency, Effective Field Goal Percentage, Turnover Percentage, Offensive Rebound Percentage, and Free Throw Percentage.

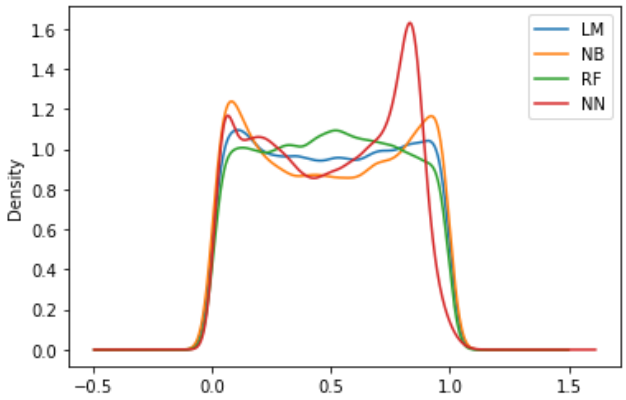
The Logistic Regression model had an accuracy of 74.88% and a LogLoss of 0.495, an improvement over the baseline. While the accuracy stayed relatively close to the baseline, this model was a little more confident in its predictions.

The Naïve Bayes model had an initial accuracy of 73.88% and a LogLoss of 0.866 when using *all* the features in the set. The Random Forest Regressor evaluated the top four important features to be a team's Adjusted and Raw Net Efficiency, and the opponent's Adjusted and Raw Net Efficiency. Running a Naïve Bayes model using these top four features resulted in an improved accuracy of 74.17% and a LogLoss of 0.515, this is very close to the baseline but is still a little less confident than the Logistic Regression Model.

The Random Forest Classifier had an accuracy of 73.8% and a LogLoss of 0.512, close to our baseline but ultimately not as strong as the Logistic Regression model as well.

The Neural Network model was composed of a feature layer, 5 hidden layers, two Gaussian Dropouts with a value of 0.15, one Dropout layer with a value of 0.45, an L1L2 Regularization layer with values of 0.001 and 0.001, a learning rate of 0.0001, a batch size of 2500, and a validation split of 20% of the training data. When evaluated against the test set, this Neural Network had an accuracy of 74.95% and a LogLoss of 0.5045, all around improvements over the baseline and comparable to the Logistic Regression model. It's also interesting to note that this Neural Network took much longer to train and converge than the other Neural Network models. The data was a bit noisier and the learning curve was fairly stochastic.

Looking at the probability distributions, there is nothing too out of the ordinary. However, the Neural Network here also had a higher density of prediction probabilities in the 75-100% than it did in the 0-25%.



Probability distribution for the "Adjusted Ratings" feature set

Combining All Feature Sets

A final set of features was created that combined the four most important features from each set. Again, each of the four machine learning models was run on each set. The final set of features was composed as follows:

- "Fundamentals"
    - Assists
    - Defensive Rebounds
    - Turnovers
    - Free Throw Attempts

- "Percentage"
    - Winning Percentage
    - Net Efficiency
    - Assist Percentage
    - Free Throw Percentage
- "Adjusted Ratings"
    - Opponent's Raw Net Efficiency
    - Team's Raw Net Efficiency
    - Team's Adjusted Net Efficiency
    - Opponent's Adjusted Net Efficiency
- Rankings
    - POM
    - MOR
    - SAG

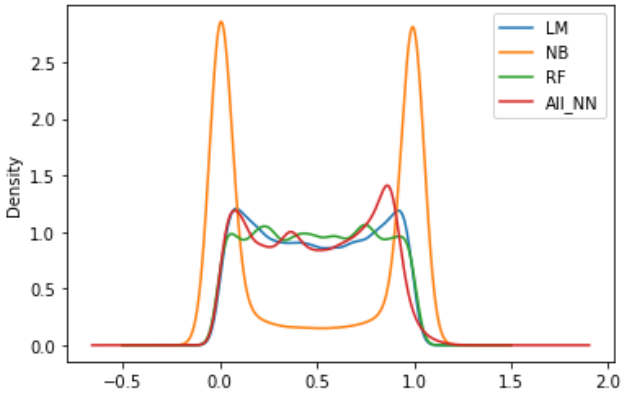
The Logistic Regression model for this new combined feature set had an accuracy of 76.7% and a LogLoss of 0.476, all improvements over the baseline scores.

The Naïve Bayes model for this new combined feature set had an accuracy of 75.2% and a LogLoss of 1.343. The accuracy was impressive as it was above the baseline and higher than any other Naïve Bayes model but the LogLoss was extremely high, almost double that of any other LogLoss. As mentioned before, this could be due to the number of features, determining a smaller subset of features could easily improve the LogLoss. A high accuracy and high LogLoss indicates that the Naïve Bayes model may have been very wrong with some of its predictions, indicating it may have encountered upsets in the test set.

The Random Forest Classifier for this new combined feature set had an accuracy of 75.7% and a LogLoss of 0.486. An improved accuracy across all Random Forest Classifiers and a lower LogLoss as well.

The Neural Network model for this new combined feature set had an accuracy of 76.51% and a LogLoss of 0.4864. This was the best Neural Network Model by all metrics, and it performed better than baseline.

