Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona

Machine Learning

NCAA ML Competition 2019-Men's data set

Authors: Javier Ferrando Monsonís - Marcel Porta Vallés
Master in Innovation and Research in Informatics.

Barcelona - June 23, 2019

Contents

1 Problem description and goals

In this work, we aim to dive into the forecasting methods used in the game of basketball for game outcome prediction. The practical analysis is done by using the dataset provided by the *Google Cloud and NCAA ML Competition 2019-Men's*[2]. NCAA refers to the college basketball league held in USA. Each year, at the end of the regular season, the 68 best teams [7] are selected to compete in the NCAA Basketball Tournament, which takes place in an single-elimination format. Due to the high unpredictability of the tournament winner in the past years, this final championship has been popularly called the March Madness.
Teams play against each other very few times during one season, or in the case of the NCAA competition, in most cases not even once. Moreover, team's players change from season to season and injuries make even more difficult to find two exactly matchups between the same two opponents. For this reason, previous games between the same teams can not be considered in predictive models. Instead, a comparison of two teams by how well they performed during the entire regular season needs to be done in order to get insightful information about teams.

2 Related work

Several studies investigated the prediction of basketball games. In [8] this problem is proposed with a direction towards the NCAA March Madness, reinforced by the appearance of the Kaggle competition. A series of machine learning models such as Logistic Regression, regularized Logistic Regression, Stochastic gradient boosting, Neural Netowrks and ensemble of the above mentioned are attempted in [1]. Moreover, this paper presents different ranking systems as possible model predictors such as Pomeroy, Sonny Moore's Power Ratings, Massey, ESPN and Percentage. Logistic Regression is tried also in [6], but including Las Vegas Point Spread metric. A model for soccer is shown to be competitive in basketball prediction in [9] with the use of a generative model that employs basic attack and defense statistics focused towards beating bookmakers.

3 Understanding the data

This section explains in detail all the previous steps before developing the predictive model. The content of this section is an initial description of the available data and its meaning, an exploratory analysis and the extraction of new explanatory variables by means of feature engineering.

3.1 Available data

Multiple csv files are given as input data in the *Google Cloud and NCAA ML Competition 2019-Men's*[2]. This files are split in different sections of data, each section is composed by different data sets of the same nature.

Although one can feel overwhelmed with amount of information given, useful data for building a predictive model can be found in a few csv files. [6] affirms that two easily accessible sets of predictors for NCAA basketball tournament outcomes are information from prior tournaments and results from regular season competition. We agree on the importance of this data, which can be found specifically in:

- **NCAATourneySeeds.csv:** election done by the NCAA's committee, which rank every team before the start of the tournament based on a poll[7], for 1985-2019 seasons.
- **NCAATourneyCompactResults.csv:** tournament results scores for 1985-2019 seasons.
- **RegularSeasonDetailedResults.csv:** detailed season games results, with basic box-scores statistics of the two teams.

3.2 Exploratory Analysis

Detailed season statistics

'RegularSeasonDetailedResults.csv' contains box-score statistics of every game from 2003 season up to now. With this information we can analyze the relationship between box-score data and the outcome of a match. To do that, each possible result (Win or Loose) distribution is illustrated depending on the frequency of the values obtained for each of the traditional basketball statistics.

In the first figures three types of Field Goal metrics are exposed: %FGM, %3PM and %FTM. All of them are the ratios as a result of shots made from those attempted (M/A).

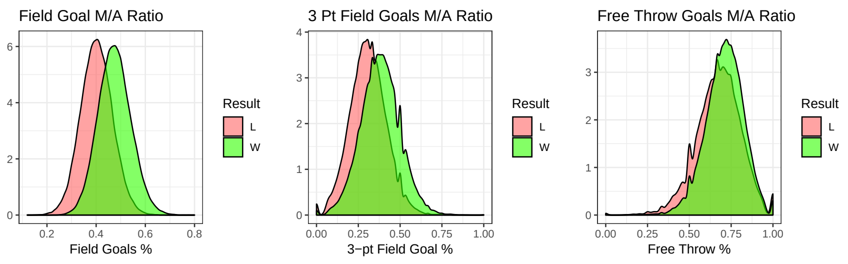

Figure 1: Statistics related with scoring.

