

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

Data

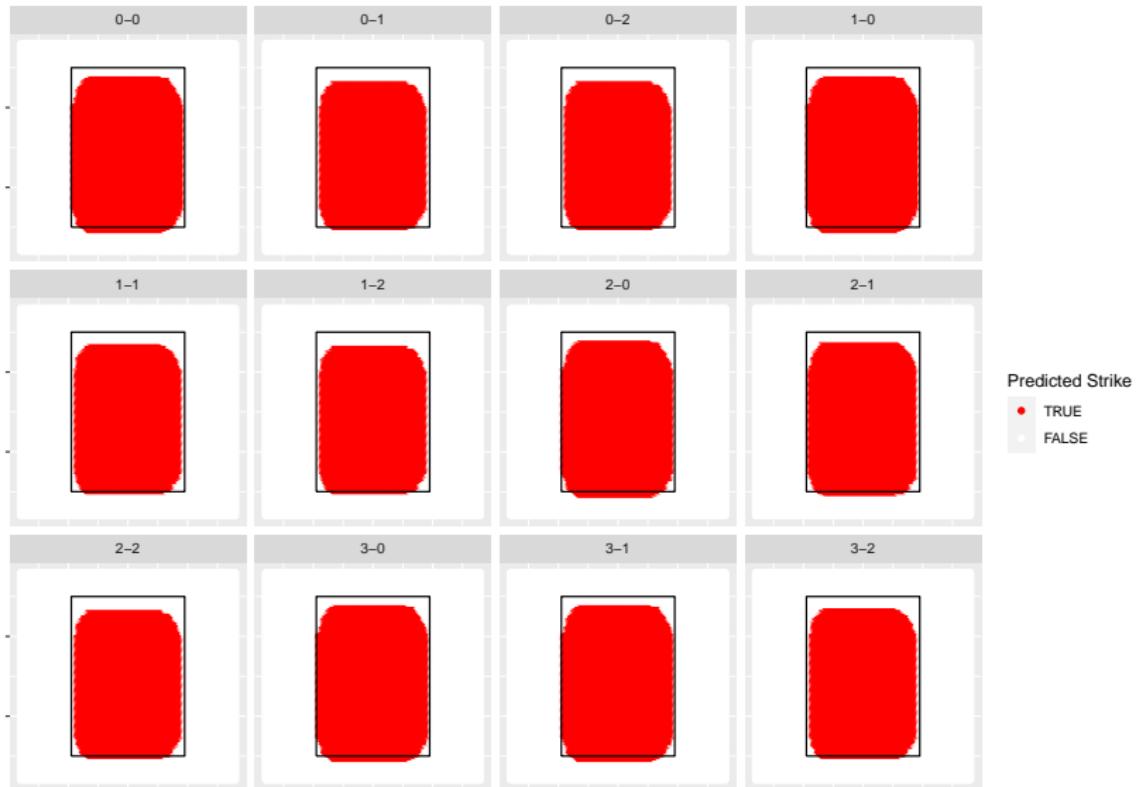
- ▶ 2021 pitch data scraped from Baseball Savant through baseballr package
- ▶ ~700,000 rows (each for a single pitch)
- ▶ Wanted to look at pitches that were not swung at by the batter (called strike or ball)
- ▶ ~350,000 rows remain

