Predicting NBA Player Injuries Using Machine Learning

Huairui Wang, Shengkai Yang, Tianze Zheng

December 11, 2024

Abstract

Injuries in professional sports significantly impact players' health, team performance, and economic outcomes. This study applied machine learning techniques to predict injuries for NBA players using a comprehensive dataset that integrates player injury records, physical attributes, and performance metrics. The data underwent rigorous preprocessing and feature engineering. A range of machine learning models, including Random Forest, Logistic Regression, and Multi-Layer Perceptron (MLP), were employed to predict injury types and locations. Despite the imbalanced nature of the dataset, our multi-model approach achieved high predictive accuracy, with most models demonstrating precision and recall between 75% and 98%. This study highlights the potential of machine learning in sports analytics, offering actionable insights for injury prevention and management in the NBA. Future work could focus on integrating additional features, such as player movement data, to enhance prediction accuracy.

Keywords: NBA Player Injuries, Data Preprocessing, Machine Learning, Injury Prediction, Sports Analytics, Feature Engineering, Multi-Layer Perceptron (MLP)

1. Introduction

In recent years, machine learning has become an important tool for sport analytics. By applying machine learning techniques, researchers could gain valuable insights in player performance and team strategies. These kinds of results could be used to improve training efficiency and team performance. Predicting game results is not only appealing to professionals but also to attract fans. For example, if the predicted win rate is relatively low, the coach can adapt a more conservative strategy. Fans also can use the result to guide their gambling. Based on this, a large portion of literature in this field focused on predicting game results. For instance, Thabtah et al. (2019) applied machine learning methods such as Naïve Bayes and decision trees to analyze NBA game data. Their results found that features like defensive rebounds and three-point percentages were important factors when predicting game outcomes. Similarly, Ouyang et al. (2024) used XGBoost and other machine learning models to predict NBA game outcomes. Notably, they also applied the SHAP algorithm, which can provide interpretable insights. This study points out the potential benefits of machine learning in sports fields.

While predicting game outcomes has its application, predicting player injuries could offer immediate benefits to player and team,which seems more attractive. By using historical player data, teams can understand potential injury risks and even take action before an injury happens. They can adjust training strategies or provide specific medical support for high-risk players. This not only improves players' overall health but also minimizes the economic losses associated with injuries. And several studies have focused on this topic. Wu (2020) used both team-level and individual-level data as features in their analysis. Their result suggests that individual player features, such as average 3-point shot and field goal attempt, were particularly important in predicting injury risks. Other scholars compared several machine learning algorithms, including Random Forest, K-Nearest Neighbors, and Gradient Boosting (Farghaly & Deshpande, 2024). Their results point out random forests have best performance with precision of 92.04%, while Gradient Boosting is slightly behind, with 88.55% precision. Attracting the practical usage of predicting injuries, scholars have designed a specific deep learning model for injury prediction (Cohan et al. 2021). Their model incorporates historical injury and current player data, and achieved high precision and recall rates.

In our study, we aim to predict NBA player injuries by applying machine learning techniques. Our dataset includes player injury information, player physical attributes, and their performance data. The injury data includes the type and method of their injury, while physical attributes provide information like height, weight and age during injury. At first, we use all available performance features, which can retain maximum information in the raw data. Subsequently, we calculate indicators based on rowdata, which includes game style, workload, efficiency, and behavioral tendencies. We plan to compare the performance of the models that use full features and that use summarized features. Because the summarized model is more directly related to the player's style, making the results more interpretable. We cleaned all the datasets to address missing values and ensure consistency. After this, machine learning models such as Random Forest, Logistic Regression, and Gradient Boosting were applied. Finally, we evaluate the performance of different models and identify important factors in our dataset.

2. Methodology

2.1. Data acquisition and Preprocessing

The data we used in the study were collected from several websites. The injury data came from Kaggle, including all recorded injuries in the NBA up to 2023. Each observation contains data of the player's name, team, and description of their injury. Players' physical attributes and performance were collected using web scraping from the NBA official website and basketball-reference.com. The datasets include seasonal statistics of players.

During the data cleaning process, we analyze and process each dataset step by step to ensure that they are accurate and consistent. For the injury data, we started by removing missing values and duplication. After this, we focus on the "notes" feature, which is an unstructured text that cannot be used directly in model fitting. We extract the keywords and categorize them into standard injury methods (Figure1) and locations (Figure2). Injury methods include categories like fractures, inflammation, and sprains.

