Method to Madness: Modeling NCAA Men's Basketball

Mehmet Cagri Kaymak
Michigan State University
kaymakme@msu.edu

Connor Masini
Michigan State University
masinico@msu.edu

Ian Whalen
Michigan State University
whalenia@msu.edu

1. Introduction

March Madness refers to the intense National Collegiate Athletic Association (NCAA) Men's Basketball (NCAAB) tournament that is held yearly. The tournament commences in mid-March and ends in early April. Sixty-eight teams are selected to compete for the national title based on their performance during a particular year. In 2017, the American Gaming Association estimated that Americans wagered $10.4 billion dollars on March Madness [2], with no changes foreseen in 2018 [3]. Beyond monetary gain, March Madness presents an interesting predictive modeling challenge: given the data of previous conference and non-conference results between all NCAAB teams, how well can one predict the outcome of the current year's tournament?

As noted in [27], college basketball presents a much different challenge than professional basketball and other sports in general. Players leave teams quite often due to professional offers or the four year limit on NCAA competition. This causes any one particular team's performance to be variable from year to year. There are dozens of statistics relating to a team's overall success. Metrics like offensive and defensive efficiency efficiency have been considered most important in the past [18]. Here we will show our results on the importance of particular metrics, predictive modeling results, and explore the structure of NCAAB.

2. Previous Work

Limited work has been done to specifically predict the outcomes of NCAAB games. However, there is a fair bit of work among other professional sports that we draw on in this study.

2.1. Interseasonal Variability

At the center of NCAAB data variability is that the NCAA limits all collegiate athletes to four years of participation in a particular sport. Therefore, at a maximum, a player participating in the 2017-2018 season could have only been on a team since 2013 (ignoring "redshirt" seasons). Leung and Joseph [20] also point out the fact that this phenomenon is compounded when comparing teams, since there will be even more pairwise differences when considering data between two teams that may have players that have never competed against each other mixed with 4 year veterans. We also note that Shi et al. [27] mention this phenomenon. Beyond bringing up the potential biases this may introduce, little has been attempted to measure this *interseasonal variability*.

2.2. Feature Engineering

Pythagorean expectation is a metric that uses offensive and defensive efficiencies to predict what percentage of its games a team will win in a given season. It can also be used as a metric of a team's strength and can be used as a predictor of which team will win a given game. Our Pythagorean expectation is calculated as follows:

$$\text{Pythagorean Expectation} = \frac{1}{1 + (\frac{\text{def\_eff}}{\text{off\_eff}})^8}$$

The exponent of 8 was chosen based on findings in [23] that showed using 8 as the exponent provides the best accuracy when predicting games.

In this study, we also employ rankings percentage index (RPI) to estimate a given team's strength. RPI takes into account how difficult a team's schedule was in order to more accurately assess a team's strength. RPI is calculated as a weighted average of a team's weighted win percentage, its opponent's win percentage, and its opponents' opponents win percentage. The teams weighted win percentage awards 1.4 wins for an away win, 0.6 wins for a home win, and 1 win for a neutral site win, and divides the total number of wins by the number of games played. This is done to value road wins more than home wins, as statistically, road games are 1.4 times harder to win than home games. RPI can then be calculated as

$$\text{RPI} = \frac{\text{win\_rate}}{4} + \frac{\text{opps\_win\_rate}}{2} + \frac{\text{opps\_opps\_win\_rate}}{4}$$

Glicko-2 [13] is Mark Glickman's extension of Arpad Elo's well-known chess rating, Elo[8]. The philosophy of rating systems like Elo or Glicko is to try to estimate the true skill value of a player or team. When two actors compete, their rankings are adjusted based on the outcome and the differential of their ratings. If a high rated team beats a low ranked team—an expected outcome—the ratings of the two teams ranking's do not change much. However, if a high rated team loses to a low rated team—an unexpected outcome—both rankings will be adjusted more dramatically. This description fits closely to Elo's original system [8]. Both Glicko and Elo assume each agent's skill follows a normal distribution, however Glicko takes into account the variance of an agent's rating during an update. Glicko has been used to as an objective measure of "skill" across a wide variety of applications like sports [5], psychological studies [19], chess [12] (its original purpose), and multiplayer online games [11]. From here on we refer to the Glicko-2 metric interchangeably as Glicko, without its version qualifier; with the fact in mind that only Glicko-2 was considered.

2.3. Predictive Models

Tree methods lend themselves well to predicting sports outcomes because they allow for feature importance extraction as well as remove the necessity for data normalization. Zimmermann et al. [27] showed success in using a decision tree predict the outcome of NCAAB games. Here we will extend those results by applying the random forest (RF) algorithm. RF [4, 16] uses an ensemble of decision trees to classify data. With some extra computation, feature importances can also be calculated with this method. Breiman presents this feature importance calculation in [4]. In summary, after training a RF, a certain feature is randomly permuted and its out-of-bag error is calculated across the forest and averaged. This allows direct observation of feature importance, providing insight into the prediction task. Here, we apply the Scikit-learn implementation of RF, which stores this calculation automatically [22].

Adaptive boosting, or AdaBoost, is a successful extension to the boosting paradigm presented in [9]. Freund and Schapire [10] explain that *boosting* is the method of combining "moderately predictive" decision rules into an effective classifier. Adaboost extends this idea by learning weights for each "weak" classifier. AdaBoost has shown success in many areas like finance [25, 1] and bioinformatics [21]. Intuitively, boosting suits sports prediction. Freund and Schapire's classical horse race gambling example [9, 10] is directly applicable in NCAAB as well. A gambler may use certain factors about a team to make a judgment on a wager, AdaBoost attempts to capture and formalize this process. We again apply a Scikit-learn [22] implementation using a decision tree of depth one as our weak classifier.