The probability distribution is relatively evenly distributed for all models except the Naïve Bayes model. The Naïve Bayes model was more likely to assign a probability extremely close to 0% or extremely close to 100% with much fewer assignments in the middle. Essentially, this Naïve Bayes model did accomplish the original goal, reasonably accurate with high confidence in its probability predictions. However, it also was wrong with some of its predictions, resulting a higher LogLoss.

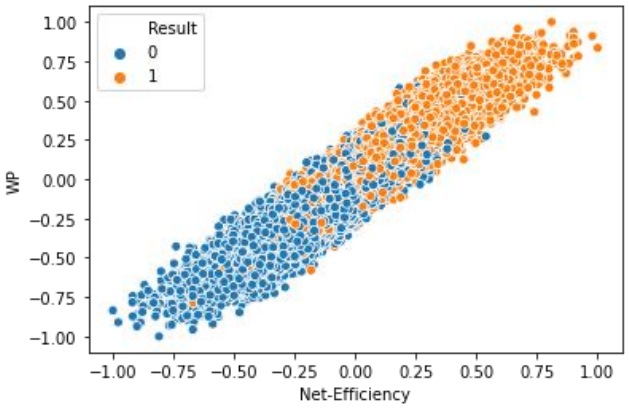


Probability distribution for the models using combined feature sets

Results Evaluation and Discussion

As mentioned before, there are limitations when predicting basketball games. It is impossible to predict games with 100% accuracy as there will always be upsets or games that defy common trends. It is also impossible to be extremely confident in *all* your predictions as there will inherently be matchups where teams have extremely similar statistics. These limitations proved true throughout the evaluation of the feature sets and models in this paper. A LogLoss of 0.5 indicates that the prediction probabilities for all matchups fall mostly between 25% and 75%. In fact, the distribution of probabilities appeared as a normal distribution for most of the prediction probabilities, regardless of the model.

Each probability distribution is also determined by the distribution of the games in test set. As each train and test set was randomly sampled, some test sets may have more games resulting in a win than a loss or vice versa. This would explain the density plots for Neural Networks from the "Percentages" and "Adjusted Ratings" probability distributions.



There was an interesting difference between the generative and discriminative models that were used. In all cases, the discriminative classifiers (Logistic Regression, Neural Network) outperformed the generative classifier (Naïve Bayes). This is generally unsurprising as a Naïve Bayes model is reliant on the distribution of the games it receives. In fact, reshuffling the training and test sets often resulted in slightly different Naïve Bayes accuracy and LogLoss results. It was also interesting to see that the Naïve Bayes often performed much worse when using complete feature sets as opposed to using a refined set of features as defined by the Random Forest Regressor.

Showing results of games based on differences in Win Percentage and Net Efficiency

In terms of the original goal of this paper, a LogLoss of 0.2 or less was not reached by any model. As the results of the models showed, this is, perhaps, an unlikely metric for any model to reach. No matter the group of statistics, there is a limit to how confident the predictions can be. To explore why this is, we can look at Net Efficiency and Winning Percentage from the "Percentages" set (figure above). There are some obvious boundaries; for games where there was a large difference in net efficiency and winning percentage, the results were easy to predict. However, in the bottom left of the graph, there are a few games that went against the trend and therefore, decrease the prediction probability for future games with similar statistics. In the middle of the graph, it is clear that games where teams have similar winning percentage and net efficiency scores are much harder to predict. Thus, for the few games that defy trends or are a toss-up, our models LogLoss is subject to increase. The Logistic Regression and Neural Network models do a good job of avoiding overly inaccurate predictions, resulting in an average LogLoss around 0.5. The Naïve Bayes model is more focused on the probability distribution of its input, and thus is adept to being overly confident and inaccurate in some of its predictions.

Conclusion

Through evaluating these different feature sets and models, it can be decided that a discriminative model is the best method for predicting basketball. It is also important to measure these models in LogLoss and accuracy in order to understand the true power of the model. This project also showed that confidence is not easy to have when predicting basketball matchups. The data available is filled with games that have mostly even matchups, this limits models on how confident they can be in assigning probabilities because of the noise in the data. This is unavoidable however as this is the reality of college basketball and sports in general.

This project also shows that no one statistic in particular, or group of statistics, is informative of why teams win basketball games. This project *used paper statistics*, or statistics that come from results of events on the basketball court. For future updates to these models, I would like include statistics such as experience, coaching metrics, fan sections, and other data points that are not directly derived from the 10 players on the court. Including these statistics will likely refine the model to understand the importance of experience on the court, strength of schedule, and school spirit.

For those who wish to take a task like this on, the key is with feature and data engineering, as well as using discriminative models. A Logistic Regression model proved to be just as powerful as a Neural Network for this project; understanding that will save time and complexity when determining the best machine learning models for the task.

References

Pattani, Alok, and Elissa Lerner, 2019, *Fitting It in: Adjusting Team Metrics for Schedule Strength. Medium. Analyzing NCAA Basketball with GCP*. https://medium.com/analyzing-ncaa-college-basketball-with-gcp/fitting-it-in-adjusting-team-metrics-for-schedule-strength-4e8239be0530, accessed April 19, 2020