Predicting basketball injuries using ML

M203 Alternative finance project

Soughati Kenza, Henry-Biabaud Briac, Collin Thibault

1 Introduction

The aim of this project was to work around machine learning models to predict *NBA* basketball player injuries for the next game-ahead, using a collection of box score individual statistics from previous games.

We believe that there is a two-fold **financial** interest in this study. Indeed, most betting venues do not cancel bets on players that end up getting injured mid-game, and being able to predict a high probability of injury for a player would help us refine our betting strategy. Moreover, acting as an endorser or a general manager, predicting injuries could help with longer-term investment decisions, as we would not want to sponsor or trade for a player with a higher-than-normal propensity for injuries.

A third **sport**-related benefit would be from the point of view of a coach, who could decide to rest current high-risk players based on our indicator. Our production may finally be useful for the medical staff and the players themselves, to monitor and work more thoroughly on precise body areas of concern.

We will therefore implement various techniques to try and obtain a sufficiently correct predictive model. We found that using a combination of outliers handlers and recursive feature selection within a neural net does a correct job at predicting injuries. We finally take advantage of our results to build a visual betting indicator that exhibits a straightforward investment strategy.

2 Data Gathering and Processing

The extraction and construction of clean datasets was our work entirely. We scrapped all individual player statistics from [SH24] and used a compilation of comprehensive and official injury reports from [Pro24]. We decided to extract eleven NBA seasons worth of individual stat lines, starting from 2012-13.

We put together in a single table the entirety of the in-game statistics from any player registered on a team for any of the selected seasons. These statistics cover the broad aspects of basketball, from the efficiency in offense to the ability of correctly protecting the rim and perimeter. To standardize those statistics without excessively generalizing, we divided all statistics by the season average produced by the player. Thus, a given statistic relates to how an individual does in that precise area compared to his usual stance, which is being reset every year. A league-wise comparison would heavily distort the data and might greatly hurt the future predictive ability of our algorithms.

To account for the fact that injuries can be developed throughout a lengthy period, we must consider the presence of trends in the data, whether they be clear or hidden. To adress this issue, we also account for the stats line from the last game, as well as the averages from the last three and last five games. This is applied to any statistic for which studying a trend is relevant.

Statistic	**Meaning**
FG%	Field goal percentage
MP	Total minutes played
TOV	Number of turnovers
+/-	Score differential while on court
3P%	3-Point shots percentage
GS	Game started or not

Table 1: Some game-related used statistics

The complete code for this extraction will be attached to this paper under *extraction.ipynb*, as well as the cleaned saved statistics tables. Within the code, comments are written along the way when judged necessary.

3 Literature Review

To do..

4 Theoretical Foreword

This section will be dedicated to giving theoretical foundations to the various methods and algorithms experimented. We implemented several techniques to deal with the various issues of our data. The final machine learning pipeline can be decomposed into the following three parts:

- *Data pre-processing* which will include handling the inherent class imbalance between games that led to an injury and those that did not. We will both perform class **resampling** and custom class **weighting**.
- *Features engineering and selection* which will allow the reduction of the **dimensionality** of our dataset. In other cases we would also produce feature standardization, but this was performed during the extraction.

- *Model selection* where we will construct our machine learning model with carefully finetuned parameters. We will implement artificial **neural nets** for they have built-in support for handling imbalanced data.

4.1 Data Pre-processing

Because our dataset is heavily imbalanced in favor of the games that did not led to an injury, we need to rebalance. Undersampling the majority class helps in preventing the model from being biased towards the majority. This technique must be used with caution, as it may lead to the loss important information if the discarded data contains valuable patterns not present in the retained subset.

