Shooting Stars

MARCH DATA CRUNCH MADNESS

AN OPTIMIZED MACHINE LEARNING MODEL FOR PREDICTING 2016 MARCH MADNESS MATCH-UP PROBABILITIES

AUTHORS:
Nan (Miya) Wang
Armi Thassim

1 Introduction

The National Collegiate Athletic Association (NCAA) Men's Basketball Tournament is informally referred to as "March Madness". With 68 college basketball teams competing in a single-elimination tournament, March Madness is played every spring in the US to determine the national championship of the major college basketball teams. The 68 teams are divided into four regions and respectively ranked 1 through 16 which determines the match-ups. Winning teams proceed through a single game elimination through 5 rounds. As a national tradition, "brackets" are filled in all over the country trying to predict the winners of each game until ultimately a champion of the tournament.

Various strategies are implemented to predict outcomes where people leverage regular season metrics: Seed, ratings percentage index ('RPI'), and even coach statistics are taken into account. Using various regular season metrics for each team in a given game, our goal is to use historical data for the past 14 years to predict the outcomes of every possible game for all 68 teams within the March Madness tournament.

2 Dataset

The original dataset for this project is very comprehensive and multi-dimensional, including game attributes (e.g.: location, season), team attributes (e.g.: seed, field goals, blocked shots, assists, offensive efficiency, defensive efficiency), professional basketball game indicators (e.g.: RPI, Preseason Coaches Poll Ranking) and coach career metrics. The dataset spans 14 seasons ranging from 2002 to 2015.

3 Feature Selection and Feature Engineering

The most challenging aspect of this project lay in identifying the most important features from such a large dataset. While the data is interesting in its original format, it is not necessarily efficient from a

machine learning perspective. One of the most difficult tasks is to ensure that accurate and comprehensive features are selected while combining multi-dimensional information.

If we were to just dump all of our initial features into the model to yield a ranking output of feature importance, this would give us, as one might expect, an undesirable result. For instance, "coach appearance count" appears on a higher rank than "team defensive efficiency". Therefore, to make the model more logical and informative, all features about coach information, due to its limited validity and measurability, were removed.

After some simple data transformations (such as, calculating distance from game location to the host location for each team), the feature of team seed which was in a ranking format was standardized into a score using the following function:

Team 1 Adjusted Seed = 0.5 + 0.03 * (Team 2 Seed - Team 1 Seed)

We standardized all the team features to get a more concise feature list and then transformed the dataset via feature reduction by deriving the variances of each games attributes:

Team 1 score of an attribute - Team 2 score of an attribute

This completed our transformation of the dataset to import into the machine learning models.

4 Model Building and Performance Validation

4.1 Model Building

4.1.1 Decision Trees and Random Forest:

Decision trees are flowchart-like tree structures, where each internal node (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. (Han, J 2006)

A random forest is an ensemble of decision trees which outputs a prediction value. Each decision tree is constructed by using a random subset of training data. After training a forest, each test row is passed through it, in order to make a prediction with final classification decisions made based on majority rule of the multiple decision trees.

4.1.2 Logistic Regression

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variable by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

Logistic regression can be seen as a special case of generalized linear model and thus analogous to linear regression. The model of logistic regression, however, is based on quite different assumptions (about the relationship between dependent and independent variables) from those of linear regression. In particular the key differences of these two models can be seen in the following two features of logistic regression. First, the conditional distribution $y \mid x$ is a Bernoulli distribution rather than a Gaussian distribution, because the dependent variable is binary. Second, the predicted values are probabilities and are therefore restricted to (0,1) through the logistic distribution function because logistic regression predicts the probability of particular outcomes.

4.1.3 Support Vector Machines

Each game in the training set X has a series of characteristics based on the difference in team statistics and a label y(win or loss). These can be plotted in a p-dimensional Euclidean space. The support vector machine then finds the hyperplanes with the greatest distance between them that partitions the space such that wins and losses are separated. To obtain probabilities, a logistic model is fit to the decision values for each binary classifier (Platt, 1999).

4.1.4 Naive Bayes

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. The Bayes Theorem algorithm is as follows:

$$\Pr(X|Y) = \frac{\Pr(Y|X) \cdot \Pr(X)}{\Pr(Y)}$$

4.1.5 Nearest Neighbors

This method uses the same starting framework as the support vector machine with each training set game labeled according to the difference in team features. For each new game lacking a label, we can obtain all of different statistics for same features. Then, the labeled points closest in Euclidean space to the new point, “vote” to determine the label of the new game. For example, if 3 out of 5 nearest neighbors are winners, then the new point will be assigned to be a winner, or given a 0.6 probability of winning.

