MLB Injury Prediction Analysis

Colby Reichenbach

2024-05-03

Introduction

Major League Baseball (MLB) is faced with the task of balancing player performance and health with success and financial stability. As a result, predictive algorithms have become increasingly important and popular but have not been resistant to ethical dilemmas. The study done by Karnuta et al. (2020), "Machine Learning Outperforms Regression Analysis to Predict Next-Season Major League Baseball Player Injuries' ', showcases the potential of machine learning (ML) to forecast injuries with high accuracy. This paper will serve as an attempt to validate the findings through the modeling of a novel data set on self-created ML models.

Analysis of Methods

I attempted to recreate the study's results by implementing KNN and LR models in r studio using a novel dataset since the original data was unavailable. This novel data set was comprised of player data from the year 2017 to 2022, encompassing data for 327 pitchers and 653 position players. Data was scraped from a reputable online baseball site, Pro Sports Transactions (https://www.prosportstransactions.com/), using python. Player stats were downloaded from Baseball Savant (https://baseballsavant.mlb.com/). All data set were cleaned and merged into their respective CSV files using python. Pitchers and batters were then run through the models separately to accommodate for their distinct profiles. The methods of both models are described below.

For the KNN model, data was read from the respected CSV file, and the outcome variable `Injured` was converted into a binary factory. The player category was dropped from the data to prevent the model, since it was unique, to prevent overfitting. The data was normalized using centering and scaling. Data was split into training and testing sets with a 75-25 split, and the model was trained using a 10-fold cross-validation, as done in the original study. This aims to reduce overfitting by training and evaluating the model on different subsets of the data. the TuneLength parameter was set to 10, allowing the model to try a range of k values to find the optimal value, which results as k being 19. Results of the model were validated by the accuracy and AUC of the model, which is seen in table 1.

Similarly for the LR model data was read from a CSV file with the outcome variable `Injured` converted to a binary factor. The player category was dropped from the data to prevent the model, since it was unique, to prevent overfitting. The data was normalized using centering and scaling. The data was split into training and testing sets using a 75-25 split. The model was trained using 10-fold cross validation to reduce overfitting and used a binomial family. Results of the model were validated by the accuracy and AUC of the model, which is seen in table 1.

Discussion of Results

The models demonstrated high accuracy and AUC for both batters and pitchers, with the LR model achieving near perfect scores. Though perfect scores seem appealing, it is unrealistic in practice and indicates that the model was overfit to the data. It raises the concern that the model is memorizing the data rather than

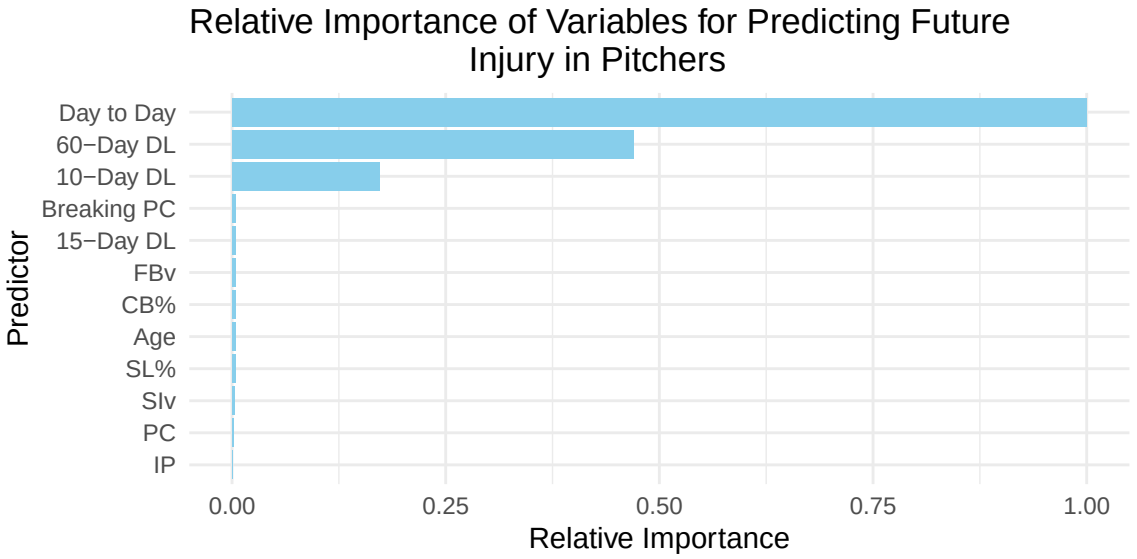
capturing the patterns in the data. Overfitting leads to the lack of generalizability and undermines its predictive power and application to the real-world.