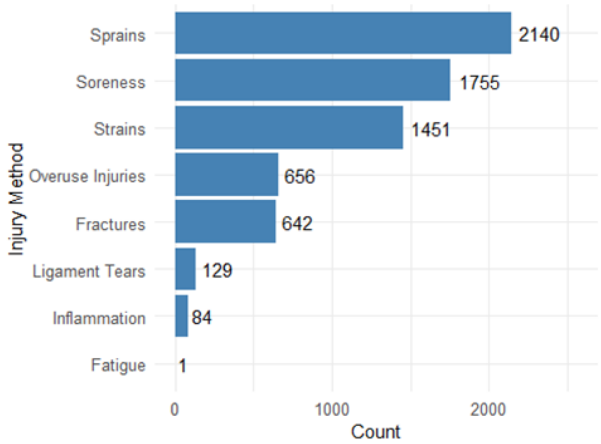
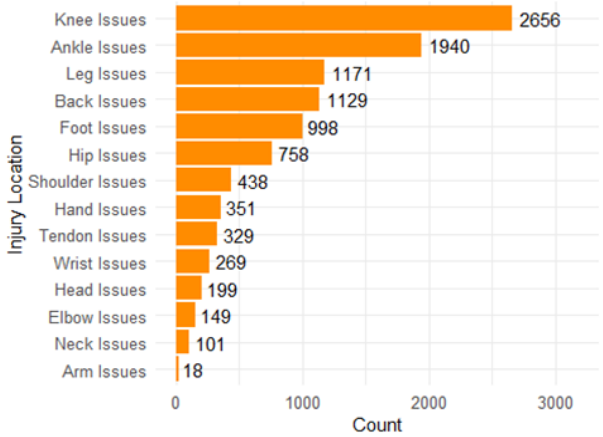


Figure 1: Distribution of Injury methods



Fugure2: Distribution of Injury locations

Similarly, the injury location was identified by searching specific body parts that were mentioned in the notes. For example, a note feature written as “placed on IL with right elbow inflammation”, will be assigned to “elbow issues” for location, and “inflammation” for method. This process helps us turn unstructured data into analyzable structured data.

For player’s physical attributes data, we standardized the player names by converting them to lowercase and removing special characters. We converted heights to centimeters, and weights to kilograms. In addition, we also used the players’ birth date to calculate their age during each injury. What’s more, we found that some players have identical names, leading to potential ambiguity in the data. To address this problem, we calculated whether the players with the same name have overlapped professional careers. For players with overlapping careers, we simply exclude their data to prevent misleading records. For the player’s performance data, we followed similar cleaning steps. After removing missing values and duplicate rows, we standardized player names to ensure consistency.

2.2. Feature Engineering

Since we plan to study the injured location and method separately, we re-split the cleaned data into location and method. Then, we remove records with excessive missing features. For the remaining data with missing features, we applied Nearest Neighbor Interpolation to fill the missing value. Next, we calculated higher-level indicators based on the existing features. These features could reflect the playing style and behaviors. The offensive style is calculated as

PTS+AST+0.5*ORB. While Points (PTS) and assists (AST) could represent a player's direct offensive behavior, offensive rebound (ORB) partially represents players' effort to score. Conversely, defensive style was calculated from steals, blocks and other defensive behaviors. Apart from playing style, we believe that a player's workload is also an important indicator. Therefore, we calculated their total playing time. Finally, we also calculated turnover and foul rate. These variables could provide information about the players' strategic preferences. In order to evaluate the relationship between these higher-level features, we constructed the correlation matrix, as shown in Figure 3.