On the other hand, statistics that are not directly related with scoring are shown.

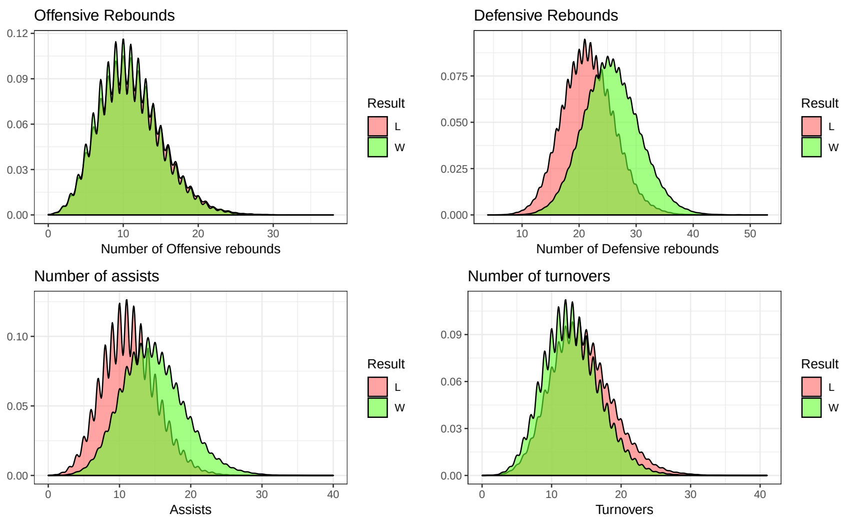

Figure 2: Statistics not directly related with scoring.

With this plots an intuition of what makes a team win can be acquired. It can be deduced that the winner team achieves better Field Goal Made/Attempt ratios in the whole range of possible shots, higher values of defensive rebounds and assists while lower values of turnovers.

3.3 Feature extraction

Once the exploratory analysis and an initial understanding has been done, the next step is to decide which features can be used in a predictive model and if we can create new variables based on existing ones.

Nowadays common advanced statistics [3] used by experts to measure a team performance per game are the following ones:

- **Poss**: represents the number of possessions a team uses during a game, and is calculated as follows
$$Poss = 0.96 \cdot (FGA + TOV + 0.44 \cdot FTA - OR)$$
Where FGA = Sum of attempted field goals, TOV = Turnovers, FTA = Free throws attempted and OR = Offensive rebounds.
- **Offensive Rating (OffRtg)**: points scored by averaging over 100 possessions.
$$OffRtg = 100 \cdot \frac{PointsMade}{Poss}$$
- **Defensive Rating (DefRtg)**: same as previous but on the defense end
$$DeffRtg = 100 \cdot \frac{PointsAllowed}{Poss}$$
- **NetRtg**: Offensive rating - Defensive rating, shows a balanced indicator of a team offensive and defensive capabilities
$$NetRtg = OffRtg - DeffRtg$$
- **EFG%**: effective Field Goal takes into account that 3-pointers are worth 1.5 more than 2-pointers,
$$eFG\% = FGM + \frac{0.5 \cdot 3PM}{FGA}$$
- **PIE**: Team (or Player) Impact Estimate groups several indicators into a single metric and is calculated as follows:
$$PIE = PTS + FGM + FTM - FGA - FTA + DR + 0.5 * OR + AST + STL + 0.5 * BLK - PF - TOV$$
Where FTA = Free throws attempt, DR = Defensive rebounds, AST = Assists, STL = Steals, BLK = Blocks and PF = Personal fouls
- **Elo rating**: 'method for calculating the relative skill levels of players in zero-sum games such as chess. It is named after its creator Arpad Elo, a Hungarian-American physics professor'[4]. In this work, a basketball adaptation is used [5]. This feature brings information about the 'form' of a single team. Every team in the competition starts with an Elo rating of 1500 points and, each game of every season its value gets updated based on game scores and home advantage factor by the following rule
$$Update = K * \frac{margin + 3}{7.5 + 0.06 * (elo_{winner} - elo_{loser})} * (1 - \frac{1}{10 - \frac{(-elo_{winner} - elo_{loser})}{400} + 1})$$
Where margin represents the game score difference, elo_{winner} and elo_{loser} the elo points prior the game and k the update factor, in charge of adjusting the speed at which Elo ratings are updated from game to game (K = 20 for NBA). The team that plays at home receives 100 Elo points, while the visitor gets substracted 100.