4.1.1 Cluster-based resampling

Cluster-based undersampling is a technique that involves identifying clusters within the majority class and then sampling these clusters to reduce their size. This approach aims at retaining a representative subset of the majority class while preserving diversity, and will be part of our experimentation. While yielding satisfying results, the added gain from this method does not justify the extreme cost of computational time. Indeed, this model might identify a large number of clusters, each needing to be processed separately. Moreover, resampling involves repeatedly drawing samples from each cluster, which adds to the computation time, especially if the resampling method is complex.

4.1.2 Synthetic minority oversampling

We will also try doing the opposite, by generating synthetic examples of the minority class instead. This method is called *synthetic minority over-sampling* and is directly implemented in the machine pipeline. It is usually complemented by the use of *custom class weights*. This approach modifies the learning behavior, by forcing it to pay more attention to the minority class. This mechanism is interesting, but we decide to push this idea further by using an extension of this technique.

4.1.3 Adaptive synthetic sampling

This method also generates synthetic examples for the minority class, but goes further by focusing on generating synthetic data for subsets that are harder to classify, rather than uniformly for the entire set. It does so by computing a weighted distribution for different minority class samples based on their learning difficulty, prioritizing those that are harder to learn. Given the high dispersion in the different game stat-lines, this method appears well-suited and will be used going forward.

4.2 Feature Engineering and Selection

Given we have over 80 features for each player's game performance, dimensionality reduction is essential to simplify the model, reduce overfitting and potentially uncover more generalized patterns in the data.

4.2.1 Principal component analysis

We have first tried using *principal component analysis* to this end. This technique reduces dimensionality by transforming the original variables into a set of uncorrelated principal components that account for the most variance in the data. It therefore allows us to choose a threshold of final entropy we would like to explain with our features. This method however does not yield satisfying results, which can be due to two reasons mainly.

PCA works by selecting the directions that maximize variance, while not necessarily maximizing the separation between classes. In imbalanced sets, the variance is often dominated by the majority class, so the principal components may not capture the nuances necessary to distinguish the minority. Moreover, the assumption that all features are linearly linked is rather strong, and could be further aggravated given the imbalance of the set.

4.2.2 Recursive feature elimination

We deviate from principal component analysis and orientate our focus on *RFE*, which iteratively removes features and stops when the marginal loss of losing the next feature becomes too harmful for the model. We found this strategy to be beneficial given our set, essentially because the strategy is *supervised*, meaning the algorithm learns from a teacher from known examples to make predictions or decisions about unseen data

RFE therefore considers the target variable when selecting features. This is crucial for imbalanced datasets, where the minority class is of interest. It preserves discriminative features for the minority class by evaluating predictive accuracy. In contrast, PCA, an unsupervised method, overlooks target relevance and may produce components less useful for prediction, potentially missing discriminative features for the minority class.

4.3 Model Selection

Given the challenging nature of the data at hand, several algorithms were implemented through a methodology of trial and error. We briefly present them theoretically below, alongside our findings related to their use in our research.

4.3.1 Gradient boosting machines

These machines are a family of ensemble machine learning algorithms that build models in a stage-wise fashion. They work by sequentially combining and averaging simple models, such as decision trees, to create a stronger prediction.

These models start with an initial prediction, usually being the log odds of the target variable for classification tasks. Then at each step, a new decision tree is added that predicts the residual errors made by the previous trees. Gradient descent is used to minimize the loss when adding these trees. A loss function measures the difference between actual and predicted values, and the gradient descent algorithm is used to find the direction in which adding a new tree will most reduce the loss.

These machines are usually accurate and flexible, but also inherently perform their own feature selection at each node of each tree, which is beneficial for high-dimensional data. They are however complex to manipulate and interpret, with the sequential nature of boosting leading to longer training times.

4.3.2 Deep learning and neural networks

Deep learning can be effective in dealing with classification tasks involving highly imbalanced classes, but it requires careful handling and additional techniques to optimize performance. Imbalanced datasets pose a challenge because models might become biased towards the majority class, leading to poor generalization for the minority class.

