Machine Learning and Data Mining project: Basketball

Guido Cera[1] and Simone Cappiello[2]

[1] problem statement, solution design, solution development, data gathering, writing
[2] problem statement, solution design, solution development, data gathering, writing

Course of 2022-2023 - Introduction to Machine Learning

1 Problem statement

A dataset containing information about all NBA games from season 2003 to 2022 is available. The aim of the project is to design and assess one or more machine learning models able to predict the outcome of games. It is, therefore, a binary classification problem where the following convention for the response variable Y is adopted:

$$Y = \begin{cases} 1, & \text{the hosting team of the game, "home team", wins (Negative);} \\ 0, & \text{the visitor team of the game, "away team", wins (Positive)} \end{cases}$$

Each row of the provided dataset, $D = X \times Y$, contains team-level details, X, about a single game (and its outcome Y), most of which can be known only after the game has ended. We are instead in the position where the game has not yet been played and want to predict its outcome. Consequently, feature engineering to compute usable predictors is needed. It will take this form:

$$f_{pre-proc} : \mathcal{P}(X \cup Y)^1 \longrightarrow X'$$

- X provided: each observation contains team-level data about one game (date, points scored, rebounds by each team etc.), it cannot be directly used to predict Y;
- X' to compute: contains the actual predictors that summarize, for each game, results of previous matches played by the two teams involved in the game.

[1] *$\mathcal{P}(A)$ is the powerset of A, the set of all possible subsets of A (notation $\mathcal{P}^*(A)$ allows duplicates).*

Then it is required to find a suitable learning technique:

$$f'_{learn} : \mathcal{P}^*(X' \times Y) \longrightarrow M \qquad f'_{predict} : X' \times M \longrightarrow Y$$

2 Assessment and performance indexes

Considering we are dealing with binary classification we choose the following assessment methods:

- **Accuracy**: computed using 10-fold Cross Validation.
- **Confusion Matrix**.
- **ROC Curve and AUC**.
- **FNR, FPR**: we identify visitor team victory as positive ($Y = 0$, the least frequent case) and home team victory as negative ($Y = 1$).

To compare the **efficiency** of different learning techniques we use **time** spent in learning the models, expressed in seconds.

3 Proposed solution

3.1 Algorithmic overview

A sketch of our approach to solve the problem is the following:

1. Feature engineering
2. For each chosen Learning Technique:
   (a) Choose Features (select all in first iteration, optionally remove least significant ones in following iterations)
   (b) Fit model with chosen features, using 10-fold CV
   (c) Assess Model, using indexes presented in Section 2.
   (d) If satisfied [2], stop, otherwise go back to (a).

The learning techniques we focused on are Random Forest and Logistic regression. A detailed description of the first step, the most important, follows.

[2] "Satisfied" refers to the whole process of model assessment present in Section 4. Outperforming the dummy classifier would be sufficient, though in practice we tried different models each time attempting to improve results that our engineered features could reach.

3.2 Feature engineering

Computing usable features is required. Here's a summary table to define them (**important note**: each value in the list is calculated for both the home team and the away team, for a total of 14 predictors):
1) Elo Rating Measure of a team's skill level widely used in zero-sum games like basketball [**?**]. We used this formulation from Silver et al. [**?**], it takes into account both margin of victory and homecourt advantage (common phenomenon for which multiple factors combine to increase the likelihood of the home team winning. [**?**]). :

- Initialize to 1500 the Elo rating of each team.
- $\mathbf{Elo\_diff = Winner\_Elo - Loser\_Elo + (-1)^{1-y} \cdot 100}$ [$y = 1$ if home team is the winner, $y = 0$ otherwise. The term $(-1)^{1-y} \cdot 100$ is used to consider homecourt advantage in calculating Elo ratings, giving +100 points of advantage to home team]
- $\mathbf{mvm} = \frac{(\boldsymbol{Winner\_PTS - Loser\_PTS + 3})^{\mathbf{0.8}}}{\mathbf{7.5 + 0.006 * Elo\_diff}}$ [m.v.m. stands for "margin of victory multiplier", it takes into account the margin of victory with which a team won (or lost if negative) a game evaluating the difference between scored points] [**?**]

- Calculate estimates of probabilities for the actual winner and loser teams to win before the game [**?**].