Below is shown the evolution of end-season Elo points for each of the 4 teams that achieve the Final Four in 2019 tournament. All of them show a positive trend in the past years.

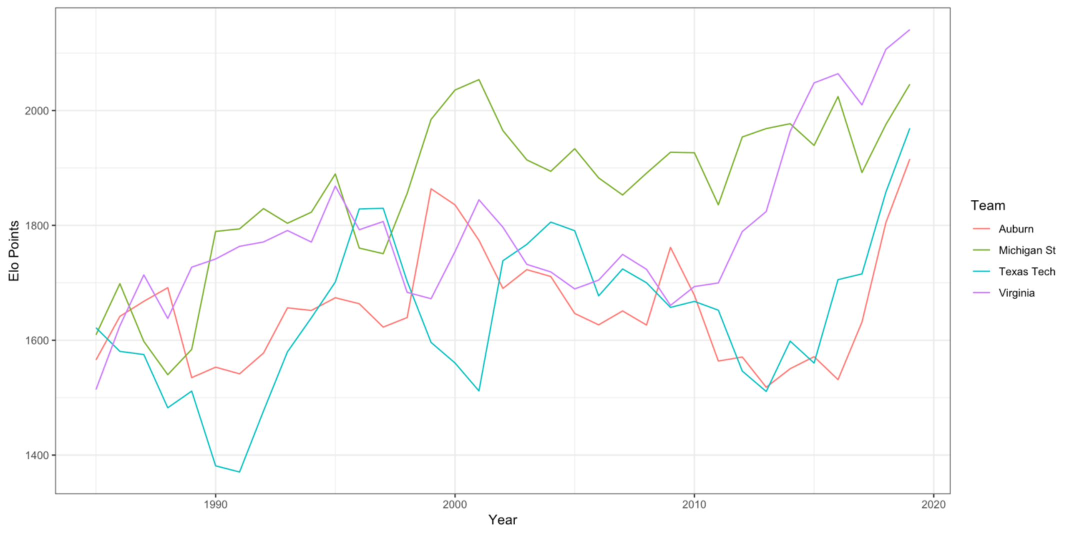

Figure 3: Elo points 2019 Final Four teams

Aggregated statistics from the whole performance of a team's season, as the ones showed before can be utilised to predict the winner in a tournament matchup.

All this advanced statistics will be stored in a new file named "NCAASeasonDetailedResultsEnriched.csv" and "season_elos.csv".

3.4 Pre-processing

The pre-processing could be summarized in the following diagram:

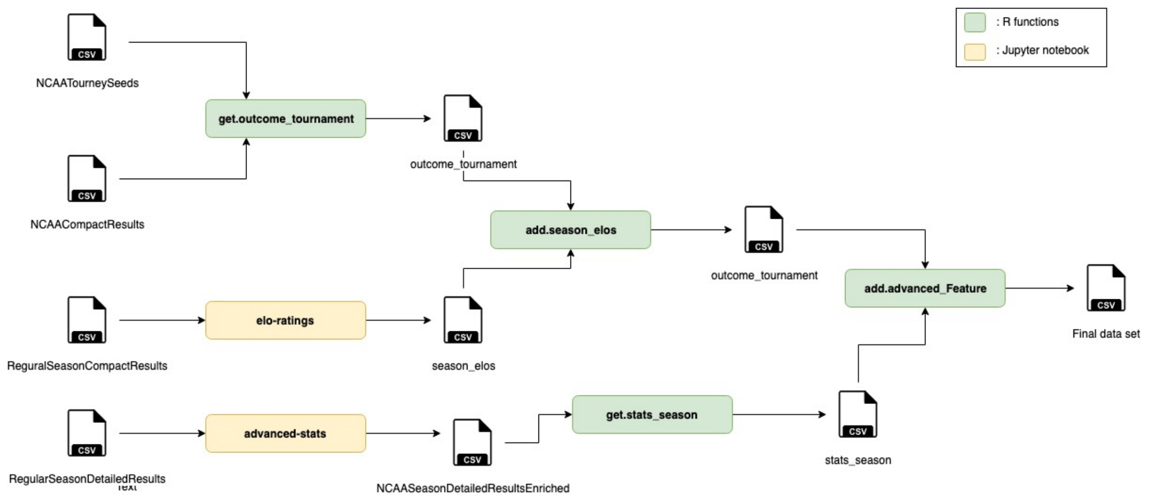

Figure 4: Final pre-processing flux diagram.

- **get.outcome_tournament**:
  - Replaces NA seeds from NCAATourneySeeds.
  - Gets numerical seeds without letters from NCAATourneySeeds.
  - Keeps only season, day num and win/lossing team from NCAACompactResults.

  - Keeps only data from same or earlier seasons than 2003.
  - Adds t1_rank and t2_rank, that will be a numeric variable representing the seed.

- **elo-ratings [10]**: computes Elo rating as explained before.
- **advanced-stats[11]**: computes the before described advanced statistics.
- **add.advanced_Feature**: adds any chosen advanced statistic.

On the final set each row represents a game in an specific tournament year (season) between teams 1 and 2. The result of the game, **team1win** (1 if team 1 won, 0 otherwise), is the target variable . Following columns represent a subset of both teams extracted features discussed before.

	team1win	t1_rank_n	t2_rank_n	t1_season_elo	t2_season_elo	elo_prob_1	t1_mpie	t2_mpie	t1_netrtg
407	1	1	16	1988.413	1423.301	0.9627824	0.6205715	0.4941001	26.621518
365	1	1	9	2121.651	1820.973	0.8495201	0.6078353	0.5631013	16.160860
501	0	11	2	1889.214	1969.172	0.3869202	0.5872799	0.6019132	14.636510
607	1	1	16	1953.955	1513.566	0.9265649	0.6297290	0.5478787	19.772421
475	0	14	3	1640.586	1941.502	0.1503043	0.5306970	0.5966854	6.000455
208	0	6	11	1871.976	1744.552	0.6755753	0.5702567	0.5612980	10.128534

Table 1: Random sample with a subset of features of the self-made data set

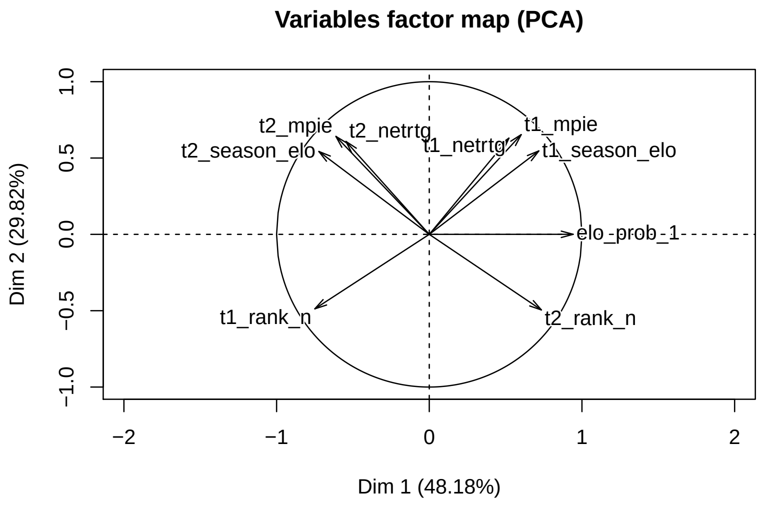

Figure 5: Variables in PCA

In the PCA plot is showed the correlation between the variables and the first principal components. Team 1 and team 2 aggregated statistics appear inversely correlated with respect to the first principal component and both teams rank are plotted in opposite directions to the aggregate statistics. Teams with high MPIE, Season elo and Netrtg are rank lower.

