

Machine Learning and Data Mining project: Predicting the outcome of NBA games using home statistics

Bruno Bonaiuto Bolivar, Carlo De Nardin, Fatma Sayed Mahmoud
Sayed Moustafa

Course of AA 2021-2022 - DSSC

Abstract

Basketball is one of the most popular sports in the US and the world. Many sponsors invest their money in teams and players. With the enormous amount of data that exists, the task of predicting the winning team of an NBA game based on different statistics have interested many researchers in the field of machine learning. Also many fans have high hopes when their team plays at home and would even bet that their team will win given the “land” factor. In the course of the last few years it was found that different home statistics had an important impact on why a home team won with a 60 % probability [1].

1 Problem statement

The purpose of this project is to predict the winning team of an NBA game based on the match statistics of the home team. The home team is the team with the land factor since it plays in the home stadium with many fans. Instead the away team often comes from afar and is supported by few fans during the game. The input variables are samples composed of different game statistics: ranking, field goal percentage (FGP), field goal three point percentage (FG3P), free throws made (FTM), rebounds (REB), assists (AST), steals (STL) and the blocks (BLK). The output is a binary result {YES, NO} that specifies if the home team wins or not.

2 Assessment and performance indexes

Once we build our models using different techniques, it is crucial to evaluate their performance and check if the predictions made were good or bad. The task of selecting the appropriate performance metrics to evaluate the models we built should suit the problem and the data. We are solving a **binary classification** problem and our dataset is slightly balanced with {YES = 3845, NO = 2715}. The indexes should take into consideration that the cost of false positives and false negatives are not different. In our problem, a wrong prediction in any way means, for example,

losing betting money or change in sponsorship deals which are both costly. The metrics that could be considered to evaluate our algorithms are:

- **Confusion matrix:** the positive class definition, in our analysis, is: home team not winning, since it is the class with the least values
- **Accuracy:** intuitive performance metric and very useful one when the dataset is balanced or slightly balanced
- **AUC:** area under ROC that measure the classifier performance

We did not consider the sensitivity or specificity because for our problem a false prediction is costly whether it is a false positive or false negative.

3 Proposed solution

Since it is a binary classification problem, different models have been proposed:

- **Trees:** Trees [2] are one of the most common solutions in a binary classification problem. They are fast to learn, readable and also intuitive.
- **SVM:** Support Vector Machines [3] offer a high flexibility since it uses the kernel functions to separate the learning observations.
- **KNN:** K-Nearest Neighbor [4] captures similarity to predict new data points based on the distance. It is useful when it is more important to obtain a high accuracy model instead a human-readable model

4 Experimental evaluation

4.1 Data

In this project, 5 NBA seasons, taken from a Kaggle dataset [5], have been considered (2014 - 2018) since from 2019 to 2021 the home team winning probability decreases due to the covid-19 pandemic. The dataset used is composed from a selection of variables combining 3 different datasets: games, games details and rankings. This allowed for a more broad overview of the impact of different statistics on our problem. Only the home team statistics were considered for the prediction of home team winning to avoid any correlation between the variables. The dataset was split in two parts randomly by setting a seed to perform reproducible splits: train set (80 %) and test set (20 %).

4.2 Procedure

Different models have been used to solve the problem, as described in the proposed solution.

4.2.1 Single tree and trees aggregation

The first model that was considered was a simple unpruned tree with the highest complexity possible. The best complexity parameter was taken to build a pruned tree and to find the best single tree.

Two different aggregation techniques were applied: bagging and random forest. For the random forest, the parameter of variables taken for each tree was chosen by default ($m = \sqrt{\text{number of variables}}$).

4.2.2 Support Vector Machines

Support Vector Machines might be a less interpretable technique than trees; however, they are highly flexible. The train set, with all the variables included, was fitted with the default parameters: C-classification, radial kernel and cost = 1.

4.2.3 KNN

The data were normalized to compute the K-Nearest Neighbor. To obtain the best K a sequence was done from a range of the Euclidean formula.

4.3 Results and discussion

The models' results were obtained by training classifiers with the training data. For the prediction phase the test data were used.

4.3.1 Single tree and trees aggregation

The unpruned single tree resulted in a tree with high complexity. By pruning the tree a better result was obtained.

