University of St.Gallen

Data Science Fundamentals Program

Predicting the Success of NBA Teams during the Playoff Season based on their Statistics during the Regular Season

Marc Oppliger (21-612-361)

Mica Brutschin (21-622-485)

Céline Tschirky (21-610-480)

supervised by

Prof. Dr. Johannes Binswanger
Prof. Dr. Lyudmila Grigoryeva
Jonathan Chassot

December 4, 2022

Contents

List of Figures

List of Tables

1 Introduction

Nowadays, data science and machine learning play a central role in almost all industries and sectors — including the sports betting industry. The large financial transactions in sports betting underline the growing importance that machine learning algorithms faced in recent years. Basketball, especially the national basketball association (NBA) of the United States, is one of the most watched and discussed sports in the world which attracts bettors and millions of fans on a global scale (Thabtah et al., 2019, p.104).

This machine learning project focuses on predicting the success of an NBA team in the playoffs. We simplify this problem into five subproblems and evaluate separately whether a team makes it into the first or second round of the playoffs, whether they get to the conference finals, the league finals or if they win the championship. This allows us to treat the problem as a binary classification problem. The goal is to train a machine learning model to predict with the highest possible accuracy whether a team will participate and win in the respective playoff rounds.

The prediction is based on the data of the last 43 NBA seasons, since the last major changes in regulation were implemented in season 1979/1980. We use different variables for our project, such as field goals per game, rebounds per game, points per game, and 24 other independent variables. After preprocessing our dataset, with PCA and SMOTE, it contains ten different independent variables, five dependent variables, and a total of 1164 observations.

We aim to build a concise algorithm and therefore build a variety of different models and compare their validation scores. We first build a logistic regression model, before working with more advanced machine learning models, such as random forests and boosted trees. Furthermore, we try to further optimize our results with k-nearest neighbors models and neural networks. The models with the highest accuracy for each subproblem are then chosen as our final models and will be tested on our test dataset.

2 Data Preprocessing

Our analysis is based on the data, that is available on the Website *Sports-Reference*. There we found different variables for each team and for each season. We decided to work with the "per game stats" and the "advanced stats", which we downloaded for each season since 1979. In 1979 the NBA introduced the three-point line into the game of basketball, and this led to big differences in the game and stats. We added information about the success of a team in the playoffs to our datasets manually, before we used a for loop to first merge the datasets for per game stats and advanced stats of each season. Afterwards, a second for loop appended each dataset to the first one, so that we were able to work on the data properly.

Since there is generally a large circle of interested parties, including fans, sports analysts and representatives of the betting industry, the data is rather easy to gather, and we have faced almost no missing values. Nevertheless, we first had to do some data cleaning to ensure that it fits the models. We handled the two columns containing missing values by dropping them, in order to ensure that we do not lose any observations. Afterwards, we created new columns to ensure that we have no absolute values but all in relation to the

number of games a team has played. This is important, because the seasons have been shorter during the pandemic, and this could have disturbed our analysis. Furthermore, it allows us to catch all interactions in the data properly, and to enable our analysis accordingly.

Finally, the cleaned dataset had 27 dimensions. To reduce the number of dimensions, we performed principal component analysis (PCA). On the one hand, reducing the number of dimensions using PCA helps the algorithms to perform better due to fewer inputs. On the other hand, PCA also reduces noise, and this improves the accuracy of the predictions. To perform PCA, we first dropped our target values, the columns containing the playoff data. Second, we evaluated how many principal components we need to explain 85% of the variance in our data. Finally, we scaled the data and ran the PCA with ten components. The following plots visualize the first three principal components for each subproblem.

