MLB Next-Pitch Type Prediction

Alex Kikos

Review the dataset and start outlining the way you would go about the model-building process with the eventual goal of predicting the probability that the next thrown pitch will be a fastball, slider, change-up, etc., in a real-time environment.

I would deploy an ensemble tree to predict the type of pitch thrown in the next sequence. Given that the difference between a correct/incorrect prediction could cost the company (or client) a large amount of money as it relates to sports betting, I would utilize the boosting method to penalize the model for misclassifications. This model could be run for a single pitcher and utilize all his previous throws **or** could be generalized using all the available pitches/data to deploy for any given pitcher in a given scenario. Using an ensemble tree would also account for overfitting of the model that a singular regression/classification tree would have issues with. Additionally, a linear regression model could also be constructed to weight all the predictor variables with the outcome variable being a numerical value for the corresponding pitch type (i.e. fastball = 1, changeup = 2, etc.).

Build and evaluate a model that would be acceptable in a production environment after improvement, with the understanding that delivering predictions with any degree of accuracy is unlikely in this short time span. Additionally, please provide any associated data analysis (plots, graphs, etc..), feature engineering, and code assembled in the form of a python notebook (or similar). Please include markup text to explain your analysis, graphs, etc. If you include a notebook, please also add a PDF version of that notebook, to facilitate review.

Please see the R file included in the zip and Appendix below for boosting-method charts.

Lastly, please provide some details around future steps you would take from a data and technology perspective to finalize this project and the ways you would measure success.

I would look to incorporate any/all relevant datasets available either publicly or available via APIs that would provide further insight into factors that could influence pitch type. After collecting as much data as possible, I would further build upon my boosting tree model to continue to penalize the model for misclassifications since the consequences for failure could result in huge loss of money for the company and/or our clients. Additionally, I would consider different methods of assessing historical data depending on if we are focusing more on *situational* pitching and correlations to pitches that are being thrown that match what is happening live on the field OR rely solely on historical data regarding splits a pitcher has when facing a given batter. I also think utilizing batter-stats such as chase rate, zone-contact, etc. could be leveraged to enhance the historical stats model for a given pitcher vs. batter matchup. Model accuracy could be quantified by a confusion matrix that states the percentage, as well as area under the curve values in terms of sensitivity and specificity values. If a linear regression model proved to be a superior model, I would look to capture as much information as possible in as few variables to increase the speed at which the model could be run/executed in a real-time environment. These models would be compared to one another based on their adjusted R^2 values, with the best model having the highest value. A KNN model could also be created to find the most similar pitch type thrown in a given scenario. The drawback to this is that KNN is computationally expensive and could take a while to run, so it would depend on how much time is allocated for the live odds to be updated and what the team is comfortable with.

Appendix

Figure 1: Boosting method decile-wise lift chart (single pitcher)

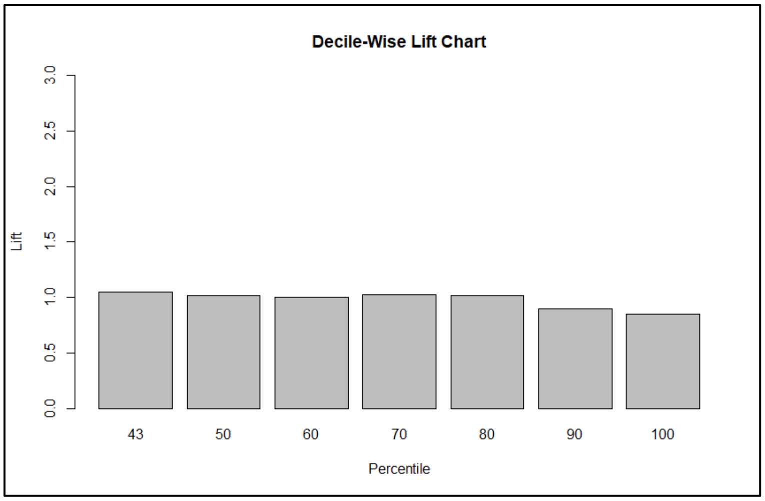
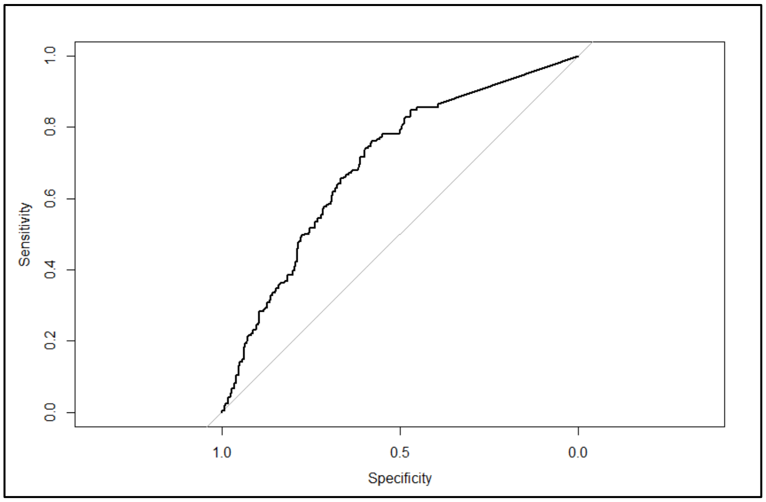


Figure 2: Boosting method ROC curve with an area-under-the-curve (AUC) value of 0.6963 (single pitcher)


Sensitivity
Specificity