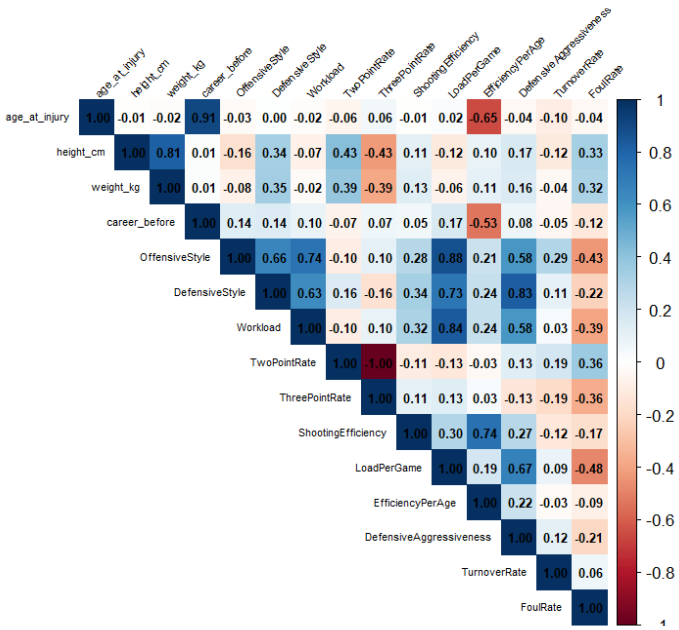


Figure 3: Correlation matrix for higher-level features

In order to identify potential patterns in the data, we used PCA and K-means clustering. We used four clusters in K-means clustering and extracted the first two principal components in PCA. However, our result did not show clear grouping, as there were overlaps between clusters (Figure 4). Based on this, we also tried to use players' positions (Figure 5). Still, we did not observe any obvious grouping pattern in the picture either.

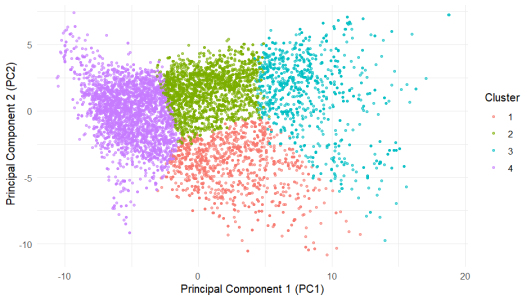


Figure 4: PCA with K-means clustering

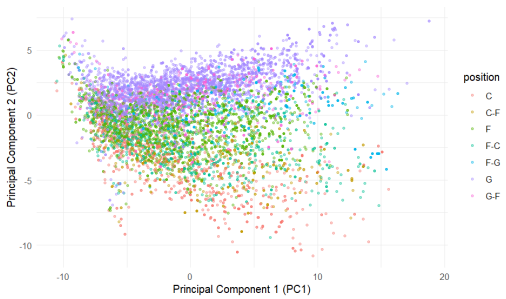


Figure5: PCA with player's position

Considering the complexity and our imbalanced dataset, we decided to use a one-vs-all strategy for our classification. This approach allows us to treat each injury method and location as a separate binary classification problem. To implement this approach, we one-hot encoded injury methods and locations.

3. Modeling

The following models were used: Ridge Regression, Lasso Regression, Random Forest, SVM, and XGBoost. **Logistic regression** serves as a benchmark model for binary classification tasks. It predicts the probability of a binary outcome based on a linear combination of input features. Our datasets contain several binary target variables, such as injury location and method (Binary_Injury_Location, Binary_Injury_Method), which align with the strengths of logistic regression. It allows for interpretable relationships between features like "DefensiveStyle" and likelihood of injury. **Ridge Regression** extends linear models by adding L2-regularization to address multicollinearity and stabilize coefficient estimates. Features like "OffensiveStyle," "Workload," and "EfficiencyPerAge" may exhibit multicollinearity. Ridge regression effectively handles these correlations. It provides a robust framework for predicting injury probabilities while avoiding overfitting in high-dimensional data. **Lasso Regression** introduces L1 -regularization, enabling automatic feature selection while minimizing the impact of irrelevant or redundant features. Variables like "TwoPointRate" and "ThreePointRate" are partially redundant. Lasso regression identifies the most impactful variables, simplifying the model without sacrificing accuracy. Its feature selection capability is critical for large datasets with numerous predictors. **Random forest** is an ensemble learning method that builds multiple decision trees

and aggregates their results for classification or regression tasks. It excels in capturing complex nonlinear relationships. The dataset exhibits nonlinear interactions between features, such as the combined impact of “OffensiveStyle” and “DefensiveStyle” on injury outcome. Random forest provides feature importance rankings, helping identify which variables, such as “DefensiveAggressiveness” or “Turnoverrate”, drive predictions. **BART** is a probabilistic model that combines regression trees with Bayesian inference, allowing for the quantification of uncertainty in predictions. Features such as “ShootingEfficiency” and “EfficiencyPerAge ” involve uncertainty that BART can capture effectively. BART is particularly useful for generating confidence intervals in injury risk predictions, offering insights beyond point estimates. **SVM** is a classification algorithm that constructs hyperplanes to separate data points. It uses kernel functions to handle nonlinear separations. For binary classification tasks with complex boundaries, such as predicting injury risks based on “TurnoverRate” and “DefensiveAggressiveness”, SVM performs well. Its flexibility with kernel functions (e.g., radial basis function) makes it a strong candidate for capturing intricate patterns in the dataset. **XGBoost** is a gradient boosting framework that combines decision trees to deliver high-performance predictions. The dataset’s diverse feature types and high dimensionality benefit from XGBoost’s ability to handle complex interactions and sparse data. Its scalability and efficiency make it suitable for building a predictive system for player injury risks, with added interpretability through feature importance analysis. The **Multi-Layer Perceptron (MLP)** is a feedforward neural network widely used for tasks that involve complex and nonlinear relationships among input features. It consists of an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is fully connected to neurons in the next layer, enabling the model to learn intricate patterns in the data, The MLP was chosen for its ability to handle complex relationships among numerical and categorical features, providing accurate and actionable predictions.