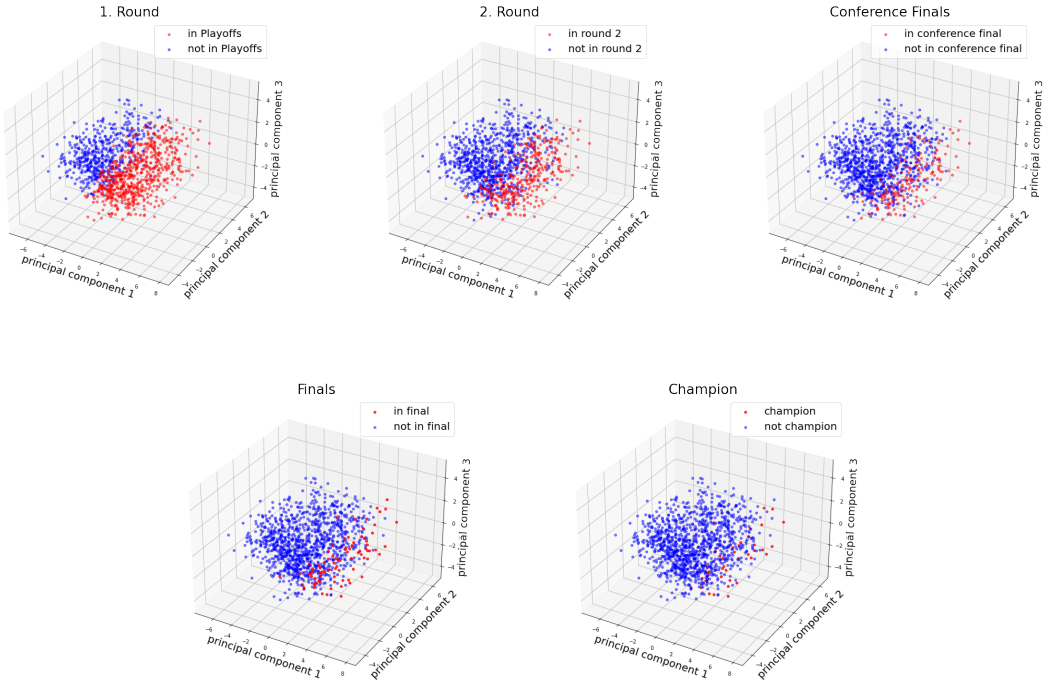

Figure 1: Pricipal Components for each Target Variable

Looking at the Figure 1, we can see the distribution of our data points and the challenges our models face when trying to learn the pattern of our data. In addition, the plots above show a significant imbalance between classes in our data. Since the NBA Championship is an elimination tournament, there is a natural class imbalance between teams that do not participate in the playoffs and teams that do advance to the playoffs.

At first glance it appears that predictive models work well for data with class imbalances. However, this is only due to the disproportionately good prediction of the majority class. We address this class imbalance problem with the data sampling method synthetic minority oversampling (SMOTE). SMOTE creates minority instances, similar to the preexisting minority instances synthetically. SMOTE is a type of oversampling and consistently

outperforms other methods (Last et al. 2017, p.16). In addition to oversampling, undersampling would have been an alternative method. We did not consider undersampling in our project, mainly because it would have reduced the majority class to the amount of the minority class, potentially losing important data.

One of our main challenges was to prevent any data leakage from happening during our process. This aspect is particularly important for the implementation of SMOTE, since we had to ensure that we do not validate our model on the synthetic sample based on the original training data. We faced a similar problem while cross-validating. Furthermore, data leakage appeared when we standardized our data based on the whole dataset instead of the subset used for the cross-validation. Therefore, we had to introduce pipelines as a new concept to our code. Pipelines allow to assemble several steps. We united all steps that could lead to a data leakage into our pipeline and used it first to find the optimal models and second to cross-validate them.

3 Models

We decided to work with five different models from logistic regression to neural networks to find the optimal algorithm to solve our problem. First, we built these models and tuned their parameters with the grid search function to optimize them. Afterwards, we evaluated the performance of our model with cross-validation.

The logistic regression model is characterized by the assumption that the relationship between the explanatory variables (i.e., an NBA team's season statistics) and the conditional probability is given by a logistic function. As the logistic regression model is a rather simple model, that is well suited for classification problems, we have implemented it as our first prediction mechanism. As the standard logistic regression model is almost too simple for a serious classification experiment, we have implemented a polynomial logistic regression with the optimal polynomial degree instead.