The support vector machine (SVM) classifier learns a hyperplane that maximizes the margin between the decision boundary and the closest points to the decision boundary (called support vectors). Optimizing a SVM is equivalent to solving a quadratic programming problem whose formulation is strictly convex. Hence the optimum for SVM is unique and global [6]. With the help of kernel tricks (linear, polynomial, RBF, etc.) SVM can be applied to problems with nonlinear decision boundaries as well. It has been shown that SVM yields impressive results across many domains. Shin and Lee's work has showed that SVM could outperform BPN (Back Propagation Network) to the problem of corporate bankruptcy prediction [24]. Also, SVM has been used in text classification [17], face recognition [14], bioinformatic tasks [7] and many other domains. Since it has been used to predict the results of NBA matches [15] and soccer matches [26], we have included SVM as one of our predictors as well. The SVM classifier designed by the Scikit-learn library [22] is applied here.

3. Problem Statement

Our high level problem is to predict the outcome of games within the NCAAB tournament. This lends itself to many experiments. In addition to presenting results of predicting the NCAA tournament, we will also test the following hypotheses about NCAAB:

- Data from too far in the past becomes overly variable due the factors discussed in 2.1. Prediction and feature engineering on data collected from too far in the past will decline in predictive quality.

- The Glicko-2 metric alone is an effective predictor of the outcome of the NCAAB tournament. Other features will increase predictive ability, but only marginally.

- Serial prediction, or predicting the tournament games in order, will drastically reduce accuracy. This is more or less self evident, since predictions made incorrectly early on in the tournament will have long lasting effects.

4. Approach

4.1. Feature Engineering

Because NCAA basketball is such a popular sport, there is a multitude of data available for us to use. To create our feature set, we began by computing the averages for all 14 statistical categories the NCAA records. A list of all of these basic statistics and their abbreviations in our data matrix can be found in Table 1.

In addition to using basic season averages, we created some more advanced metrics for team strength. The most basic of these metrics are offensive and defensive efficiencies. Offensive efficiency is the measure of how many

Statistic	Abbreviation
Field Goals Attempted	fga
Field Goals Made	fgm
3-Point Field Goals Attempted	fga3
3-Point Field Goals Made	fgm3
Free Throws Attempted	fta
Free Throws Made	ftm
Assists	ast
Blocks	blk
Offensive Rebounds	or
Defensive Rebounds	dr
Steals	stl
Turnovers	to
Personal Fouls	pf
Points	points

Table 1: List of the basic features derived from the season averages of NCAA statistics.

points a team scores per possession, while defensive efficiency is the measure of how points are scored on that team per possession of its opponent. These are easily calculable as points divided by possessions; however, possessions are not a statistic that the NCAA records. Fortunately, since a possession can only end when a team scores, shoots free throws, or turns the ball over, we can accurately approximate the number of possessions in a game.

$$\text{possessions} = \text{fga} - \text{or} + \text{to} + (0.4 \times \text{fta})$$

This formula works because all field goal attempts that do not get rebounded by the offensive team result in a change of possession, all turnovers result in a change of possession, and on average, 40% of free throw attempts result in a change of possession [18].

In addition to the basic statistics and efficiency ratings, we also calculated Glicko, RPI, and Pythagorean Expectation values for each team throughout the season in the manner detailed previously in the paper. In addition, we add win percentage, weighted win percentage, opponents' win percentage, and opponents' opponents win percentage as features since they needed to be calculated in order to determine team's RPI ratings and contain valuable information about teams success relative to how difficult their schedule was. A list of all of the advanced statistics we calculated can be found in Table 2.

We constructed five separate data matrices while trying to optimize our models. Each model began calculating statistics starting from a different year. Since college basketball players are only allowed to play for four years, we limited our calculation of statistics to fours total. This gave us four matrices, each with slightly different stats for a given year. After evaluating the performance of features, we found that glicko was the most important feature. To investigate this further, we made a fifth data matrix that calculated glicko from 1985 (The start of our dataset) to test against the other datasets.

Statistic	Abbreviation
Possessions	poss
Pythagorean Expectation	pyth_exp
Win Percentage	win_rate
Weighted Win Percentage	weighted_win_rate
Oppt Win Percentage	oppt_win_rate
Oppt Oppt Win Percentage	oppt_oppt_win_rate
Glicko	glicko

Table 2: List of the advanced features derived from the basic season averages of NCAA statistics.

4.2. Glicko-2 Experiments

Special attention was given to the Glicko rating for NCAAB since it is quite a popular metric for gaming predictions. We performed three experiments to observe the properties of Glicko in NCAAB: (1) we gathered Glicko ratings from varying start times in the past to observe the distribution of average Glicko rating, (2) see how well these differently gathered Glicko scores can predict the 2017 NCAAB tournament, and (3) see how well correlated Glicko is with the amount of money the top universities spent on men's basketball in the 2016-17 season.

4.3. PCA Experiment

To gain more insight about our feature set, we applied principal component analysis (PCA) to see which features contribute most to the variance. Because the features are on different scales, we normalized the data with using the Min-Max normalizer. We did not prefer standardize using a normal distribution because it assumes the features are normally distributed. For brevity, we present this experiment for only the 4 season data for PCA plots. As it is understood from Figure 1, the first 15 principal components are able to capture 89.4% of the overall variance from 76 total features. The feature projection into $\mathbb{R}^2$ is shown in 2. The data is highly intermingled in two dimensions, however a the plot still provides a suggestion of a decision boundary between the two classes.