4 Models

The data is split into training and test samples. 2003-2013 tournaments has been used for training purposes[1] while the rest of the data (2014-2018 tournaments) for the testing phase. Finally, the

[1]Note that season games are only used to extract aggregated features describing the 'quality' of each team entering the tournament

prediction model is submitted to Kaggle and tested with every possible 2019 tournament matchup[2].

The loss function used for training and testing the proposed models is the one provided by Kaggle competition

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

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) Below is showed the behaviour of the presented LogLoss function [1] when the prediction obtained is wrong (left) and when it is correct (right).

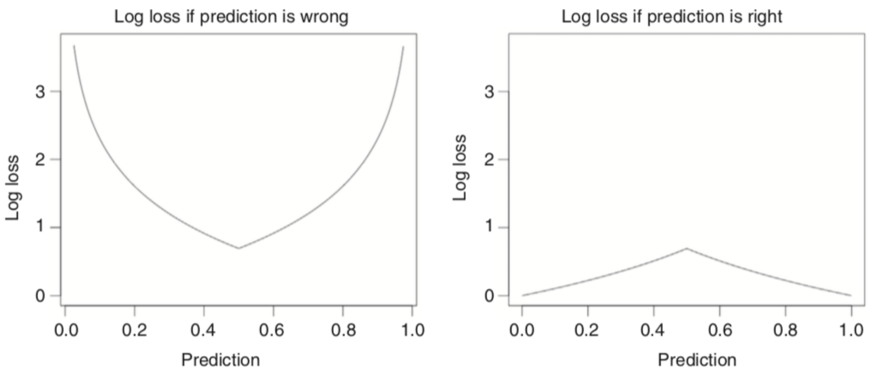

Figure 6: LogLoss behaviour

So, the smaller values of LogLoss function the better predictions achieved by the model.

The number of total final predictions for the 2019 season equals to every possible matchup that could take place $\binom{63}{2} = 2278$, but only 67 of them (the ones who actually occurred) will be used in the loss function.

4.1 Logistic Regression

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.3665695	4.7299151	0.5003408	0.6168351
t1_rank_n	-0.1220677	0.0421675	-2.8948275	0.0037937
t2_rank_n	0.0862475	0.0385741	2.2358918	0.0253589
t1_season_elo	0.0026498	0.0031840	0.8322290	0.4052797
t2_season_elo	-0.0035884	0.0031796	-1.1285886	0.2590714
elo_prob_1	-1.0054503	2.6053797	-0.3859131	0.6995610
t1_mpie	1.6839688	4.5263052	0.3720405	0.7098627
t2_mpie	-1.6231156	4.6706039	-0.3475173	0.7282027
t1_netrtg	0.0130383	0.0179830	0.7250311	0.4684329
t2_netrtg	-0.0115212	0.0192459	-0.5986334	0.5494174

Table 2: Logistic Regression coefficients

[2]We can do this since the tournament has already taken place

By means of R, a Logistic regression model has been built using *glm* taking as predictors Regular season Elo points, PIE and NetRtg from both teams. The family used is binomial since we have two classes to predict and the link function, the logit[3] one.

From the summary of the linear model created in R we can observe the z-statistic performed over the regression coefficients. Z-test null hypothesis H_0 : the regression coefficient is equal to 0, so p-values lower than 0.05 indicate that we reject the null hypothesis with a 95% confidence and affirm that the coefficient is significantly higher than 0 and conclude that has a significant effect in the model. The ranking of both teams seem to be the most significant features.

LogLoss test results output by the LR have been 0.54.

4.2 Multi Layer Perceptron

A one-hidden layer MLP has been tested with the full features data set and standardized values to mean = 0, std = 1. 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. So, as expected, results are comparable to the ones obtained by the Logistic Regression model. Below can be seen the final MLP model with the values of the weights of each connexion between the input-layer and hidden-layer unit, and between hidden-layer single unit and the output layer, with both biases (in blue).