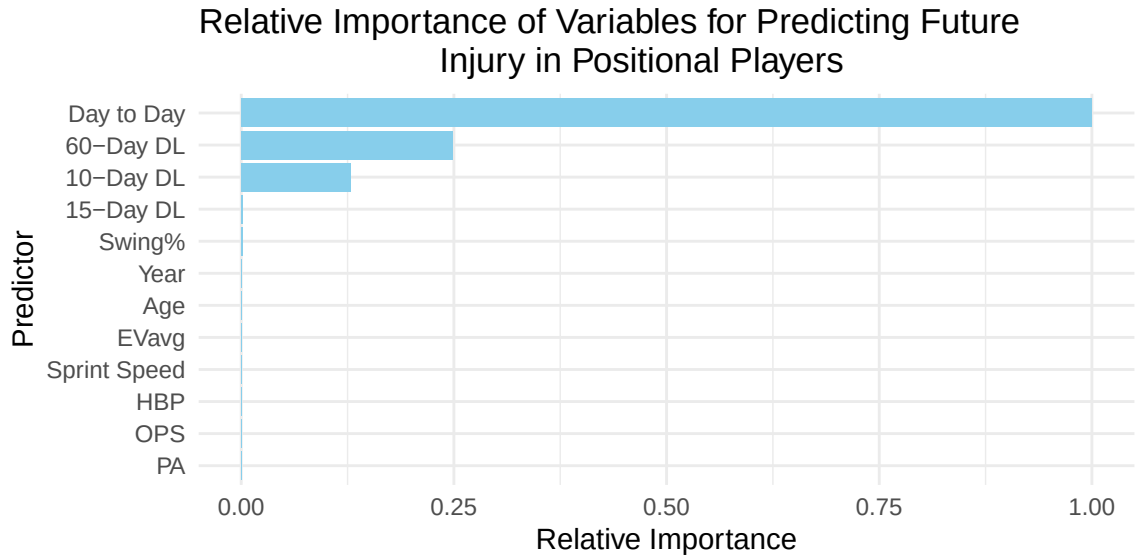
Some reasons that may have led to the model's overfitting the data include the discrepancies in models and data used from the original study. The original models may have been better at understanding the patterns in the data, rather than memorizing the data. The difference in features used to predict the injuries may have played a role in the model's predictions as well. Furthermore, the original data set covered 17 seasons from 2000 to 2017 and encompassing 1931 position players and 1245 pitchers, compared to the novel dataset capturing data from seasons 2017 to 2022, encompassing data for 327 pitchers and 653 position players. The smaller novel dataset may have not been representative of the population, and therefore the model may have learned biases that lead to a near perfect accuracy and AUC.

Table 1: Accuracy and AUC for KNN and Logistic Regression Models

Model	Accuracy	AUC
KNN - Batters	0.84	0.93
Logistic Regression - Batters	0.99	0.99
KNN - Pitchers	0.96	1.00
Logistic Regression - Pitchers	1.00	1.00

To further explore the role of variables in predicting injuries, we see that when looking at the importance of variables in predictions, past injuries signified by `Day to Day`, `10-day DL`, and `60-day DL` were all at the top of the list, shown in the figures below. This suggests that past injuries are the greatest predictor of future injuries. This finding is backed by the original study, that shows a similar graph that has `Injuries` as the variable with the highest importance. However, the original study saw other predictors, such as actual baseball stats, as having a high importance, which we did not see in our results. This suggest that our model was biased towards the past injury designations while making predictions, and defaulted to use those in predictions more than the player stats.





Analysis of Normative Consideration

The original study on injury prediction in baseball showcased promising results, demonstrating a high degree of accuracy. However, my failed attempt to replicate those results has raised important concerns. This experience highlights the necessity for transparency and replicable results in deploying machine learning models. If these models were to be implemented across front offices without proper scrutiny, they could fall into the hands of individuals who are not experts in the field, leading to significant mistakes. For example, issues such as dataset biases or imbalanced classes can lead to a biased model, which could severely impact players' livelihoods. In fields other than sports, such concerns would undoubtedly provoke outrage, but in professional sports, they are often overlooked due to the fascination with statistics and the normalization of these practices.

In the context of consequentalism, the ethical evaluation of actions is based on their outcomes, with the aim of maximizing overall utility. From a consequentialist point of view, the overfitting of the model could lead to harmful outcomes such as making innacurate predictions about injuries in the players. As a result, if these models were used these players could lose out on millions of dollars from their contracts, or they could be unnecessarily benched for injury prevention.

My study highlights these issues through the difficulty of reproducing results. As a result, I would advocate against the deployment of these models in a real-world setting until a model is created that offers adherence to ethical guidelines in place to ensure its reliability and prevent misuse by those who are not experts in the field. The model must be transparent in what variables are being used to predict injuries and be interpretable to those who use it. Furthermore, the model should be able to reproduce results on novel data sets. This ensures that predictions are accurate, and do not lead to harmful consequences that can negatively affect player's livelihoods. Once all of this holds true, I believe that ML models can be beneficial to MLB, and other professional sports, however for the meantime they should be restricted from deployment.

Conclusion

The original study done by Karnuta et al. (2020), "Machine Learning Outperforms Regression Analysis to Predict Next-Season Major League Baseball Player Injuries' ', showcases the potential ML to forecast injuries with high accuracy. Although I was able to create a novel data set and model through the methods described in the original study, I was not able to replicate the results. My study resulted in the model overfitting the data, leading to a near perfect accuracy and AUC score. This highlights the necessity of transparency and

interpretability in the model making process. For me to be confident that these models can be ethically deployed, there needs to be guidelines in place that ensure the model is reliable and prevent its misuse.