By integrating these models, we aim to balance interpretability, predictive accuracy, and robustness, providing comprehensive insights into the factors influencing player injuries and enabling reliable predictions. We tested all the models individually and found that all of them were very accurate. Due to the diversity of NBA player data and feature engineering, we found

that using a multi-model structure can more accurately predict the injuries that players may encounter in the future. For example, by inputting the player's features into the model, we can get whether the player has a knee injury, and add up the scores of the eight models to get a more accurate prediction.This multi-model approach ensures that our analysis adapts to the varying complexities of the dataset.

4. Result

The average accuracy of the MLP model across all features is comparable to other mainstream models (such as XGBoost and Random Forest), and the performance is stable. The MLP model generally outperforms other models in terms of recall and is suitable for tasks that require identifying more positive examples (such as high-risk injury prediction). After feature selection, the model performance is more focused, and the precision is generally improved, but it may be at the expense of a slightly lower recall.

For most classification tasks, the accuracy metric is between 0.75 and 0.98, showing the high predictive power of the model. XGBoost: Performs well in multiple tasks, with high precision and recall and is particularly suitable for tasks that require a balance between high precision and recall. Also performs well in consistency (Kappa). MLP (implicit in the table): Performs steadily for tasks that require more positive examples (high recall). In some cases, its consistency (Kappa) is slightly lower. Random Forest: It has a high recall and is suitable for tasks that cover more positive examples. Performs averagely in precision and consistency. Logistic Regression, Ridge Regression, and Lasso Regression: Perform well in simple classification tasks, but have limited ability to model complex nonlinear relationships. Some tasks (such as is_Other_Lower_Limb_Issues and is_Knee_Issues) show significant differences in precision and recall between different models. For easier tasks (such as is_Other), all models perform relatively similarly.

5. Discussion

By observing the operation of our model, we found that due to the wide range of features, the relationship between features and dependent variables is not very close. In addition, in the

process of cleaning the data, we found that many players' injuries were not clearly marked as specific injured parts, and some players were resting instead of injured. We believe that in the future, we can extract some features with a relatively high correlation rate with injuries, such as the distance the player runs on the court, the number and frequency of actions that the player makes with high risk of injury (such as making more crossover actions), whether the player has maintained high-intensity and high-consumption games for a long time, and so on.

6. Limitation

According to the data cleaning work we did, the data distribution is uneven. Among the types of injuries, knee injuries are the most common, followed by ankles and legs, but arm injuries and neck injuries are the least common. This will cause the model prediction to be inaccurate. The method we adopted is to summarize some locations with a small number of injuries into one type of injury to ensure data balance.

7. Conclusion

We extracted the injury data and game data of NBA players, created features, and combined multiple machine learning models to create multiple models to predict the probability of player injuries. Our goal is to make this model applicable to the team's coaching team and medical team so that they can better understand the health status of the players and create exclusive training and treatment plans based on the model's prediction results to better prevent player injuries.

References

Cohan, A., Schuster, J., & Fernandez, J. (2021). A deep learning approach to injury forecasting in NBA basketball. *Journal of Sports Analytics, 7*(4), 277–289.

Farghaly, O., & Deshpande, P. (2024). (2024). Leveraging machine learning to predict national basketball association player injuries. Paper presented at the *2024 IEEE International Workshop on Sport, Technology and Research (STAR),* 216–221.

Ouyang, Y., Li, X., Zhou, W., Hong, W., Zheng, W., Qi, F., & Peng, L. (2024). Integration of machine learning XGBoost and SHAP models for NBA game outcome prediction and quantitative analysis methodology. *Plos One, 19*(7), e0307478.

Thabtah, F., Zhang, L., & Abdelhamid, N. (2019). NBA game result prediction using feature analysis and machine learning. *Annals of Data Science, 6*(1), 103–116.

Wu, W. (2020). Injury analysis based on machine learning in NBA data. *Journal of Data Analysis and Information Processing, 8*(4), 295–308.