The results obtained with the bagging approach and the random forest approach were better than a single tree. The most important part was selecting the best number of trees to use to get the optimal classifier. By training the trees several times, through the use of test error and oob error, an approximate number of 80 to 100 trees were selected.

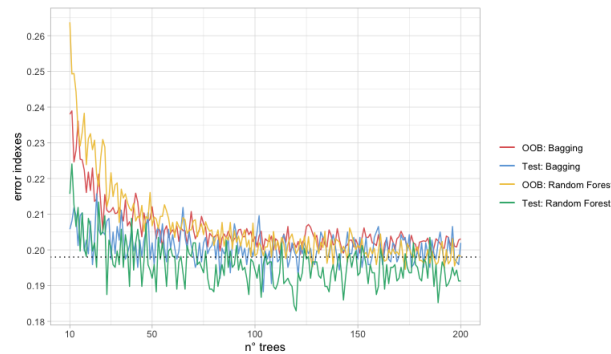


Figure 1: test error and OOB error for different number of trees

4.3.2 Support Vector Machines

Cross Validation with 10 folds was done for different SVM kernels: linear, polynomial, radial and sigmoid and then the CV error was plotted against the cost. The CV error for the radial kernel decreases dramatically initially but tends to increase after a few iterations. The other kernels show a non-uniform pattern. These results are maybe due to the numbers of support vector which is 2433. High number of support vectors means that we have high tolerance, low variance and high bias.

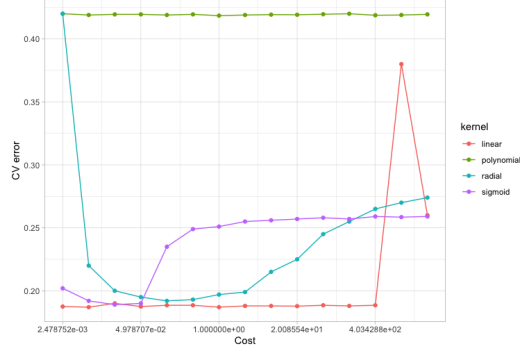


Figure 2: CV error against different values of cost for different kernels

4.3.3 K-Nearest Neighbor

The results obtained with the K-Nearest Neighbor methods by computing the best k across the test set was k equals to 65, which is a local estimate respecting the number of observation.

4.3.4 Results and conclusion

For the different indexes discussed in the assessment the results are shown below.

| Index | Single tree | Pruned tree | Bagging | RF | SVM | KNN |
|----------|-------------|-------------|---------|-------|-------|-------|
| Accuracy | 0.719 | 0.779 | 0.8 | 0.808 | 0.817 | 0.811 |
| AUC | 0.79 | 0.767 | 0.788 | 0.797 | 0.805 | 0.801 |

In general, the results show that although we get a better accuracy and a higher AUC for the SVM it remains computationally complex. We have 2433 support vectors out of the 5248 observations. Thus, it is not the best solution for this problem with this dataset. The results obtained with the tree models reflect the general theory of these models. The random forest has turned out to be one of the best techniques. Also, the bagging technique has shown results similar to the random forest, so it means that the variables chosen in the exploratory phase were strongly independent. The results obtained from K-Nearest Neighborhood was good although the computational effort was high with a large dataset.

For this project each member participated to the common tasks. Then each student studied and analyzed the behaviour of a model chosen from the proposed solution:

- **Tree, bagging and random forest:** Carlo De Nardin
- **Support Vector Machines:** Fatma Sayed Mahmoud Sayed Moustafa
- **K-Nearest Neighbor:** Bruno Bonaiuto Bolivar

Finally a GitHub repository has been made for this project [6].

References

- [1] J. Duan. Predicting the outcome of nba games using machine learning. <https://medium.com/nerd-for-tech/predicting-the-outcome-of-nba-games-using-machine-learning-676a62549040>.
- [2] Gareth ed Al James. *An introduction to statistical learning*. Springer, 2013.
- [3] S. Jain and H. Kaur. *Machine learning approaches to predict basketball game outcome*. 2017.
- [4] B. Lantz. *Machine learning with R*. 2013.
- [5] N. Lauga. Nba games data. <https://www.kaggle.com/nathanlauga/nba-games>.
- [6] Bonaiuto Bolivar De Nardin and Moustafa. Predicting the outcome of nba games repository. <https://github.com/carlodenardin/ML-predicting-the-outcome-of-NBA-games>.