4.4. Training Predictors

For training all of our predictors, we make heavy use of cross-validated grid search to choose hyperparameters, as implemented by Scikit-learn [22]. For all training procedures, we use 5-fold cross-validation during the search. This means that whichever set of hyperparameters achieved the highest average validation accuracy was chosen and we

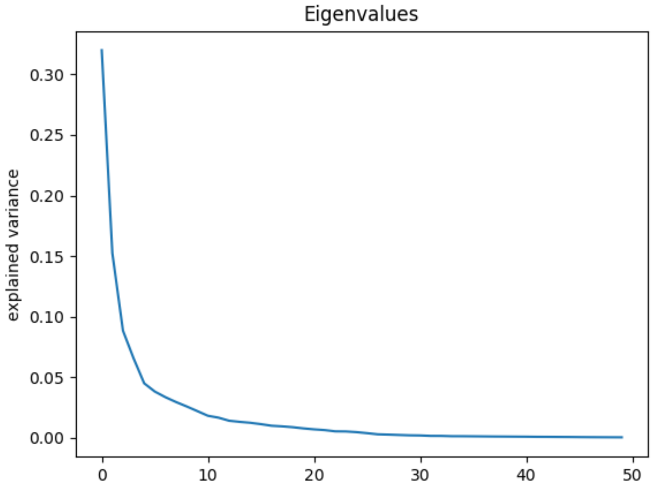


Figure 1: Plot of variance explained by the first 50 eigenvalues of the covariance matrix.

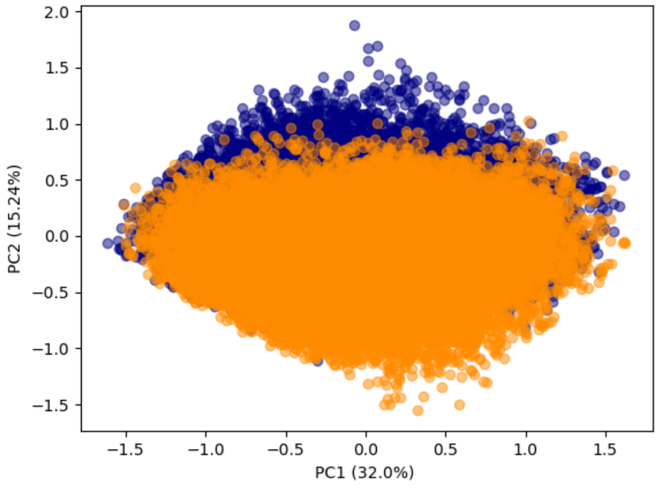


Figure 2: Projection of the data into $\mathbb{R}^2$ using the first two principal components.

report its testing accuracy on data that was not used during grid search. In most cases the final range reported was not the only range that was searched over, just the range that found an optimal solution within the bounds of the search. In all cases, the testing data is the NCAAB tournament for a particular season.

5. Data

The data used for this study was gathered as part of a Kaggle competition hosted at `https://www.kaggle.com/c/mens-machine-learning-competition-2018`. In addition, this dataset contains detailed game information for games played during or after the 2003 season. This data contains game information for both the regular season and the post-season tournament dating back to 1985. This includes the date and location of each game, in addition to the made and attempted shots for all three kinds of baskets, number of offensive and defensive rebounds, number of assists and turnovers, number of blocks, and total score for each of the two teams.

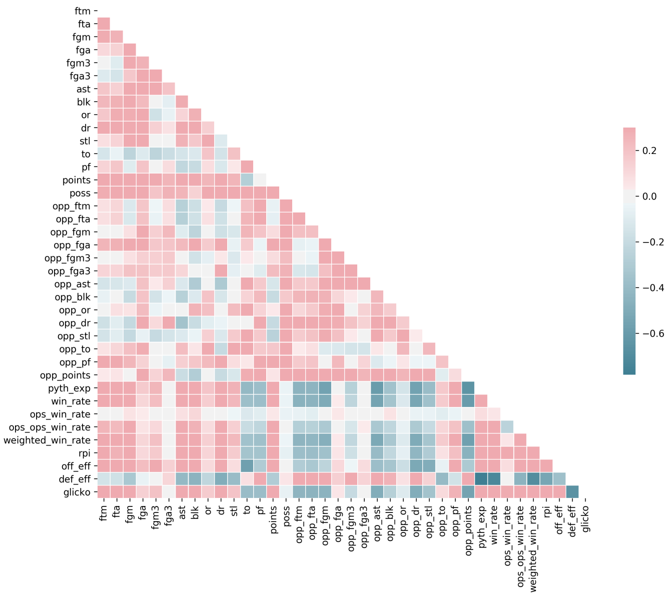


Figure 3: Correlation plot of the engineered features.

To compile the features for each given year, we iterated over each game one by one, calculating and storing statistics for each team. For a given game, we use these stored statistics to create a row with all the features of the team with the lower id number, followed by the same set of features for the team with the higher team ID number. We used the rows we constructed for the regular season games to train our models. This effectively turned each game into a binary classification problem, predicting whether the team with the lower ID number will win or not.

6. Experimental Analysis

6.1. Glicko-2 Results

Recall that both Glicko, and its ancestor Elo, assume that a team's true skill follows a normal distribution [12, 8]. If we average the Glicko ratings of the entire NCAAB, we get some insight into how skill is distributed as a whole. For team j, we have that the team's Glicko ranking follows $\mathcal{N}(\mu_j, \sigma_j^2)$. We then observe that the average Glicko follows the distribution, for n teams,

$$\mathcal{N}(\frac{\sum_{i=1}^{n}\mu_i}{n}, \frac{\sum_{i=1}^{n}\sigma_i^2}{n}).$$

We use this distribution to observe the effect of gathering the Glicko rating data from games further in the past. The average distributions over time are shown in Figure 4. A curve's label refers to the year Glicko began to be gathered.

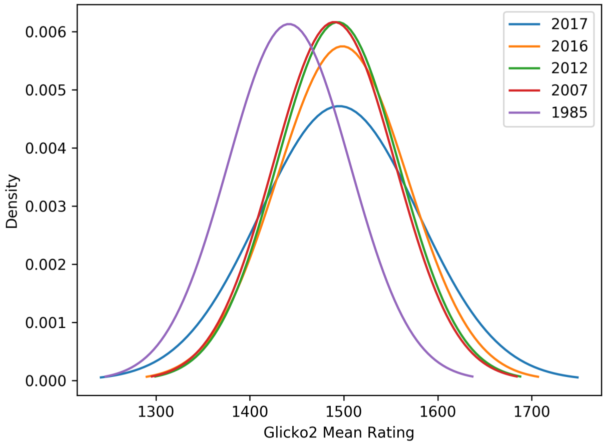


Figure 4: Mean NCAAB Glicko distribution shown over varying metric calculation start years. Each curve's label refers to the year Glicko began to be gathered.

