

A Machine Learning Approach to MLB Catcher Framing

Nathan Hemenway and Matthew Boyd

12/10/2021

Introduction

- ▶ “Catcher framing is the art of a catcher receiving a pitch in a way that makes it more likely for an umpire to call it a strike – whether that’s turning a borderline ball into a strike, or not losing a strike to a ball due to poor framing.” - MLB.com Glossary

Motivation

- ▶ Baseball catchers can influence the call of a ball or strike on how they catch the ball
- ▶ Some catchers are better than others at this skill
- ▶ Baseball teams are aware of this and are acquiring players good at this skill to win more games
- ▶ We want to quantify the best catcher's at framing for the 2021 season
- ▶ There are several factors that influence whether a pitch will be a strike or ball
- ▶ Catchers getting more strikes translates to more outs, and fewer runs for the opposing team
- ▶ <https://baseballsavant.mlb.com/sporty-videos?playId=273b22a4-f522-4009-87d1-14176772182d>

Data

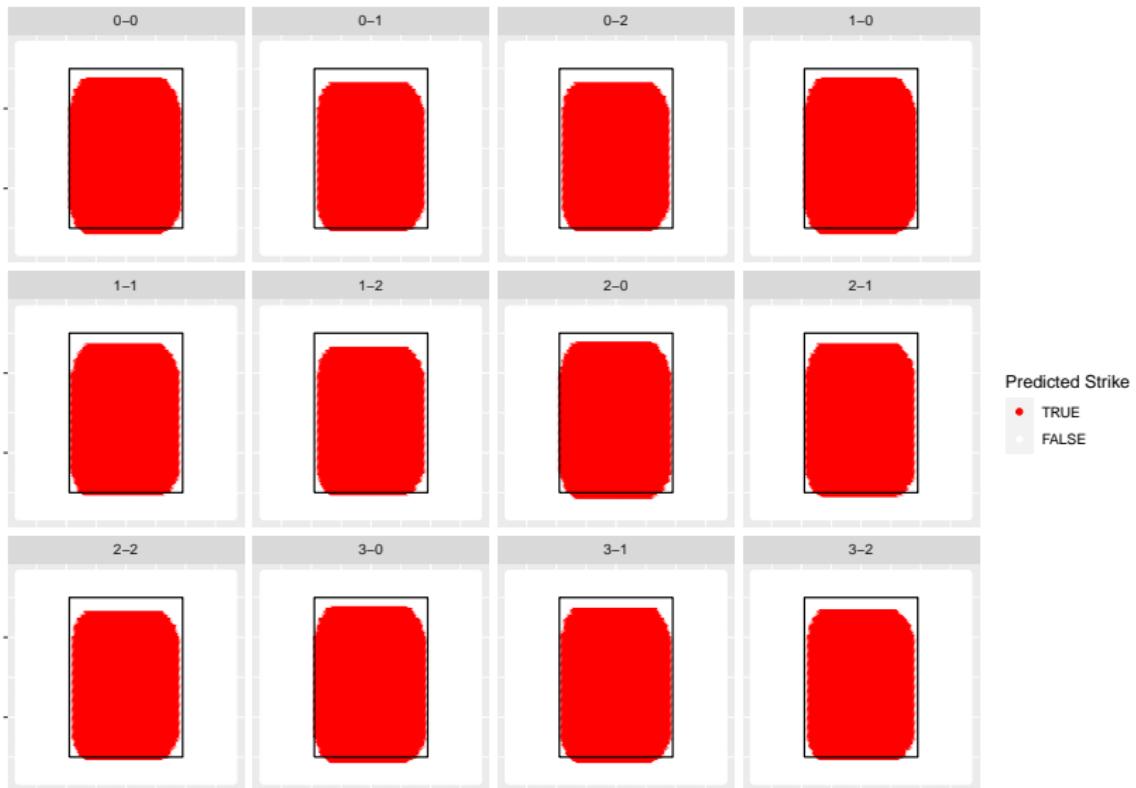
- ▶ 2021 pitch data scraped from Baseball Savant through baseballr package
- ▶ ~700,000 rows (each for a single pitch)
- ▶ Wanted to look at pitches that were not swung at by the batter (called strike or ball)
- ▶ ~350,000 rows remain

