Machine Learning and Evolutionary Robotics Project - Basketball

Nicolò Ermanno Millo[1], Devan Sedmak[2], Riccardo Zavadlal[3], and Matteo Petris[4]

[1,2,3,4] problem statement, solution design, solution development, data gathering, writing

Course of A.A. 2023/2024 - Ingegneria Elettronica e Informatica

1 Problem statement

The aim of this project is to build a tool based on Supervised Machine Learning that can predict the outcome of an NBA match. The goal is to allow the user to predict the winner between a team playing home and a team playing away. We will base the prediction on the teams stats averages in the past 10 games and their win percentage (from the start of the season) after the last game. So our observations are $x \in X = \mathbb{R}^p$ and our responses are $y \in Y = \{0, 1\}$, where $y = 1$ if the home team wins (positive case) and $y = 0$ if the away team wins (negative case). This is a binary classification problem.

2 Assessment and performance indexes

Firstly we split the preprocessed dataset (see 4.1) according to a random train-test division, 80% for the training and 20% for the test. Secondly we scale the learning set using standardization. We then scale each datapoint in the test set, using the mean and the standard deviation of the learning set.
We choose a learning technique and then use the grid search on the learning set. The grid search splits the learning set with 5-fold cross-validation and computes the average accuracy for all tuples of parameters in the grid. The grid search returns the best parameters (according to the average accuracy). We then learn a model on the entire learning set, using the best parameters. To check if there is no overfitting we apply the model on the learning set and we compute the training accuracy. Now we use the model on the test set (prediction) and we compute the test accuracy and other performance indexes related to binary classification such as FPR ($\tau = 0.5$), FNR ($\tau = 0.5$) and EER. We also plot the Confusion matrix and the ROC curve, evaluating the AUC. We then repeat this process for all the chosen learning techniques.

3 Proposed solution

The solution we propose is to use 3 different supervised learning techniques to build the model that will be used for the predictions. The supervised learning techniques chosen are: Random forest, SVM, Gaussian Naive Bayes. Our baseline is the dummy classifier, which its prediction is always the most frequent case, between winning and losing at home.

Once the model is trained, the tool is ready to make its predictions. Before the match, a user can insert as inputs these features: the stats averages of the home and away teams in the last 10 games and their win percentage from the start of the season till the last match played. The tool will predict if the home team wins ($y = 1$) or loses ($y = 0$).

4 Experimental evaluation

4.1 Data

We use the data collected in games.csv and ranking.csv[1], choosing just the NBA games from 2013-2014 season to 2018-19. Usually the season begins in October and ends in mid-April. But because we need the statistics of the 10 games play at home and 10 ten games played away before a match, we consider just the matches that occurred during the period from middle of December to mid-April.

To make our prediction we need to consider the information that we have before the start of the match. So the features that we think can be used to predict the outcome are the stats averages of the home and away teams in the last 10 games prior to the their match. In particular we consider: Number of points scored, Field Goal Percentage, Free Throw Percentage, Three Point Percentage, Assists, Rebounds. Furthermore we use the winning percentage of the teams in the season till the previous match, that during our evaluation, firstly not used and then added, came out that was the most significant feature, improving models accuracy by approximately 4%.

During the phase of preprocessing of the data we create a dataset in this way. We consider a match between two teams. We compute the stats averages of the last 10 games played at home for the home team and stats averages of the last 10 games played away for the away team. In this way we can take into account the likely advantage of the home team (as we expect from our domain knowledge). Then we consider the winning percentage of the teams (given by ranking.csv). Finally we associate all these features with the result of the match. Moreover, in order to apply the Naive Bayes, we assume feature independency, even if in reality this is not true, e.g field goal and three points percentage are correlated.

As said we keep the matches home and the matches away apart, because we suppose a correlation between playing home and winning the game. As a matter of fact after some data exploration, we noticed that in the 60% of cases the home team wins, as expected.