Above we presented our average distribution experiment, to gain further insight into the efficacy of the Glicko rating, we evaluated its prediction accuracy on games in the 2017 NCAAB tournament. More precisely, we predict a team a to beat team b if the Glicko of team a is higher than team b. Similarly to our first Glicko experiment, we then gather the Glicko ratings from the regular seasons starting in 2017, 2016, and so on until the beginning of our dataset in 1985. See Figure 5 for the plot of these accuracies as a function of the start year. Glicko prediction accuracy was at a maximum at 74.63% when 2016 was used a start year. From this peak, as training year decreases, the prediction accuracy becomes variable and eventually flattens out at 71.64% accuracy in 2001.

To conclude our Glicko experiments, we consider the correlation of Glicko with men's basketball spending. Using the Glicko score calculated from 1985 to 2017 on all regular season data, we took the top-25 teams and obtained their men's basketball 2016-17 season total expenses. These top-25 teams were (in order): Kansas, Villanova, Kentucky, Arizona, North Carolina, Gonzaga, Duke, Oregon, Louisville, Virginia, West Virginia, Baylor, Southern Methodist, Purdue, Notre Dame, Iowa State, Wisconsin, Wichita State, University of California-Los Angeles, Miami, Maryland, Cincinnati, Butler, Michigan, and Michigan State. The spending statistics were retrieved from the U.S. Department of Education's Equity in Athletics Data Analysis website: `https://ope.ed.gov/athletics/`. Before we performed a correlation analysis, we removed spending outliers using the Tukey fence method. In other words, those schools that spent outside of 1.5 of the interquartile range were removed from the experiment. North Carolina, Kentucky, Duke, Louisville, and Michigan were removed. See Figure 6 for the scatter plot and trend line. A correlation value of $r = 0.6058$ is obtained, indicating a fairly strong trend between all-time Glicko ranking and how much a University spent on Men's Basketball in the 2016-17 season.

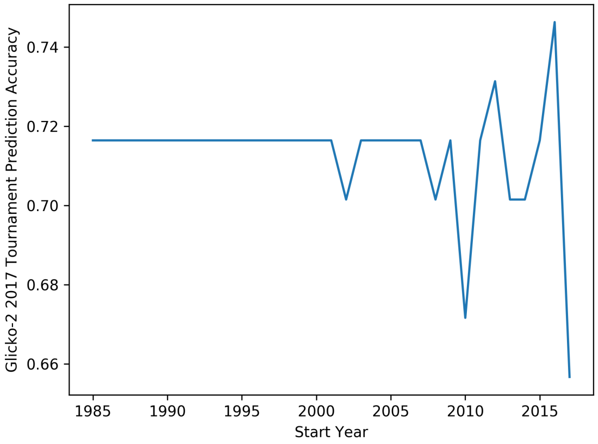


Figure 5: Glicko prediction accuracy on the 2017 NCAAB tournament plotted as a function of the year Glicko began being gathered. The highest accuracy value was observed at 74.63% using 2016 as the start year.

6.2. Predictor Results

Random forest was trained on all datasets, each searching for different optimal hyperparameter sets. The number of trees in the forest was the only hyperparameter tuned in the RF grid search. The range for the number of estimators was 90 to 200, with a step size of 10, i.e. 90, 100, 110, etc. The results of the best hyperparameters found and their testing error for each particular dataset is presented in Table 6. There is little variation among testing errors, however the RF trained on the 2 season data showed the highest testing accuracy, with the 4 season data showing the lowest. For each dataset we also extracted the feature importance for each engineered feature. Bar graphs of which feature was most important are presented in 7. It can be seen that as the number of seasons increases, Glicko becomes a more

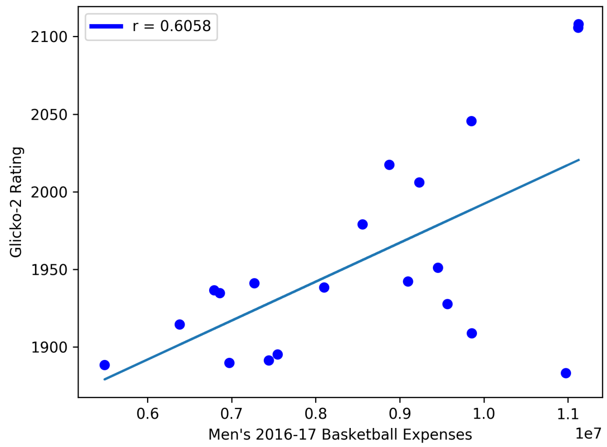


Figure 6: Scatter plot of each university's 2016-17 men's basketball expenses and their all-time glicko rating. Here, outliers in spending were removed before calculating the line of best fit. Note that the horizontal axis is labeled in 10s of millions of U.S. dollars.

important feature. However, Glicko is not the most important feature when it is gathered from the beginning of our dataset, as shown in Figure 7e.

Dataset	Num. Trees	Test Acc.
1 Season	160	0.7023
2 Seasons	170	0.7099
3 Seasons	160	0.7081
4 Seasons	180	0.6933
1 Seasons, All-time Glicko	170	0.7044

Table 3: Cross-validation results for number of trees in the RF, which dataset they were found on, and their testing accuracies. All-time Glicko refers to the Glicko ranking being gathered since 1985 at the beginning of our dataset. This is in contrast to the other datasets that gathered their statistics only for as many seasons that are given.