Variables

- ▶ We used:
- ▶ Pitch type and pitch release speed, position, and spin rate
- ▶ Whether or not the pitcher and batter are right or left handed
- ▶ Count, number of outs during the at-bat, and inning number
- ▶ Where the pitch landed
- ▶ Whether the game was played home or away
- ▶ How tall the batter is

Example

► Strike Probability by Location and Count



Logistic Regression Model

```
##           true  
## pred      0      1  
##   ball    113034  52774  
##   strike   4858   5280  
  
## [1] 0.672445
```

- ▶ The variables with significant results were:
- ▶ Pitch type, release speed, z release position
- ▶ Whether the batter is a lefty/righty
- ▶ Count
- ▶ Where the pitch landed
- ▶ Most of the innings
- ▶ Batter height
- ▶ Whether it was a home game

Ridge Regression

```
##      true
## pred      0      1
##      0 113910  53037
##      1    4259   4740

## [1] 0.6743546
```

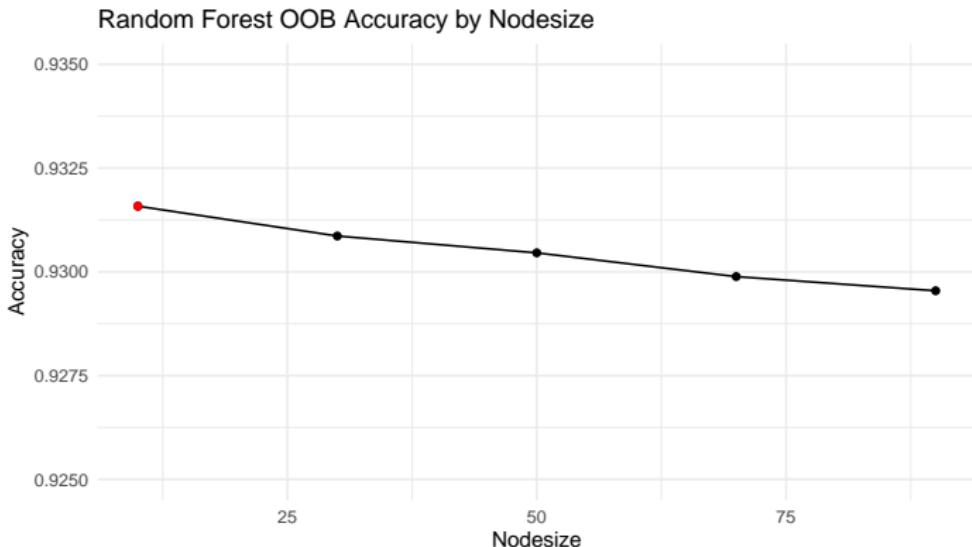
LASSO

```
##      true  
## pred      0      1  
##   0 114841 53933  
##   1    3328  3844  
  
## [1] 0.6745536
```

- ▶ Some of the non-zero coefficients include:
- ▶ Pitch type
- ▶ Count
- ▶ Outs and Inning
- ▶ Batter height
- ▶ Whether the game is home or away
- ▶ Where the pitch landed