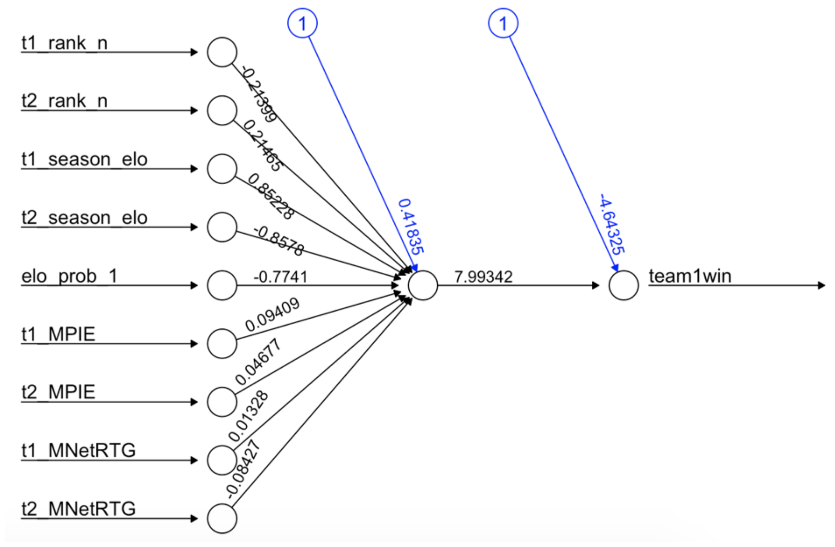

Figure 7: MLP plot with weights and biases

Based on the weight values, season Elo points from from teams are greater weighted by the model. LogLoss test results with the above model have been 0.56.

4.3 Decision Trees

A decision tree has been trained with 10-Fold Cross-validation in order to get the optimal size of the tree. In Figure 8 it is shown training error (red line) and CV error (blue line) of the optimal tree for each complexity parameter *cp* value, which controls the maximum number of leafs.

[3] $logit(p) = \log\left(\frac{p}{1-p}\right)$

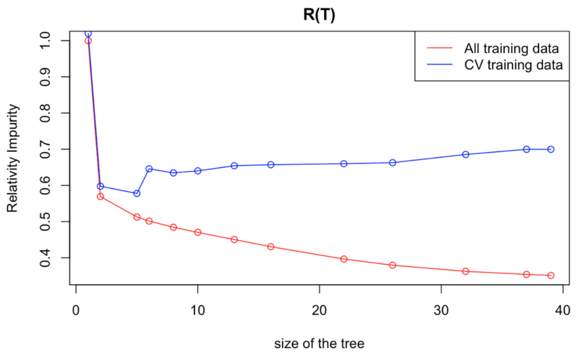

Figure 8: CV and training error vs size of the tree

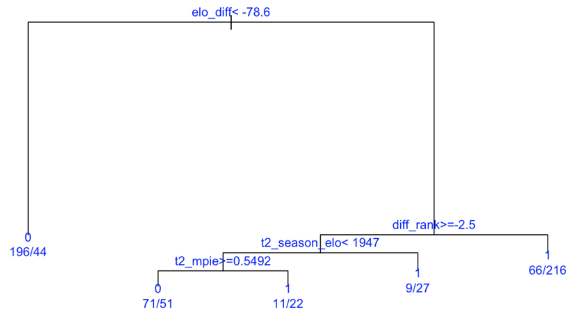

Figure 9: Optimal decision tree

The lowest CV-error occurs with $cp = 0.08$, that means 5 leafs. Although it obtains an accuracy of 0.64, the LogLoss value in the training data set goes off to 5.81. This could be due to the differences between the Gini Gain (cost function used by decision trees) and the LogLoss metric used to evaluate the classification model. Probabilities results appear to be very large or very low, which is penalized heavily by the LogLoss function.

4.4 Random Forest

Bootstrap resampling allows to estimate values of a statistic obtaining distinct data sets by repeatedly sampling observations from the original data set. Random samples are extracted to form bootstrap data set (resamples). The same observation can appear multiple times in the same bootstrap resample.

Bagging (Boostrap aggregation) is used to reduced the variance of a machine learning model by applying the same learning procedure to each of the bootsrap resamples and averaging the predictions.

Random Forest implements bagging, where each of the models is a different decision tree. The difference with simple bagging is that each of the trees is trained using only a random subset of the data set features, reducing the correlation between the different trees that make up the forest.