AdaBoost was trained similarly to our RF classifiers. Again, for each dataset we did a cross-validated search across a list of hyperparameters. Only the number of decision stumps used in the Adaboost algorithm was tuned. The range for the number of stumps ranged from 40 to 120, again with a step size of 10. The results of this hyperparameter tuning and the testing accuracy of the subsequent classifiers is presented in Table 4. Similarly to our RF results, we present the feature importance calculations found by AdaBoost in Figure 9. AdaBoost shows the same distribution for Glicko over the seasons as RF, becoming more important in the 4 season dataset, but not showing as the most important feature in the all-time Glicko dataset.

Dataset	Num. Stumps	Test Acc.
1 Season	120	0.7044
2 Seasons	70	0.7077
3 Seasons	110	0.7151
4 Seasons	40	0.7085
1 Seasons, All-time Glicko	90	0.7054

Table 4: Cross-validated results for the number of decision stumps used in AdaBoost, which dataset they were found on, and with their testing accuracies.

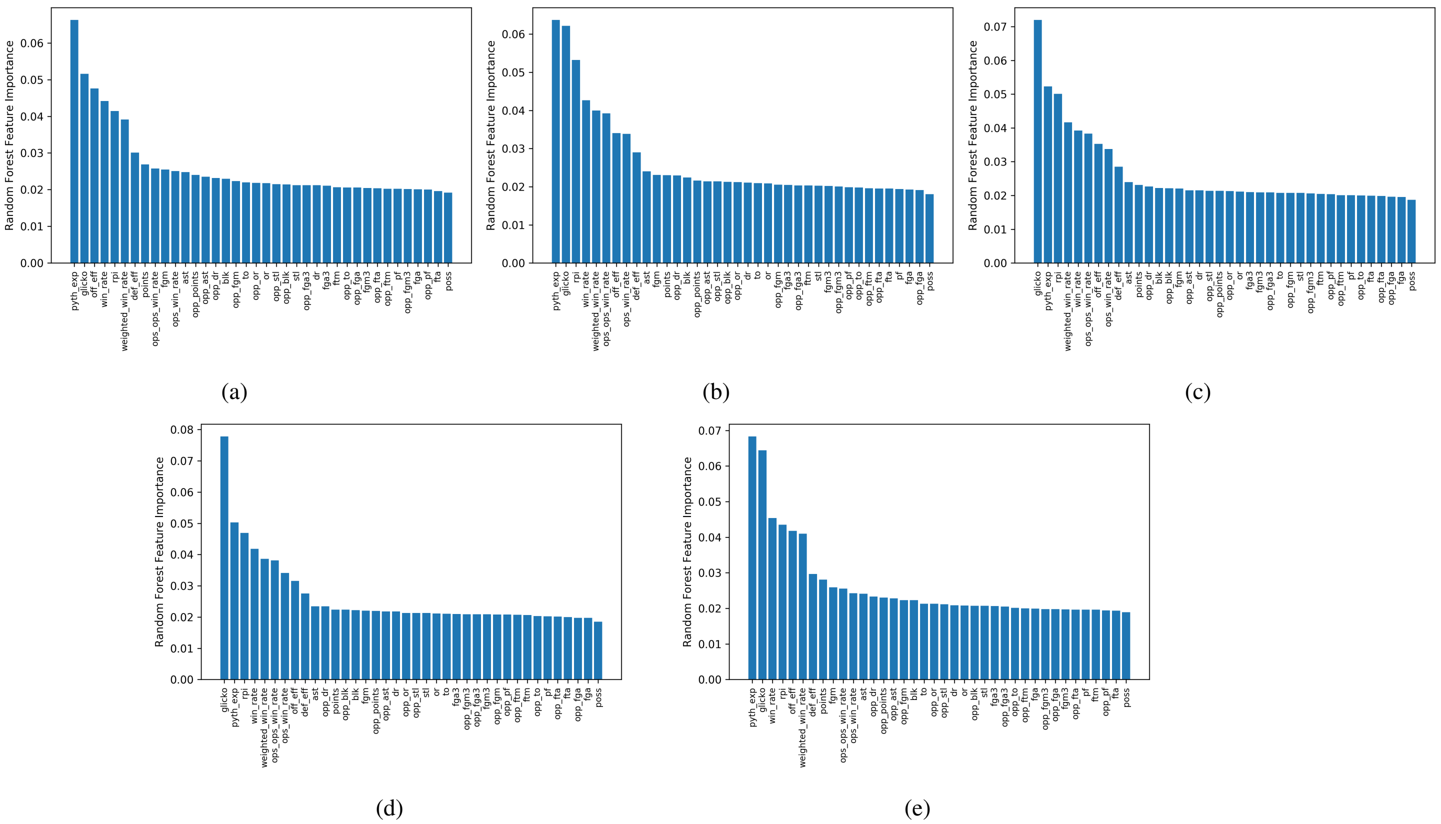
Table 5 shows the results of the ensemble methods when predicting the tournament in serial. By these, results, we show that AdaBoost when trained on the previous 3 regular seasons of data, out performs all other models. Meanwhile, RF performs poorly across the board, save for the model trained on the all-time Glicko dataset.

Dataset	Tournament Score
1 Season AdaBoost	120
2 Seasons AdaBoost	52
3 Seasons AdaBoost	128
4 Seasons AdaBoost	78
1 Seasons, ATG AdaBoost	90
1 Season RF	65
2 Seasons RF	66
3 Seasons RF	67
4 Seasons RF	58
1 Seasons, ATG RF	96

Table 5: The dataset column corresponds to which dataset a particular ensemble model was trained on, with ATG referring to the all-time Glicko dataset.