Random forests are a machine learning model based on decision trees. A random forest is the aggregation of several decision trees, which are trained on different, randomly generated subsets, so called bags, of our training data. The prediction of the random forest is the mean of those aggregated decision trees. To choose the optimal random forest, we optimized the two important parameters, the number of decision trees our random forest grows and the learning rate.

A boosted tree is based on several decision trees (weak learners), which themselves are based on subsets of our data. The more often a data point is misclassified, the higher is its probability to get selected in the next subset. Therefore, each decision tree improves the performance of the boosted tree by improving the performance on the mistakes of its predecessor. Together these decision trees form a strong and robust learner, whose accuracy is improved by each decision tree. This procedure is called gradient boosting. We evaluated the optimal model of our gradient booster, by tuning on the number of decision trees as well as the learning rate.

A neural network tries to recognize underlying relationships in our data. Therefore, it tries to mimic the process of a human brain and is a system based on neurons. The neural network consists of an input, several hidden and an output layer that comprises a certain number of neurons. Each neuron is a mathematical function that evaluates the collected

information and then classifies it via an activation function. Our neural network only consists of one hidden layer and is a feed-forward neural network. We optimized different parameters, such as the number of neurons in our hidden layer, the learning rate, and the maximal number of epochs.

The k-nearest neighbors (KNN) algorithm is based on the concept of proximity. In order to determine which data points are closest to a given data point, the distance between the query point and the other data points must be calculated. The nearest data points then allow us to classify the new data points. We optimized the number of neighbors that are used for our classification, as well as whether they are weighted according to their proximity.

4 Discussion

After building a variety of different models to find the optimal algorithm for our problem, we can now compare and analyze our results. It is important to analyze each subproblem individually, since each problem has its own challenges. The following table shows the accuracy of our models for each stage of the tournament.

	Target Variables				
	Round 1	**Round 2**	**Conf. Finals**	**Finals**	**Champion**
Polynominal Logistic Regression	91.14%	83.83%	86.02%	87.30%	90.63%
Random Forest	88.95%	82.67%	86.14%	90.51%	94.48%
Boosted Tree	89.34%	82.67%	86.14%	88.71%	93.84%
Neural Network	89.09%	82.54%	86.01%	91.79%	95.12%
K-Nearest-Neighbors	89.35%	82.29%	81.13%	87.30%	90.76%
Dumb Model	53.33%	73.33%	86.68%	93.33%	96.66%
	pred 1	pred 0	pred 0	pred 0	pred 0

Table 1: Accuracy Scores of different Models in Cross-Validation

Two trends in our results can be seen in Table 1. First, the further we go in the tournament, the more random the success of a team tends to be and the higher is the probability of a successful prediction by simply predicting a 0 in each case. Second, we see that our models perform worse in predicting the winner and finals than a dumb model that simply predicts no one to win. However, this is because only a few teams per season make it to the final stages of the tournament, and there is a huge class imbalance in our dataset. SMOTE has allowed us to build models that can handle this class imbalance and predict both classes.

In order to successfully compare our models to the silent models shown in Table 1, we need to refer to several metrics. For example, the recall value shows us the correctly predicted champions relative to all predicted champions. A dumb model would score zero because it would predict no champion. However, our final champion prediction model

has a recall value on the test set of 0.16, meaning that it correctly predicts every fifth or sixth champion. The following table lists all the accuracy and recall values for the test set.

	Target Variables				
	Round 1	**Round 2**	**Conf. Final**	**Final**	**Champion**
Model	Logistic Regression	Logistic Regression	Random Forest	Neural Network	Neural Network
Accuracy	0.92	0.84	0.85	0.91	0.95
Recall of Prediction	0.93	0.89	0.78	0.31	0.15

Table 2: Accuracy and Recall Score of Final Models on Test Set

After evaluating our final models, Table 2 suggests that it is very difficult for our algorithms to correctly predict the success of an NBA team in the playoffs. In addition, we find that the accuracy in the validation set is somewhat different from the score in the test set. This is due to two different factors: the reducible error and the irreducible error.