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

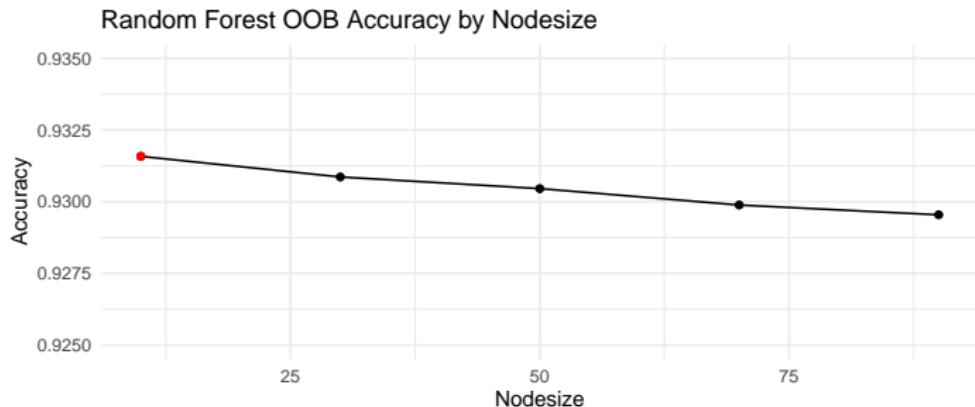


Logistic Regression Model

```
##  
## Call:  
## glm(formula = strike ~ ., family = "binomial", data = data_no_catchers[train,  
##   ])  
##  
## Deviance Residuals:  
##      Min        1Q    Median        3Q       Max  
## -1.6191  -0.9878  -0.6111   1.2276   2.4930  
##  
## Coefficients:  
##                               Estimate Std. Error z value Pr(>|z|)  
## (Intercept)           1.698e+00  6.906e-01  2.459  0.013931 *  
## pitch_typeCurveball  4.183e-01  3.844e-02 10.881 < 2e-16 ***  
## pitch_typeCutter     2.955e-01  3.189e-02  9.266 < 2e-16 ***  
## pitch_typeFastball   4.516e-01  2.885e-02 15.654 < 2e-16 ***  
## pitch_typeSinker     6.161e-01  2.719e-02 22.658 < 2e-16 ***  
## pitch_typeSlider     3.748e-01  2.810e-02 13.338 < 2e-16 ***  
## release_speed        -1.049e-02 1.789e-03 -5.863 4.53e-09 ***  
## release_pos_x         3.145e-03 7.481e-03  0.420 0.674232  
## release_pos_z        -4.714e-02 1.077e-02 -4.378 1.20e-05 ***  
## standR                7.872e-02 1.151e-02  6.840 7.89e-12 ***  
## p_throwsR            -7.410e-03 3.137e-02 -0.236 0.813241  
## count0-1              -1.020e+00 1.844e-02 -55.307 < 2e-16 ***  
## count0-2              -2.027e+00 3.482e-02 -58.230 < 2e-16 ***  
## count1-0              -1.363e-01 1.684e-02 -8.093 5.83e-16 ***  
## count1-1              -7.867e-01 2.004e-02 -39.252 < 2e-16 ***  
## count1-2              -1.857e+00 2.927e-02 -63.427 < 2e-16 ***  
## count2-0              3.026e-02 2.594e-02  1.166 0.243454  
## count2-1              -5.445e-01 2.631e-02 -20.694 < 2e-16 ***  
## count2-2              -1.446e+00 2.971e-02 -48.682 < 2e-16 ***  
## count3-0              7.629e-01 3.751e-02 20.341 < 2e-16 ***  
## count3-1              -2.723e-01 3.654e-02 -7.453 9.09e-14 ***  
## count3-2              -1.019e+00 3.705e-02 -27.503 < 2e-16 ***  
## pfx_x                -5.689e-03 7.340e-03 -0.775 0.438287  
## pfx_z                9.248e-03 1.539e-02  0.601 0.547869
```

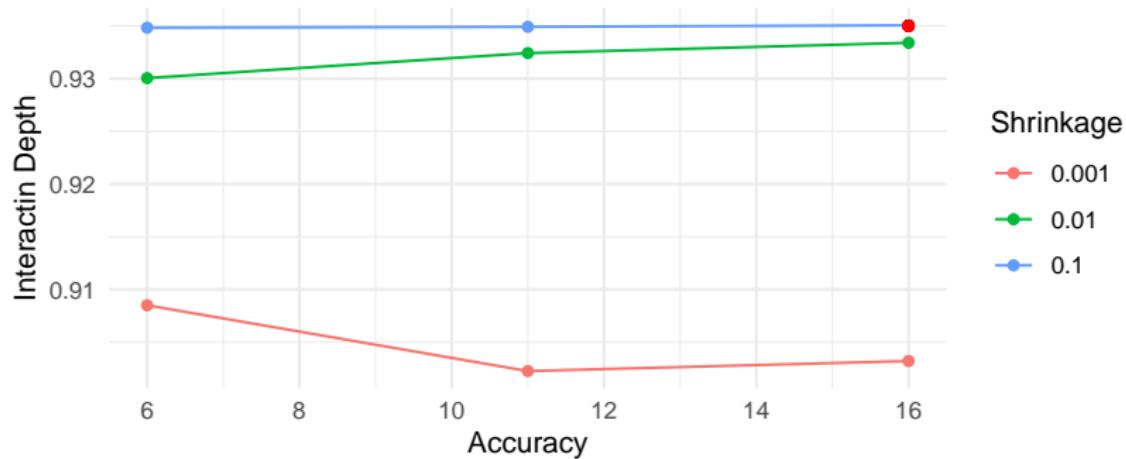
Random Forest

- ▶ 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