4.2 Model Performance Evaluation and validation:

4.2.1 Evaluation Metric:

For each stage, a list of probabilities was submitted (with values between 0 and 1) where each team in the tournament would defeat every other team in the tournament, regardless of whether this match-up actually occurs. This year, that amounted to 2278 predictions. Judgements of the predictions are based on log-loss, or the predictive binomial deviance, of the games that actually occurred. This log loss function is calculated as follows:

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

The goal of this exercise is to find a set of predictions that minimizes Log loss for the unknown outcome vector y. The scoring method heavily penalizes being confident and wrong simultaneously. It is also important to balance between being too conservative and over confident. For instance, if the result were too conservative (e.g. ˆ y 0.5), then points would not be accrued, whereas if the result were too confident, this invites vulnerabilities to huge losses.

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

4.2.2 Model Performance

The entire training dataset has been run through six different Machine Learning models. Each model consists of grid searching or parameter tuning techniques, combined with cross validation, in order to ensure an optimized performance on the whole training dataset. Furthermore, by comparing model performance, Random Forest was found to far outperform the other five models in terms of log loss metrics. Performance of all models are provided in the following table:

Model	Log Loss	Accuracy
Random Forest	0.45	80.4%
Gradient Boosting Trees	0.48	80%
Logistic Regression	0.54	72%
Linear SVM	0.544	72%
Nearest Neighbors	0.56	73%
Gaussian Naïve Bayes	0.68	74%

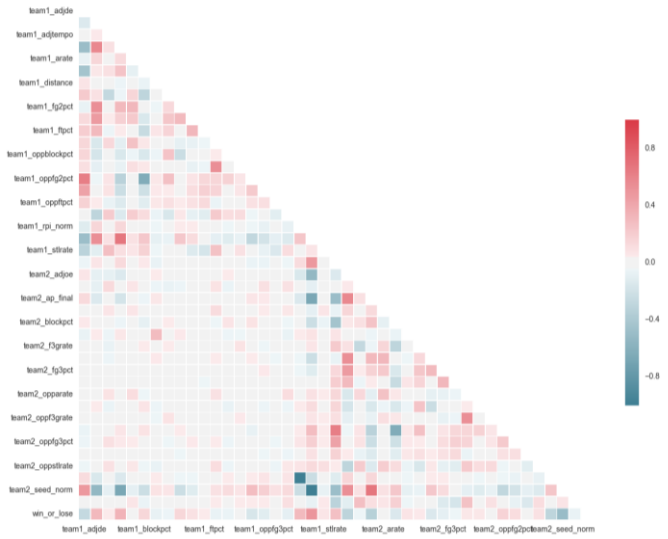
It is important to note however that the evaluations are based on the training dataset, which means it is also highly possible that variances and bias problems may have existed within the model building process.

Therefore, to optimize the models' genuine predictive strengths and to minimize potential problems of model over-fitting and under-fitting, techniques of further feature elimination, cross validation and model fusion were applied.

Feature selection techniques are widely-known as necessary for the following three reasons:

- Simplification of models to make them easier to interpret by researchers
- Shorter training times
- Enhanced generalization by reducing over-fitting.

Moreover, going by the theory that "good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other", features with a high correlation to other features were eliminated.