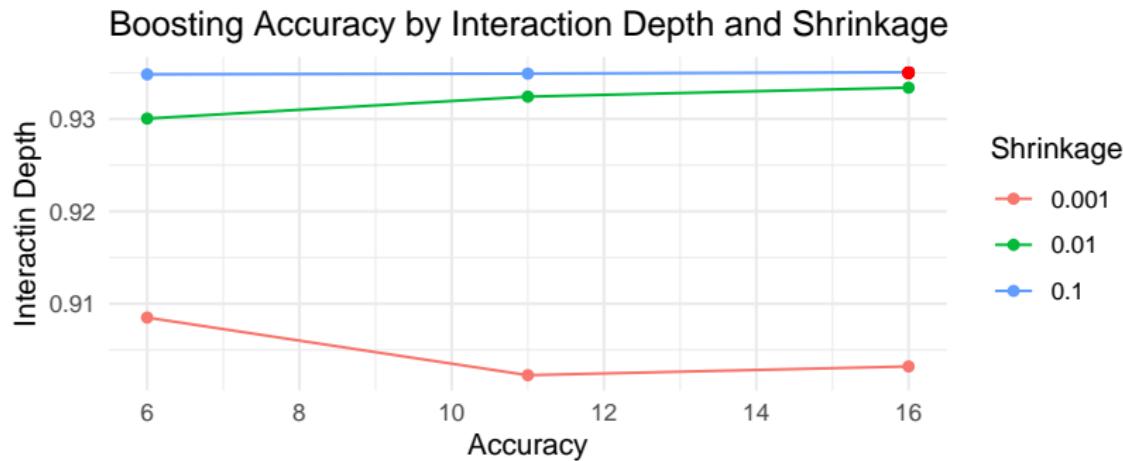
Random Forest

- ▶ Tested several Random Forest models
 - ▶ Nodesize: 10, 30, 50, 70, 90 | # of Trees: 500
 - ▶ Compared Out-Of-Bag Error



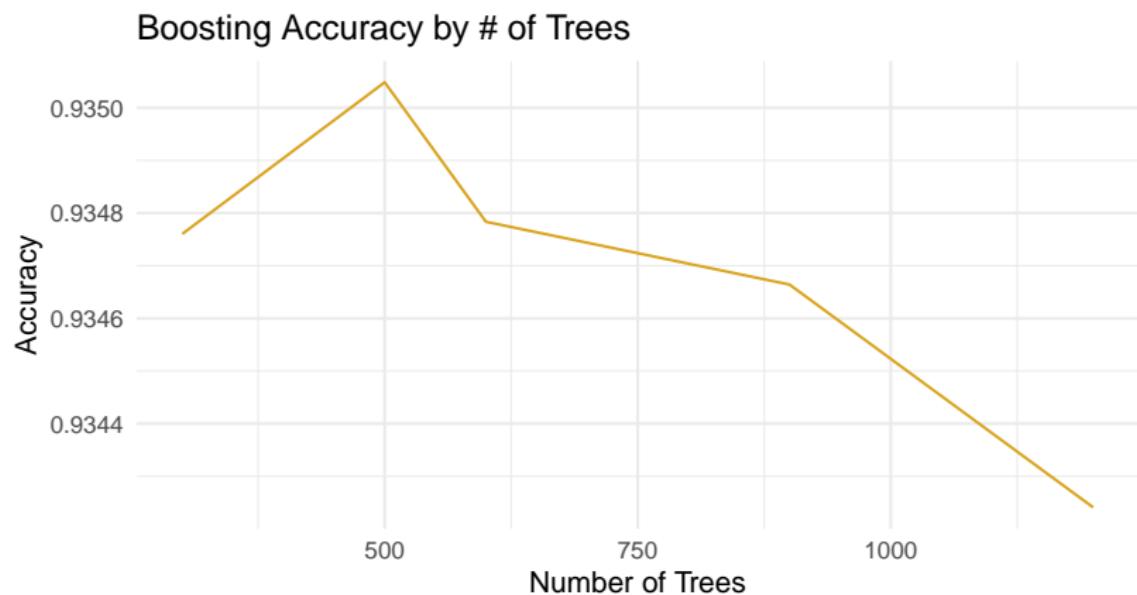
Boosting

- ▶ 3 Fold Cross Validation on Boosting
 - ▶ Interaction depth: 6, 11, 16
 - ▶ Shrinkage: 0.1, 0.01, 0.001
 - ▶ Number of Trees: 500
 - ▶ Compared CV accuracy
- ▶ 27 models took over 5 hours to run



Boosting Part 2

- ▶ Only tuned shrinkage and interaction depth
- ▶ Now tune trees with interaction depth of 16 and shrinkage of 0.1
- ▶ Trees: 300, 500, 600, 900, 1200



Model Comparison

- ▶ Boosting with interaction depth of 16, shrinkage of 0.1, and 500 trees has the highest accuracy

```
##          model  accuracy parameter
## 1  boosting 0.9350484      500
## 2  boosting 0.9347835      600
## 3  boosting 0.9347604      300
## 4  boosting 0.9346642      900
## 5  boosting 0.9342408     1200
## 6        rf 0.9315823       10
## 7        rf 0.9308637       30
## 8        rf 0.9304577       50
## 9        rf 0.9298853       70
## 10       rf 0.9295443       90
```

Results

- ▶ To give credit to each catcher, we use this equation for every pitch:
 - ▶ If strike: $1 - \text{Strike Probability}$
 - ▶ If ball: $\text{Strike Probability} * -1$