SVM was trained on all datasets and to tune the hyperparameters, cross-validation was applied. RBF kernel with different gamma and C values and linear kernel with different C values were compared to find the optimal set. We could not apply tuning separately on all the datasets because SVM requires high computational power and combined datasets (combination of all of the years) have more than 80000 rows. Due to computational constraints, we tuned hyperparameters on a subset of the dataset and used then used the best found parameters to train the models. After experimenting on various years and seasons, surprisingly linear kernel performed better than RBF kernel. For the other experiments, we used linear kernel with C = 10 and we applied Min-Max normalizer.

To understand which features are more effective for SVM classifier, we plotted the coefficients of the features. Since the features with higher coefficients affect the final



(a)

(b)

(c)

(d)

(e)

Figure 7: Random forest feature importances for varying statistic calculation methods. These plots show the calculated feature importance measures for statistics gathered for only the current season (a), previous two seasons (b), previous three seasons (c), and previous four seasons (d). Plot (e) shows the feature importance measures for the a dataset where all features are calculated for only the current season, excluding Glicko, which was gathered since the beginning of our dataset in 1985.

Dataset	Test Acc.
1 Seasons	0.7156
2 Seasons	0.7189
3 Seasons	0.7203
4 Seasons	0.7046
1 Seasons, All-time Glicko	0.7177

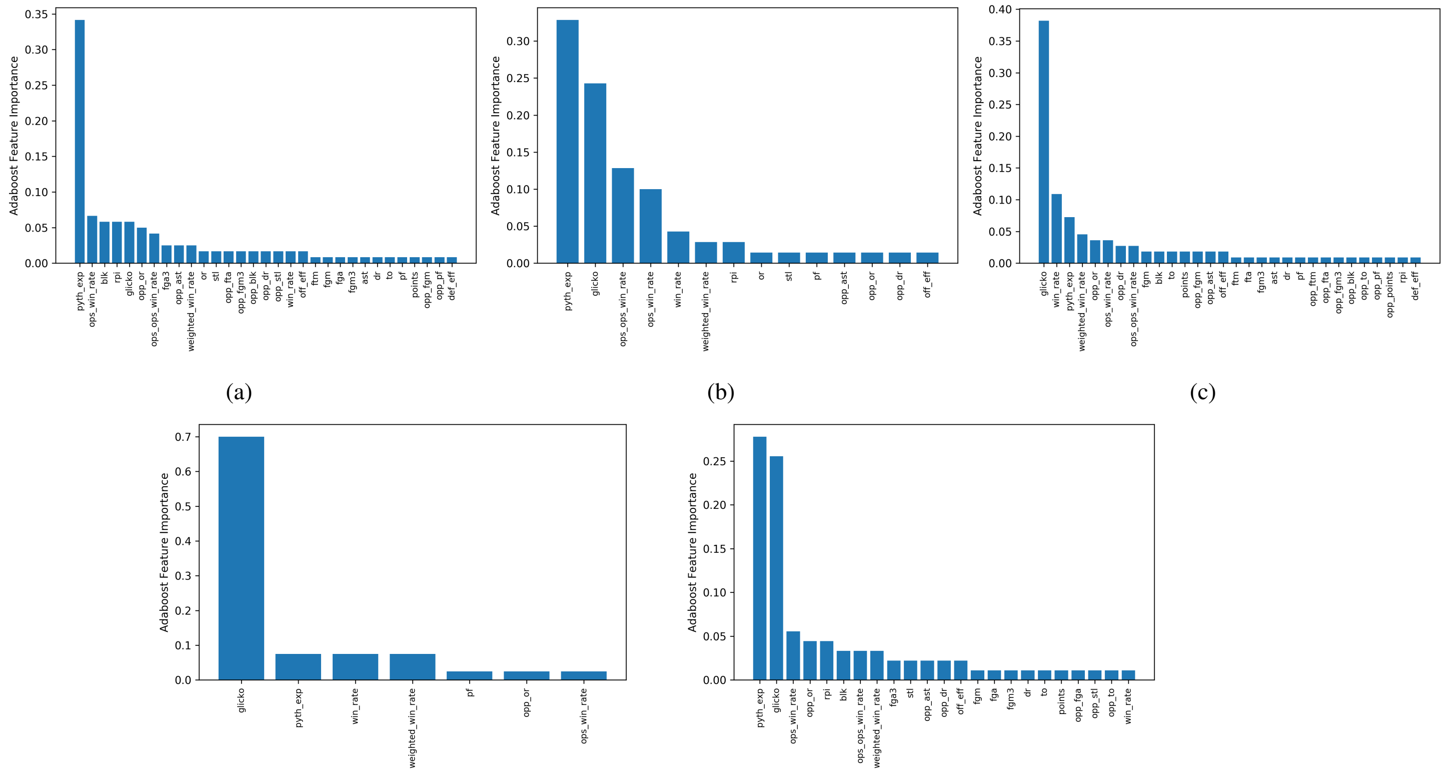
Table 6: SVM results on the varying datasets.

decision boundary more, we view them as more important. When 4 season's statistics are used, the Glicko ranking becomes much more important than the others. For 1 season data, again Glicko is the most important feature but its importance decreases. Also, Pythagorean expectancy becomes the second most important feature when we only have 1 season data. Another interesting feature is win rate. It becomes more important when we include more seasons. One possible explanation could be that win rate variates when we just look at single season. However, when we look at 4 seasons, it becomes a more stable metric.

7. Findings

The results in Figure 4 go against what we originally thought would occur when computing these distributions. It was expected that, rather than becoming less variable, they would become more variable due variable team strength cause by the NCAA four year limit. Upon further inspection, this result may indeed make sense. The average Glicko distribution is converging to show that most teams have a ”middle of the pack” true skill, and that high skilled teams are rare. Our hypothesis about past data being of poor predictive quality is better supported by Figure 5. Glicko predicts the 2017 tournament the best when it begins being gathered from 2016, and this is the best testing accuracy we obtained among any of our methods.