The reducible error means that we could further optimize our algorithms by using different methods and variables. The NBA has a special organization, the thirty teams play in two different conferences which have their own knock-out stages. This further hinders predictions of the playoffs based on the stats during the season. Our prediction deliberately focuses on the performance of the teams during the current season. Other important variables that could improve predictions, such as the value of the squad, success in recent years, or injuries in the playoffs, are not considered. Therefore, an approach with different variables could lead to more accurate predictions.

The second factor that makes our prediction difficult is that there is a lot of noise in our problems. The performance during the season does not always correlate with the performance in the playoffs. This difference is called noise and is significant in our project. That means that there is a lot of randomness involved about who advances to the next round and who wins the championship. Therefore, we find a relatively high irreducible error in our problems. We can observe this phenomenon by comparing our score to predict who will make it to the playoffs and who will make it to the second round. There can be seen that the accuracy decreases significantly, because the randomness in this subproblem is much higher than in the subproblem that is about who will make it into the playoffs, what is based on more than 32 games.

A recent study of Kirasich at al. (2018, p.22) showed that logistic regression models can outperform decision-tree-based algorithms, due to the fact that the logistic regression looks at the simultaneous effects of all variables together. That further underlines our assumption that our data is relatively noisy, and that other parameters may lead to a better success.

Furthermore, our analysis shows that our neural networks are outperformed by our logistic regression in some cases. This is quite astonishing, because a neural network without any

hidden layers can be compared to a logistic regression and therefore a more complex neural network should outperform a logistic regression. Jiao et al. (2020, p.3729) conclude in their research that a logistic regression can outperform a neural network in a problem that contains a small number of data samples.

Finally, we observe that tree-based models are able to outperform neural networks for some problems. This is since neural networks still struggle with the learning of non-smooth patterns where tree-based models are successful (Grinsztajn et al., 2022, p.8).

5 Conclusions

The ever-growing betting industry is highly interested in accurate machine learning algorithms that help to predict the playoffs. The objective of this project was to develop such an algorithm. In the process some problems have been encountered: It is difficult to determine the right variables, there is a lot of noise in the problem involved, the organization of the league has a big impact on the outcome, and it is challenging to find a data sample big enough for the variables. All these problems make it difficult to determine a good learning process.

By dividing our problem into five subproblems, we succeeded in building an algorithm that tackles these challenges and predicts the success of a team based on its performance during the regular season. We built and optimized five different models and validated them for each subproblem. The best model for each subproblem has then been tested on the test dataset to determine its final performance.

The results suggest the difficulty to predict the outcomes of the playoffs in the NBA. All algorithms predicted similar values and trends for the different rounds. The randomness increases massively throughout the tournament, but the accuracy of the models increases because they can predict the zeros more easily. Furthermore, we conclude that there is no optimal algorithm to tackle such problems. We need to optimize different approaches and compare their outcomes. We have seen that there are problems that are well suited for tree-based models, and that neural network or logistic regressions can outperform other models due to the structure of a problem.

The performance of our models is disappointing. While we can predict the participants of the playoffs quite well, we are not able to predict the outcomes of the playoffs reliably. On the one hand, we should further optimize our algorithms. On the other hand, there is a lot of randomness included in sports tournaments, especially in knock-out stages that make predictions inaccurate.

6 References

Grinsztajn, L., Oyallon, E., Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data?. https://doi.org/10.48550/arXiv.2207.08815

Jiao, S., Gao, Y., Feng, J., Lei, T. Yuan, X. (2020) Does deep learning always outperform simple linear regression in optical imaging?. Opt. Express. 28, 3717-3731. https://doi.org/10.1364/OE.382319

Kirasich, K., Smith, T. Sadler, B. (2018). Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. SMU Data Science Review. 1, 3, Art 9. https://scholar.smu.edu/datasciencereview/vol1/iss3/9

Last, F., Douzas, G. Bacao, F. (2017). Oversampling for Imbalanced Learning Based on K-Means and SMOTE. https://doi.org/10.48550/arXiv.1711.00837

Thabtah, F., Zhang, L. Abdelhamid, N. (2019). NBA Game Result Prediction Using Feature Analysis and Machine Learning. Ann. Data. Sci., 6, 103–116. https://doi.org/10.1007/s40745-018-00189-x