After performing Random Forest on our training data set, we obtained the probabilities predicted by the model and obtained 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 decsion tree (0.58 LogLoss).

4.5 Ensemble Methods

Finally, as mentioned in [1], 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 is built with the help of *caret* package.

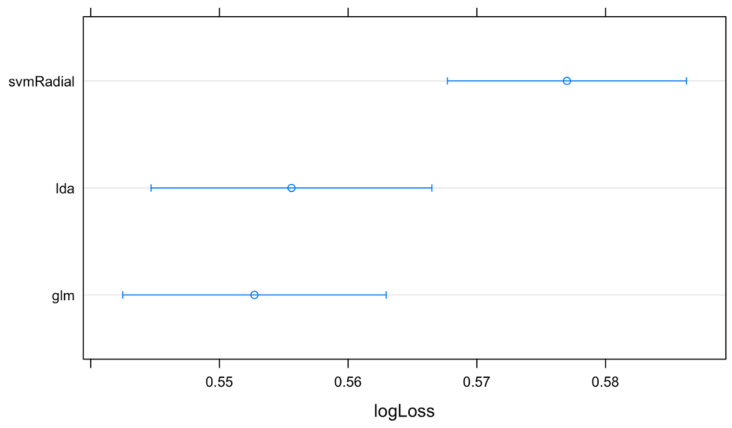

Figure 10: Individual models LogLoss results

Nevertheless, correlation between different models is pretty high which indicate that an ensemble of this methods could not obtain good results.

	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, which doesn't improve the single glm tested in section 4.1.

5 Conclusions

In this work we have explained simple and advanced statistics used in the game of basketball as well as their influence in the outcome of a game. By means of an exploratory view, feature engineering and correlation analysis, we have realised some metrics that could be useful in some predictive models. Finally, we have tested those predictors with Logistic regression, MLP, Decision Trees, Random Forest and an Ensemble methods. After testing each model on the prediction of the 2019 tournament, these have been the resutls:

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

With Random Forest we have scored 0.45006, which would have gave us 39/836 position in the Kaggle competition, which is a Top 5% result. The difference between test data set (2014-2018 tournament) and 2019 results could be explained by the low number of upsets occurred in this year's tournament compared with recent years. In future work we could include new features in the models. Since the aggregated statics showed in this work are calculated 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

Bibliography

[1] Lo-Hua Yuan, Anthony Liu, Alec Yeh, Aaron Kaufman, Andrew Reece, Peter Bull, Alex Franks, Sherrie Wang, Dmitri Illushin and Luke Bornn. A mixture-of-modelers approach to forecasting NCAA tournament outcomes. J. Quant. Anal. Sports 2015; 11(1): 13–27

[2] Google Cloud and NCAA ML Competition 2019-Men's. https://www.kaggle.com/c/mens-machine-learning-competition-2019.

[3] NBA Advanced Stats. https://stats.nba.com/help/glossary/.

[4] Elo rating system. Wikipedia. https://en.wikipedia.org/wiki/Elo_rating_system. 2019.

[5] 538. Elo rating calculation. https://fivethirtyeight.com/features/how-we-calculate-nba-elo-ratings/. 2019

[6] Lopez, Michael and Matthews, Gregory. Building an NCAA mens basketball predictive model and quantifying its success. Journal of Quantitative Analysis in Sports. 2014.

[7] How the field of 68 teams is picked for March Madness. https://www.ncaa.com/news/basketball-men/article/2018-10-19/how-field-68-teams-picked-march-madness

[8] Mark E. Glickman and Jeff Sonas. Introduction to the NCAA men's basketball prediction methods issue. J. Quant. Anal. Sports 2015; 11(1): 1–3

[9] Francisco J. R. Ruiz and Fernando Perez-Cruz. A generative model for predicting outcomes in college basketball. J. Quant. Anal. Sports 2015; 11(1): 39–52

[10] Elo ratings based on regular season games. https://www.kaggle.com/lpkirwin/fivethirtyeight-elo-ratings

[11] Enriching NCAATourneyDetailedResults.csv with Advanced Stats. https://www.kaggle.com/lnatml/feature-engineering-with-advanced-stats