Glicko proved to be an effective predictor when more than one season was considered, this can be seen in the feature importance plots in Figures 7b-7d and 8b-8d. However, it was overtaken by Pythagorean expectation in the one season cases. This is likely due to some “ramp-up” time Glicko needs in order to converge to accurate estimation of a sys-



(a)

(b)

(c)

(d)

(e)

Figure 8: Adaboost feature importances for varying statistic calculation methods. Note that these plots only show features that were assigned a nonzero feature importance value and are in the same order as Figure 7.

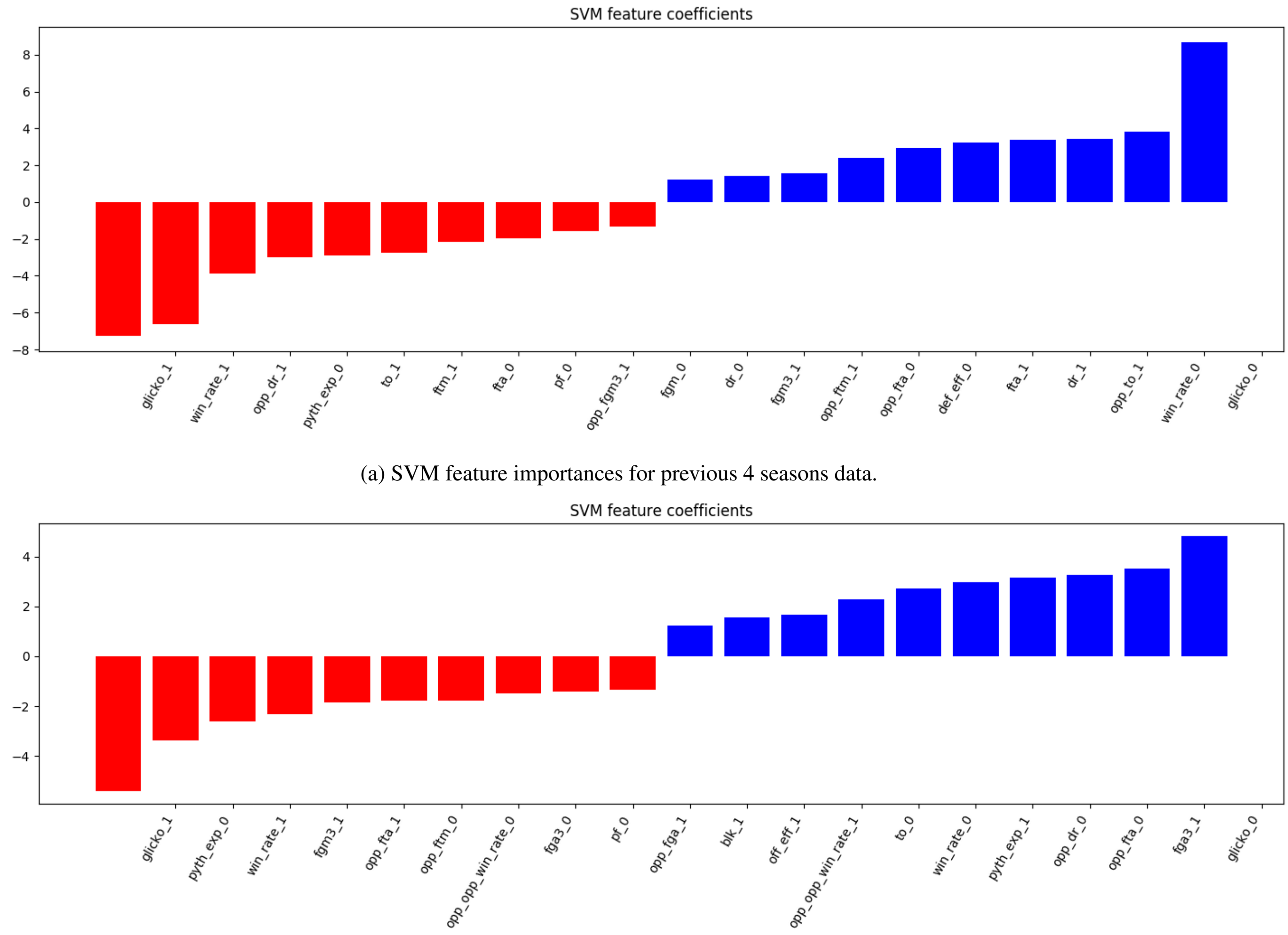
tem's true skill values that is not required Pythegorean expectation's mathematical formulation.

To conclude the Glicko findings we discuss the results of the Glicko and university spending experiment. Figure 6 shows a clear and strong trend between a team's all-time Glicko and how m much the associated university spent on men's basketball in the 2016-17 season. This leads to the classic question of which came first. Are these teams good only because their schools spend a lot of money on their men's basketball programs? Or, rather, at some point in history were some of these schools declared a "basketball school" which spurred on the increase spending due to increase revenue? In the case of the spending outliers, it seems that neither came before the other. The schools are massive, so they have a lot of money to throw around, and sports are excellent revenue sources. However, for the smaller schools in the top-25 all-time Glicko ranks like Gonzaga and Butler, there is some inherent quality that is not entirely captured by Glicko.

For the ensemble predictors, there is a weak signal that 2 to 3 seasons of statistic gathering may be the cut off for accurate predictions. Tables 6 and 4 show that RF and AdaBoost reached the highest testing accuracy in this range. This fact is further encouraged by the Glicko prediction accuracy results described in Figure 5 which yielded its best results closer to the 2017 tournament. With this said, these indications are quite weak, and all the ensemble prediction models performed relatively uniformly across the datasets. This result is consistent for SVM as well.

However, when compared to the tournament prediction scores, a different story is told. Prediction scores presented in Table 5 in this serial method are much more variable, having almost no correlation with there counterparts in Tables 6 and 4. Because it is most desirable to predict the entire March Madness bracket, these scores should be considered most closely, however from a predictive modeling view, accuracy is most important.