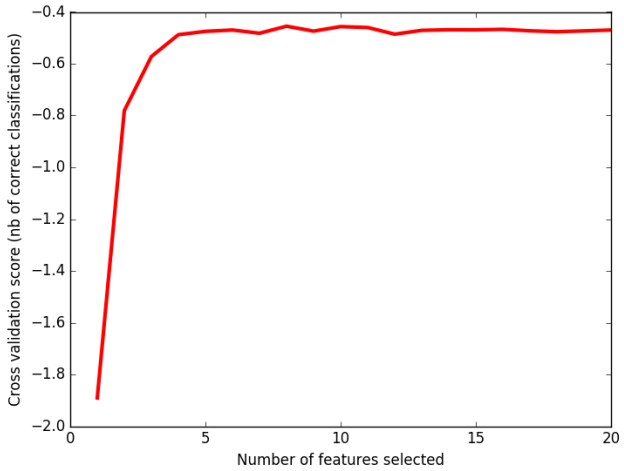


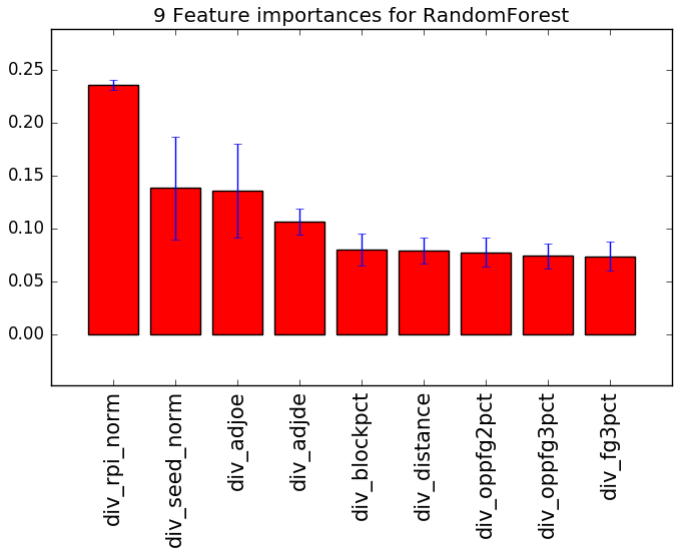
As shown in the heatmap above, the darker the module color, the higher the existing correlation between two features. As a result, it was easy to recognize that a few features in the dataset had a high correlation with other features and needed to be dropped before feeding it into the models.

Additional feature selection methods 'recursive feature elimination' and 'cross validation selection' were applied for further optimization. "Recursive feature elimination is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and weights are assigned to each one of them. Then, features whose absolute weights are the smallest are

pruned from the current set features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached." (http://scikit-learn.org/stable/modules/feature_selection.html)

By using these methods, the optimal number of features to put into the model was concluded to be nine, with the feature importance for each as follows:

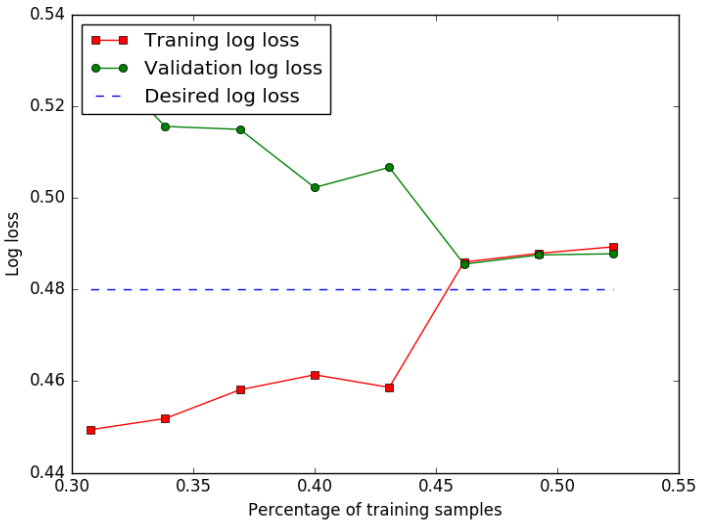




As shown in the charts above, the features the models consider important and necessary are team variances of RPI, Seed, adjoe(adjusted offensive efficiency), block shots, opponent two field goals, opponent three field goals and three field goals between Team 1 and Team 2. All other features are redundant and ineffective for the model performance. With all feature selection techniques applied,

effective features have reduced from the original 97 at the start to just 9. The log_loss, at the same time, decreased from 0.49 to 0.46 for the training dataset.

The next step was to check model performance on various training dataset sizes. Occasionally, with less training data, model parameter estimations have shown greater variance, while with less testing data, the performance statistic show greater variance. Therefore, it was necessary to partition the data so that neither variance would be too high. The entire training dataset, after cleanse and transformation, had 1,620 game records. The dataset was partitioned into a few different training and testing sizes and then studied to evaluate how the model responded to various train/test distributions, to pick the most appropriate partition size.



As shown in the learning curve above, the model performance does in fact vary by the size of the training data partitioning. The model suffers with serious over-fitting problems when the percentage of training samples are below about 43% of the entire dataset, but improves dramatically when the percentage increases to 45%. The model shows a stable performance for both training and validation dataset with a training sample percentage of about 50%. Therefore, the ideal partition size was determined to be 50%.

As a result, an unbiased log loss of about 0.48 to 0.50 would be expected when running the model to generate predictions on the test dataset.

The final step of the model building process was to use ensembling, where individual produced classifiers were merged via a weighted average (Opitz and Maclin, 1999). Prior experience has shown that ensemble methods work best using accurate classifiers which make errors in different input regions, because areas where one classifier struggles may be offset by other classifiers (Hansen and Salamon, 1990).