$$\begin{cases} \boldsymbol{Pr(Winner\ wins)} = \frac{\mathbf{1}}{\mathbf{1+10^{\frac{-Elo\_diff}{400}}}} \\ \boldsymbol{Pr(Loser\ wins)} = \frac{\mathbf{1}}{\mathbf{1+10^{\frac{Elo\_diff}{400}}}} = \mathbf{1 - Pr(Winner\ wins)} \end{cases} (1)$$

Update Elo ratings (Each will be placed and used in the next game, chronologically, played by the corresponding team) [**?**]:

$$\begin{cases} \boldsymbol{Winner\_Elo\_updated = Winner\_Elo + k \cdot (1 - P(Winner\ wins)) \cdot mvm} \\ \boldsymbol{Loser\_Elo\_updated = Loser\_Elo + k \cdot (0 - P(Loser\ wins)) \cdot mvm} \end{cases}$$

[$\boldsymbol{k = 20}$ [**?**]]
2)WIN_PCT Win Percentage in last N matches. Measures, in percentage, the amount of matches won by each team in the N most recent games they took part in before the current one ($\frac{\boldsymbol{\#Matches\_won\_in\_last\_N\_games}}{\boldsymbol{N}}$).

3)FT_avg_PCT Average of last N matches' Field Throws percentage. For each game, in $\boldsymbol{X}$, there is a value for free throws percentage ($\frac{\boldsymbol{\#Free\_Throws\_made}}{\boldsymbol{\#Free\_Throws\_attempted}}$). We considered the mean of these values over the N most recent games they took part in ($\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \frac{\boldsymbol{FT\_PCT_i}}{\boldsymbol{N}}$)).
4)FG_avg_PCT Average of last N matches' Field Goals percentage. Same algorithm used to compute FT_avg_PCT.
5)FG3_avg_PCT Average of last N matches 3 points Field Goals percentage. Same algorithm used to compute FT_avg_PCT.
6)REB_avg Average number of Rebounds in last N matches. For each team

we computed the mean number of rebounds it performed in the most recent N matches it took part in ($\sum_{i=1}^{N} \frac{REB_i}{N}$).
7)AST_avg Average number of Assists in last N matches. For each team we computed the mean number of assists it performed in the most recent N matches it took part in ($\sum_{i=1}^{N} \frac{AST_i}{N}$)

4 Experimental evaluation

4.1 Data

We worked on the "games.csv" datafile which contains team-level results of the games. We calculated predictors starting from the first game of 2003, but used only seasons from 2004 on in phase of learning and prediction.

4.2 Procedure

We compared our accuracy results with the dummy classifier baseline (lower bound: 58%) and with the accuracy upper limit reached so far according to research papers (from 70% to 74% [**?**]). We compared models fitted using $\boldsymbol{N = 10}$ and $\boldsymbol{N = 20}$ when calculating the features defined in section 3 (N refers to the number of preceding games considered). In all the models displayed in Table 1 we used all predictors defined in section 3 except for Elo Rating, inserted as Elo Difference ($\boldsymbol{elo\_difference = home\_team\_elo - away\_team\_elo}$) instead of a pair ($\boldsymbol{home\_team\_elo,\ \ away\_team\_elo}$). For Random Forest we used the deafault parameters ($\boldsymbol{n_{tree} = 500}$ and $\boldsymbol{n_{vars}} = \sqrt{\boldsymbol{p}}$).

4.3 Results and discussion

Results are summarized in Table 1. Regarding the value of $\boldsymbol{N}$, we started with $\boldsymbol{N = 20}$ as suggested by Buursma [**?**], then we tested other values and found the best one for our problem being $\boldsymbol{N = 10}$. Results suggest that a Logistic Regression model performs slightly better than Random Forest w.r.t effectiveness and is significantly more efficient.

$\boldsymbol{N}$	Model	$\boldsymbol{Acc_\mu}$	$\boldsymbol{Acc_\sigma}$	AUC	FNR	FPR	$\boldsymbol{t_l}$ [s]
	Dummy	0.589	0.0001	0.5	0.0	1.0	0.03
10	Random Forest	0.652	0.017	0.683	0.223	0.539	426
	Logistic Regression	**0.663**	**0.016**	**0.699**	**0.196**	**0.539**	**0.94**
20	Random Forest	0.645	0.017	0.678	0.225	0.534	426
	Logistic Regression	0.662	0.016	0.699	0.196	0.541	0.86

Table 1: Results for different values of N and different models