For the SVM results, we obtain the same weak signals as the ensemble classifiers. Again, the 3 season dataset performs the best, with the 2 seasons coming next. SVM also yields another interesting result relating the the separability of the data. Though the data is almost certainly not linearly separable in high dimensional space, there is some suggestion of a linear decision boundary. Our reasoning for this point is two-fold. First, we have a weak suggestion of linearity in projection to $\mathbb{R}^2$ shown in Figure 2. Second, grid



(a) SVM feature importances for previous 4 seasons data.

(b) SVM feature importances for previous 1 season data.

Figure 9: SVM feature importance plots. The suffixes in this plot refer to a team's relative ID order. For example, glicko_1 corresponds the the Glicko ranking of the team with the higher ID number, and glicko_0, the team with the lower ID.

search determined SVM with a linear kernel to be the best combination of hyperparameters.

8. Summary and Future Work

Here we have presented experiments to gain insight into the structure of NCAA mens basketball data. Our results show that games within the tournament can in fact be predicted with a reasonably high accuracy. Furthermore, we have shown that the Glicko-2 metric is also quite reliable and can, in some cases, be relied on exclusively. Though we have reached many conclusions here, there is still room for expansion on this work.

First, we point out that only Glicko-2 was given an extensive study into its properties. This was due to its popularity in previous work, and other systems. The same should be done for the other engineered features that were deemed relevant like RPI and Pythagorean expectation. More insight may be gained about the predictive quality of past data if these features are studied more closely as well.

Building the data matrices for this work was computationally prohibitive due to the necessity to calculate many statistics over the whole of the NCAA. In particular, the calculation of RPI grows cubicly relative to the size of the input matrix. Because of this restriction, we were limited to building data matrices that gathered statistics for four seasons at a maximum. The four seasons matrix alone took over 24 hours to compute. Given more time and computational resources, this work should be extended to use all of the data available to give a more insightful look into the effect past years have on future predictions.

For the predictors, since SVM and ensemble based predictors rely on different feature sets, we could combine their predictions with their respective confidences to create a stronger classifier. Since the SVM and ensemble methods seemed to identify different sets of features as important, this could lead to an overall more effective classifier.

9. Contributions

All of the code developed for this project can be found at https://github.com/ianwhale/march_madness

- Mehmet: linear methods and PCA experiment.
- Connor: feature engineering and data management.
- Ian: ensemble methods and Glicko experiments.

References

[1] E. Alfaro, N. Garca, M. Gmez, and D. Elizondo. Bankruptcy forecasting: An empirical comparison of adaboost and neural networks. *Decision Support Systems*, 45(1):110 – 122, 2008. Data Warehousing and OLAP.

[2] A. G. Association. March madness betting to top $10 billion. *American Gaming Association*, Mar 2017.

[3] A. G. Association. 97% of expected $10 billion wagered on march madness to be bet illegally. *American Gaming Association*, Mar 2018.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[5] J. Carbone, T. Corke, F. Moisiadis, and R. O'Brien. The rugby league prediction model: Using an elobased approach to predict the outcome of national rugby league (NRL) matches. *International Educational Scientific Research Journal*, 2(5):26–30, May 2016.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[7] Y.-S. Ding, T.-L. Zhang, and K.-C. Chou. Prediction of protein structure classes with pseudo amino acid composition and fuzzy support vector machine network. *Protein and peptide letters*, 14(8):811–815, 2007.

[8] A. E. Elo. *The rating of chessplayers, past and present*. Arco Pub., 1978.

[9] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[10] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, Sep 1999.

[11] M. Glickman, R. Ittenbach, T. G. Nick, R. O'Brien, S. J. Ratcliffe, and J. Shults. Statistical consulting with limited resources: Applications to practice. *CHANCE*, 23(4):35–42, Sep 2010.

[12] M. E. Glickman. The glicko system. *Boston University*, 1995.

[13] M. E. Glickman. Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics*, 28(6):673–689, 2001.

[14] G. Guo, S. Z. Li, and K. Chan. Face recognition by support vector machines. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 196–201. IEEE, 2000.

[15] M. Haghighat, H. Rastegari, and N. Nourafza. A review of data mining techniques for result prediction in sports. *Advances in Computer Science: an International Journal*, 2(5):7–12, 2013.

[16] T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA, 1995. IEEE Computer Society.

[17] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[18] J. Kubatko, D. Oliver, K. Pelton, and D. T. Rosenbaum. A starting point for analyzing basketball statistics. *Journal of Quantitative Analysis in Sports*, 3(3), 2007.

[19] M. J. Leone, D. Fernandez Slezak, G. Cecchi, and M. Sigman. The geometry of expertise. *Frontiers in Psychology*, 5:47, 2014.

[20] C. K. Leung and K. W. Joseph. Sports data mining: Predicting results for the college football games. *Procedia Computer Science*, 35:710 – 719, 2014. Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.

[21] B. Niu, Y.-H. Jin, K.-Y. Feng, W.-C. Lu, Y.-D. Cai, and G.-Z. Li. Using adaboost for the prediction of subcellular location of prokaryotic and eukaryotic proteins. *Molecular Diversity*, 12(1):41, May 2008.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] K. Pomeroy. Advanced analysis of college basketball, 2018.

[24] K.-S. Shin, T. S. Lee, and H.-j. Kim. An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications*, 28(1):127–135, 2005.

[25] J. Sun, M. yue Jia, and H. Li. Adaboost ensemble for financial distress prediction: An empirical comparison with data from chinese listed companies. *Expert Systems with Applications*, 38(8):9305 – 9312, 2011.

[26] N. Vlastakis, G. Dotsis, and R. N. Markellos. Nonlinear modelling of european football scores using support vector machines. *Applied Economics*, 40(1):111–118, 2008.

[27] A. Zimmermann, S. Moorthy, and Z. Shi. Predicting college basketball match outcomes using machine learning techniques: some results and lessons learned. *CoRR*, abs/1310.3607, 2013.