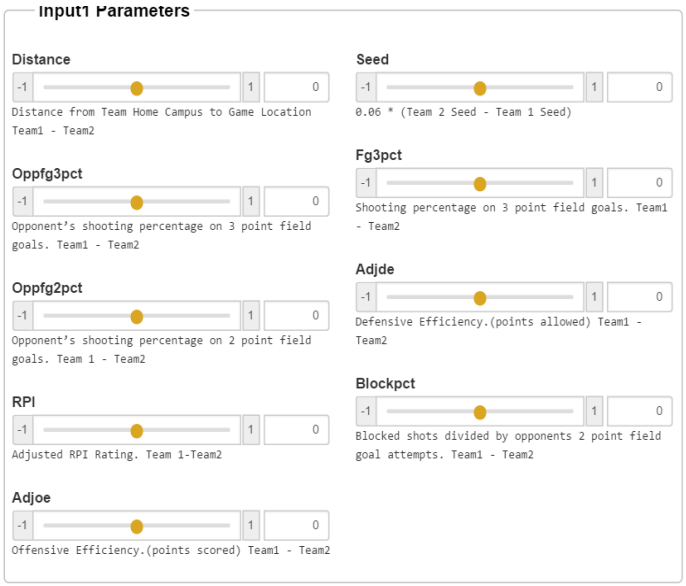
In this case, majority voting technique was applied to fuse three top-performing models: Random Forest, Gradient Boosting Trees and Logistic Regression. These method allow predictions based on different models to be combined, often using a type of voting schema, to determine one final prediction. "The benefit to using an ensemble method is that we can leverage the information gleaned from different methods, combining the benefits from each method while hopefully negating or minimizing the flaws in each. The most basic ensemble method is to average the results from a set of predictions. In this way, an ensemble method can avoid extreme predictions. Ideally, we want to assign optimal weights of each group of predictions."

5 Results and Conclusion

5.1 Results

An outcome of this project was a machine learning web application (http://shootingstarsnyc.azurewebsites.net/) based off of the built NCAA prediction model.

The application based on using user-inputted information for each of the nine features that the model used (Difference of RPI, Seed, adjoe (adjusted offensive efficiency), blocked shots, opponent two field goals, opponent three field goals and the three field goals between Team 1 and Team 2). Each of the features were normalized using MinMax scaler method.



After providing values for each feature, the application would yield a prediction on whether or not team 1 is likely to win. '1' indicates that Team 1 is more likely to win, while '0' means it is more likely to lose.

Result

Label	Value
output1	
Team 1 going to win or not?	1

5.2 Conclusion

The winning probability of a team within the NCAA Tournament is related to a variety of factors. To conclude, team attributes (e.g.: offensive efficiency, defensive efficiency and block shots), opponent statistics (e.g.: two point and three point field goals shooting percentage) and other factors (e.g.: distance) combined have the largest influence on a team's winning likelihood.

The higher a team scores in team attributes, which always means how much better that team is, the higher the likelihood that the team would also win. Furthermore, it may be obvious too that stronger opponents would lower a team's winning ability.

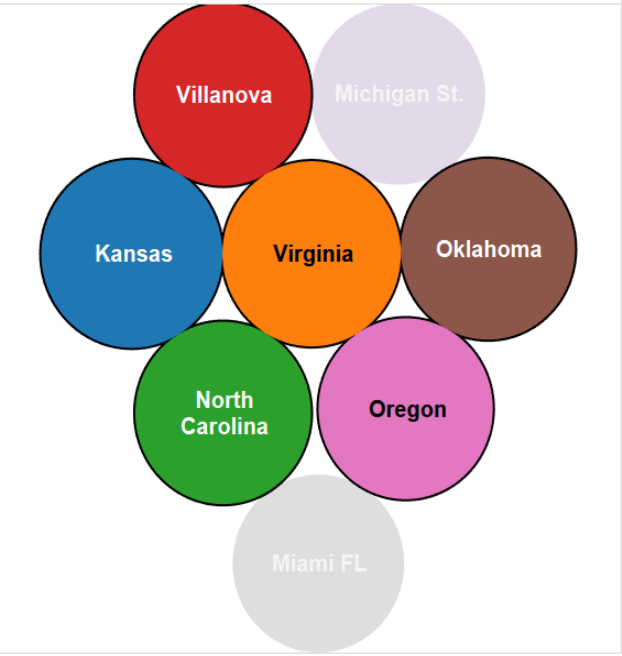
Moreover, other than a team's inner attributes, other factor of how far a team's home campus is from the game location also plays an important role. The closer to the host location of the competition, the higher likelihood that the location would give a team a better advantage.