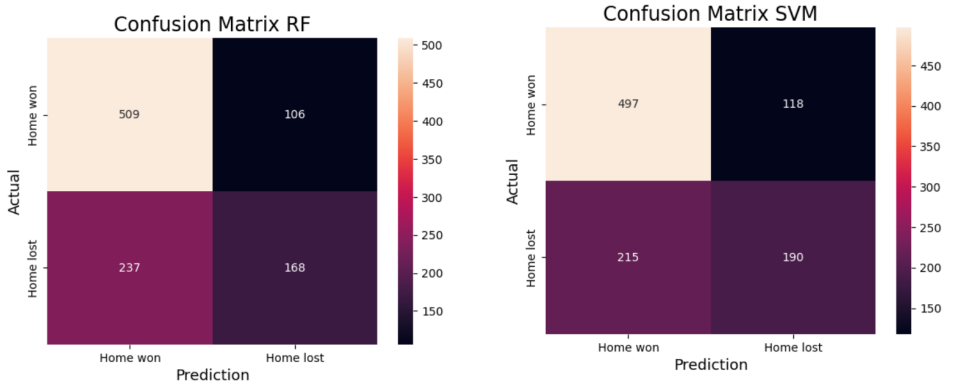
4.2 Procedure

We define the sets of possible parameters of the grid search for the considered learning techniques as follows.
For the random forest technique we try different settings, in particular:
$n_{tree} \in \{100, 200, 300, 400, 500\}$, $max\,depth \in \{5, 10, 15, 20\}$, $node\,impurity \in \{entropy, gini\}$.
For the SVM we try: $c \in \{0.1, 1, 10\}$, $kernel\,function = rbf$, $\gamma = \frac{1}{n_{features}}$.
For Gaussian Naive Bayes we try: $var\,smoothing \in \{10^{-9}, 10^{-8}, 10^{-7}\}$.

Best parameters chosen by the grid search for each learning technique: Random Forest: $impurity = entropy$, $max\,depth = 5$, $n_{tree} = 500$; SVM: c= 1, $\gamma = \frac{1}{n_{features}}$, $kernel\,function = rbf$; Gaussian Naive Bayes: $var\,smoothing = 10^{-9}$. Selecting the best parameters we obtain the following results:

Method	**Training Accuracy**	**Test Accuracy**	**FPR**	**FNR**	**EER (Threshold)**
Random Forest	0.70	0.66	0.59	0.17	0.36 (0.59)
SVM	0.70	0.67	0.53	0.19	0.36 (0.63)
Naive Bayes	0.70	0.64	0.45	0.29	0.36 (0.57)
Dummy	0.59	0.60	1.00	0.00	–

Table 1: Performance indexes for the chosen learning techniques



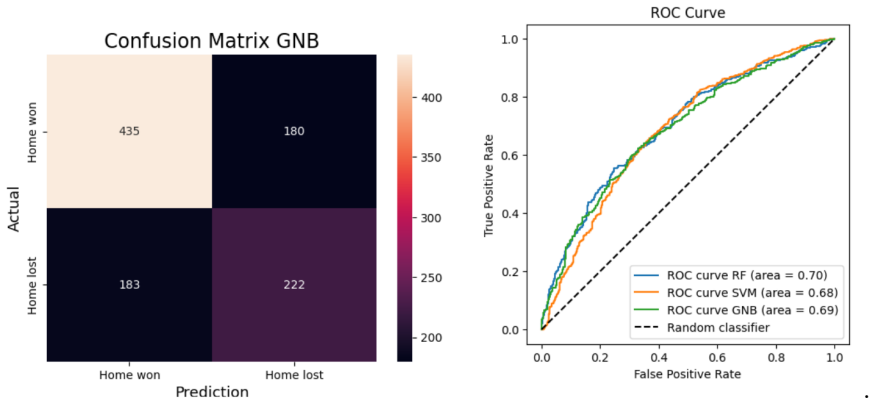


Figure 1: Confusion matrices and ROC curves for the chosen learning techniques

4.3 Results and discussion

Clearly identifying a best model is a challenging process. This difficulty arises from the fact that intrinsic effectiveness indexes fail to highlight an overall best choice. Therefore, we turn to an extrinsic analysis of results within a specific context.

Games prediction class of problems generally favors models with lower test errors, such as SVM ($Err = 0.33$), although gap to others is minimal (Tab. 1).

It is also crucial to take into account the fact that the dataset has proved to be strongly subjected to seasonal variations and broader shifts in play style, as well as major disruptions, like the pandemic period. Consequently, preference should not be considered unconditional.

Anyway, what emerges with clarity is that all employed techniques show varying degrees of predictive power, surpassing the baseline represented by the dummy classifier ($Acc = 0.60$) and likely avoiding overfitting. The computed accuracies align with a deeper examination of the same problem ($Acc_{max} = 0.65$)[2].

It should also be added that none of the models demonstrate excellent performance in the given task, as evidenced by non-negligible differences between the training and test errors across all of them and, even worse, by biased confusion matrices (Fig. 1). These effects are probably caused by the unbalanced dataset.

References

[1] Nathan Lauga, *NBA games data*: https://www.kaggle.com/datasets/nathanlauga/nba-games/data

[2] Jasper Lin, Logan Short, and Vishnu Sundaresan. *Predicting National Basketball Association Winners*. https://cs229.stanford.edu/proj2014