Basketball game prediction with Machine Learning Techniques

Monsonís J. Ferrando† and Vallés M. Porta‡

†Barcelona Supercomputing Center
‡Polytechnic University of Catalonia

Problem

The purpose of this work is to dive into the forecasting methods used in basketball, especially in NCAA March Madness competition *Google Cloud and NCAA ML Competition 2019-Men's*[2] data set. NCAA refers to the college basketball league held in USA. Each year, at the end of the regular season, the 68 best teams [3] are selected to compete in the NCAA Basketball Tournament, which takes place in an single-elimination format.

Definitions

Some important basketball metrics definitions used in this work are:
Poss: represents the number of possessions a team uses during a game, and is calculated as follows

$$Poss = 0.96 \cdot (FGA + TOV + 0.44 \cdot FTA - OR)$$

Offensive Rating (OffRtg): points scored by averaging over 100 possessions

$$OffRtg = 100 \cdot \frac{PointsMade}{Poss}$$

Defensive Rating (DefRtg): defense end metric based on team's pace

$$DeffRtg = 100 \cdot \frac{PointsAllowed}{Poss}$$

NetRtg: shows a balanced indicator of a team offensive and defensive capabilities

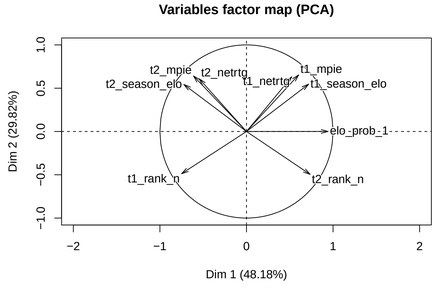
$$NetRtg = OffRtg - DeffRtg$$

PIE: Team Impact Estimate groups several indicators into a single metric:

$$PIE = PTS + FGM + FTM - FGA - FTA + DR$$
$$+$$
$$0.5 * OR + AST + STL + 0.5 * BLK - PF - TOV$$

Elo Rating[1]: represents the 'shape' of a team by updating its value every game with

$$K * \frac{margin + 3}{7.5 + 0.06 * (elo_{winner} - elo_{loser})} * (1 - \frac{1}{10 - \frac{(-elo_{winner} - elo_{loser})}{400}} + 1$$



Based on box-score statistics collected during the regular season, a new data set with advanced statistics as features is created [5].

Models

The data is split into training and test samples. 2003-2013 tournaments have been used for training purposes while the rest of the data (2014-2018 tournaments) for the testing phase. Finally, the predictive model is submitted to Kaggle and tested with every possible 2019 tournament matchup.
The objective function to minimize is the *LogLoss*, defined as

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}(y_i \cdot \log p_{i,j} + (1 - y_i) \cdot \log(1 - p_{i,j}))$$

Where n refers to the number of games, $p_{i,j}$ refers to the predicted probability of team i winning over team j and y_i the real result of the game (1 if team i wins, 0 otherwise).
Following some of the methods proposed in [6] and [4] the next models have been tested:

Logistic Regression

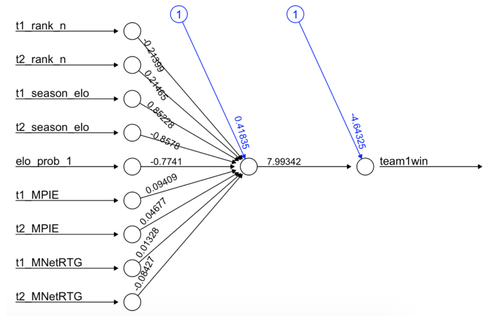
Given $P(w_1|X)$, the conditional probability of team 1 winning given the predictors values X, $logit(P(w_1|X)$ can be rewritten as a linear combination of the predictors $\beta^T X + \beta_0$. After performing the Logistic Regression and a Z-test on the predictors coefficients, both teams rankings seem to be the most significant features.

	Estimate	Std. Error	z value	Pr(>|z|)
t1_rank_n	-0.1220677	0.0421675	-2.8948275	0.0037937
t2_rank_n	0.0862475	0.0385741	2.2358918	0.0253589

LogLoss test results output by the LR have been 0.54.

Neural Network

A one-hidden layer MLP has been tested with the full standaradized features of the data set. 10-Fold Cross-Validation process has been used to tune the number of neurons in the hidden layer. Results of the cross-validation yield optimal results for a single unit in the hidden layer MLP.



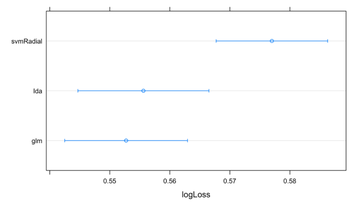
LogLoss test results with the above model have been 0.56.

Random Forest

After performing Random Forest on our training data set, we obtained the probabilities predicted by the model and got a boost with respect to a single decision tree up to 0.702 accuracy, but most importantly, LogLoss values reduced significantly respect to a single decision tree (0.58 LogLoss).

Ensemble Method

An ensemble is tried. In this case, a Logistic Regression (glm), a Linear Discriminant Analysis (lda) and a Support Vector Machine with a radial basis function (RBF) kernel ensemble.



High models correlations can be shown

	lda	glm	svmRadial
lda	1.0000000000	0.9886192866	0.9565998199
glm	0.9886192866	1.0000000000	0.9334079233
svmRadial	0.9565998199	0.9334079233	1.0000000000

LogLoss result on the test set is around 0.58.

Comparison

Final LogLoss values in 2019 tournament (kaggle score)

	0.5Pred	LR	NNet	RForest	Ensemble
2019 score	0.67	0.48	0.49	0.45	0.69

Remarks

With Random Forest we have scored 0.45006, which would have gave us the 39^{th}/836 position in the Kaggle competition, which would be Top 5% result. In future work we could include new features in the models. Since the aggregated statistics showed in this work are calculated as averaging every game each team plays during the regular season, these features could be missing important changes in a team. For example, a team could lose its most valuable player in the last days of the regular season. Although Elo rating gets updated every game, we could give higher weights to last season games or add new features taking into account just the last n games prior the tournament.
All the code and data can be found at https://github.com/javiferran/ML-Project

References

[1] *Basketball Elo rating calculation*. 2019.

[2] *Google Cloud and NCAA ML Competition 2019-Men's*. 2019.

[3] *How the field of 68 teams is picked for March Madness*. 2019.

[4] Michael Lopez and Gregory Matthews. "Building an NCAA mens basketball predictive model and quantifying its success". In: *Journal of Quantitative Analysis in Sports* (2014).

[5] *NBA Advanced Stats*. 2019.

[6] Lo-Hua Yuan et al. "A mixture-of-modelers approach to forecasting NCAA tournament outcomes". In: *Journal of Quantitative Analysis in Sports* 11 (2015), pp. 13–27.