However, an enhanced fine-tuning alongside the use of *focal loss* allows us to obtain better results than with any other combination of methods. This is therefore the setup we will use going forward.

5 Implementation results

The pipeline chosen is precisely described in the notebook, but consists in using a combination of *PCA* and *isolation forest*, alongside a *neural net* with data *resampling* and *focal loss*. We present our results and further apply the model to advise users on their betting strategy.

5.1 Model results

We ran the following sequential algorithm, for which we tuned the hyperparameters thanks to a multidimensional *grid search*:
We obtained the following results for both the validation and test sets. We chose not to use the usual *accuracy* metric as it is more appropriate for balanced sets.

Parameter	**Value**
Machine	Neural nets
Optimizer	*Adam*
Hidden layers	4
Neurons	128 to 48
Dropout rate	50%
Learning rate	0.1%
Activation	*ReLu*
Output activation	*Sigmoid*
Kernel init	*He* uniform
Loss	Focal loss
Epochs	15

Table 2: Algorithm hyperparameters

We use *AUC-ROC* instead, which is the *Area under receiving operating characteristic* curve. This is a performance measure commonly used when dealing with highly imbalanced classes.

Parameter	**Value**
Validation set	0.7113
Test set	0.7095

Table 3: Model quality measurement

A classification model is usually considered as correct when this measure is above 70%, so our results can be viewed as satisfying.

5.2 Betting indications

We further used our neural network on a more recent and unseen *testing* set different than the previous one. This set is composed of two weeks worth of games. We predict the injuries based on those games, and scrap the upcoming ones to give users some indications on games for which betting is not advised.

Name	**Team**	**Next game**	**Date**	**Injury influence**
Ousmane Dieng	OKC	CHO	2023-03-28	Critical
Brandon Ingram	NOP	GSW	2023-03-28	Critical
Day'Ron Sharpe	BRK	HOU	2023-03-29	Significant
Kira Lewis Jr.	NOP	GSW	2023-03-28	Average
Miles McBride	NYK	MIA	2023-03-29	Poor

Table 4: Example of the algorithm's betting indication output

Additionally, we computed for each player a metric to determine his added value to the team at a given point. This metric considers how a player compares to all others in the *training set* in terms of his average $+/-$ and *game score* in his last five games. This is used to rank injuries based on how badly they would influence a team, to further help the user in modifying his betting parlay.

6 Conclusion

Our research was focused on trying to predict injuries in basketball by using machine learning techniques to decipher nonlinear relationships in the data. We collected features relating to how players performed compared to their own averages, and tried to isolate key causes which could lift the veil on an injury in the making.

We implemented an artificial *neural network*, alongside techniques of *recursive feature elimination* and *cluster-based resampling*, and were sufficiently successful in classifying this highly imbalanced dataset to predict basketball player injuries to a certain extent. We then used our model as a helper for betting on individual players or games, the algorithm prompting an injury *watch list* for valuable players at risk of an injury.

7 Remarks

Several strong assumptions were made during this experimentation. The most penalizing was assuming that all injuries are developed, and none result in in-game shocks that are totally unrelated to some players past performances. Some other players might find themselves either faking injuries for diverse reasons, or agreeing to continue playing through a real one, like when Kobe Bryant played and won the *2010 Finals* with a broken index finger.

A more qualitative collection of true injuries would highly benefit our model. We could imagine building an additional *natural language processing* algorithm to analyze the specificities of each injury and address whether they were caused by unrelated sudden shocks or not. Finally, from the standpoint of the medical staff, knowing where an injury might build up can lead to specific exercises used to mitigate those risks, and increase the overall health of players by detecting hidden and potentially harmful trends.

References

[Pro24] Pro Sports Transactions. *Pro Sports Transactions: Basketball.* 2024. URL: `https://www.prosportstransactions.com/basketball/`.

[SH24] Basketball Statistics Sports Reference and History. *Basketball Reference.* 2024. URL: `https://www.basketball-reference.com/`.