In addition, useful indicators that can be referred to when predicting a team's winning probability include RPI (Rating Percentage Index) and seed.

This model achieves a high accuracy of 81% and a low log loss of 0.46 on training dataset. For 2016 testing data, a model performance with log loss ranging between 0.48 and 0.51 is expected after running model validation analysis.

For Round of 32, our model has turned out to be able to predict 25 teams out of 32 correctly, achieving the accuracy of 78%. In terms of Sweet 16 and Elite Eight, the model has caught respectively 12 of them for the former and 6 for the latter, both with the accuracy rate of 75%. Those results is within our expectation of model performance on 2016 testing data.

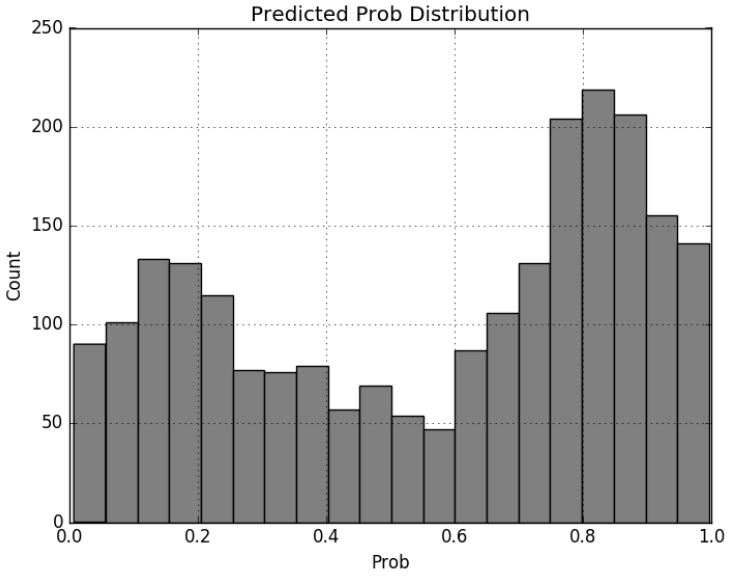
Predicted Elite Eight for 2016 NCAA



6 Scope and Limitation

With preliminary feature selection, all coach-related data without any solid foundation were dropped based on the analysis that aside from the 9 features used, that the other features play a far less important role in determining a team's winning probability at the NCAA finals.

As shown below with the distribution of the predicted probability for each game result of the 2016 NCAA tournament, this model shows an alignment with teams which are supposed to win but stays ambiguous for those considered as teams likely to lose. With that aspect, the model can be further improved in the future.



7 Future Research

For future analysis, transformed coach data may be a viable addition for the purpose of enhancing the diversity and strength of features. Further improvements of the model's predictive ability can be made on teams that are more likely to lose.

8 References

C. J. Burges. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2):121–167, 1998.

B. P. Carlin. Improved ncaa basketball tournament modeling via point spread and team strength information. The American Statistician, 50(1):39–43, 1996. doi:10.1080/00031305.1996.10473540. http://amstat.tandfonline.com/doi/abs/10.1080/00031305.1996.10473540.

N. V. Chawla. Many are better than one: Improving probabilistic estimates from decision trees. In Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment, pages 41–55. Springer, 2006.

T. G. Dietterich. Ensemble methods in machine learning. In Multiple Classifier Systems, volume 1857 of Lecture Notes in Computer Science, pages 1–15. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-67704-8. http://dx.doi.org/10.1007/3-540-45014-9_1.

P. Dizikes. Into the pool: Ncaa tourney betting booms. ABC News, March 2014. http://abcnews.go.com/Business/story?id=88479.

B. Efron. Bootstrap methods: another look at the jackknife. The annals of Statistics, pages 1–26, 1979.

A. Y. Govan, C. D. Meyer, and R. Albright. Generalizing google's pagerank to rank national football league teams. In Proceedings of the SAS Global Forum, volume 151-2008, 2008.

M. Greenfield. Team Rankings, 2014. Accessed: 2014-03-01, http://www.teamrankings.com.

T. Hastie, R. Tibshirani, and J. J. H. Friedman. The elements of statistical learning. Springer New York, 2nd edition, 2009. http://www-stat.stanford.edu/˜tibs/ElemStatLearn/.

C. D. Manning, P. Raghavan, and H. Sch¨utze. Introduction to information retrieval, volume1. Cambridge university press Cambridge, 2008. NCAA. National Rankings, 2014. Accessed: 2014-03-01, http://stats.ncaa.org.