Valuing the Process of a Pitch

Dylan Robbins-Kelley

Problem Description

Baseball is a game of discrete outcomes. Whether at the pitch (ball, strike, or in play), at bat (reach base or record out[s]), or game (win or lose) level. However, the probabilities tied to these outcomes are not static. For example: the probability of a hung curveball resulting in a homerun when thrown to 2001 Barry Bonds is very different when compared to that of one thrown to a National League pitcher who would consider breaking the Mendoza Line a wildly successful season at the plate. Thus, it is fair to question whether an outcome centric approach to player valuation is truly optimal. If a hypothetical pitcher throws the same exact aforementioned hanging curveball twice, should he really get more credit for throwing it to the pitcher who swings and misses, as opposed to Bonds who puts it over the fence? Should we not be evaluating the process, as opposed to the outcome? (i.e., the actual characteristics of the pitch [location, velocity, break, etc.], rather than its hitter dependent result).

In this report we create a NCAA baseball model that predicts the difference between the expected wOBA (xwOBA) of a pitch and the average xwOBA of the count in which the pitch was thrown. These predictions are made independent of batter skill, as to accomplish our goal of measuring the process of a pitch rather than the result. This metric then tells us the expected change in xwOBA that each pitch generates based solely on the characteristics of the pitch, the count, and the handedness of the batter that the pitch is thrown to.

Uses for this sort of metric are rather expansive. The most obvious applications are in aggregation, taking averages at the player or player/pitch level will result in an encompassing player/pitch effectiveness value. However, a few extra steps could lead us to an ERA estimator, work in sequencing, or even a tool for biometric adjustment. These extensions are discussed in detail towards the end of the paper.

Data Description

The model and metrics in this report were created for the California Golden Bears NCAA Division I team. The report utilizes data available from ballparks participating in Trackman's NCAA data sharing agreement. Any player and/or team specific information has been omitted from this report.

The model was created from a dataset consisting of Trackman data across 572,515 unique pitches thrown in NCAA ballparks during the 2019 and early 2020 (pre-cancellation) seasons. This set was trimmed down to 556,408 unique pitches after catcher's interferences, intentional balls, undefined results, and balls in play without exit velocities and launch angles were removed. It is this 556,408-row dataset that will be used throughout the entirety of this paper. For the purposes of this report, we will consider a pitch as having 9 possible outcomes: Ball, Strike (Called or Swinging), Foul Ball, Out or Reach on Error (on Ball in Play), Single, Double, Triple, Home Run, or Hit By Pitch. This information was all present in the raw Trackman data, but some manipulation was done to condense it all into a single outcome column.

Predictions are made on the covariates in *Table 1*, all of which were again present in the Trackman data (with some manipulation necessary). Pitch type is not included as to let the characteristics of a pitch speak for themselves. If one pitcher's cutter behaves exactly like another's slider, are they really different pitches? Descriptions of the variables are presented below:[1]

Table 1

Variable	**Description**
PitcherThrows	Handedness of pitcher
BatterSide	Side of plate batter is hitting from during at bat
Matchup	PitcherThrows & BatterSide (Created to allow model easier recognition of cuts on batter/pitcher matchup. PitcherThrows and BatterSide retained individually to allow for pooling)
Balls	Indication of the number of balls in the count before the pitch is thrown
Strikes	Indication of the number of strikes in the count before the pitch is thrown
Count	Balls & Strikes (Created to allow model easier recognition of cuts on count. Balls and strikes retained individually to allow for pooling)
VertRelAngle	Initial vertical (up-down) direction of the ball when it leaves the pitcher's hand, reported in degrees. A positive number means the ball is released upward, while a negative number means the ball is released downward.
HorzRelAngleFixed	Initial horizontal (left-right) direction of the ball when it leaves the pitcher's hand, reported in degrees. A positive number means the ball is released to the right from the pitcher's perspective, while a negative number means the ball is released to the left from the pitcher's perspective.
SpinRate	How fast the ball is spinning as it leaves the pitcher's hand, reported in the number of times the pitched ball would spin per minute ("revolutions per minute" or "rpm")
SpinAxis	Direction the ball is spinning, reported in degrees of tilt

[1] James Woods (2018). Radar Measurement Glossary of Terms. Trackman. https://trackman.zendesk.com/hc/en-us/articles/115002776647-Radar-Measurement-Glossary-of-Terms

InducedVertBreak	Distance, measured in inches, between where the pitch actually crosses the front of home plate height-wise, and where it would have crossed home plate height-wise if had it traveled in a perfectly straight line from release, but affected by gravity
HorzBreakFixed	Distance, measured in inches, between where the pitch actually crosses the front of home plate side-wise, and where it would have crossed home plate side-wise if had it traveled in a perfectly straight line from release. A positive number means the break away from the batter, while a negative number means the break was toward the batter (Fixed means the sign was adjusted to allow pooling across batter handedness).
PlateLocHeight	The height of the ball relative to home plate, measured in feet, as the ball crosses the front of the plate
PlateLocSideFixed	Distance from the center of the plate to the ball, measured in feet, as it crosses the front of the plate. Negative numbers are outside to a batter, while positive numbers are inside to a batter (Fixed means the sign was adjusted to allow for pooling across batter handedness).
RelHeight	Height, reported in feet, above home plate at which the pitcher releases the ball
RelSideFixed	Distance from the center of the rubber, reported in feet, at which the pitcher releases the ball. Negative numbers are outside to a batter, while positive numbers are inside to a batter (Fixed means the sign was adjusted to allow for pooling across batter handedness).
Extension	The distance, reported in feet, from which the pitcher releases the ball relative to the pitching rubber
EffectiveVelo	Velocity of ball in MPH corrected for how close to home plate the ball is released (Effectively a combination of pure velocity with extension)

Previous work was done for the Cal team in establishing NCAA Division I wOBA weights using data from the 2019 season. These wOBA weights were then used to create an xwOBA model, based on the exit velocity and launch angle of balls in play. A breakdown of the average xwOBA by count can be seen below:

Table 2

Count	Average xwOBA
0-0	.329
0-1	.267
0-2	.187
1-0	.378
1-1	.304
1-2	.215
2-0	.458
2-1	.378
2-2	.270
3-0	.584
3-1	.511
3-2	.396

*Determined from the result of all at bats that reach a given count, not just those that ended at that count

Methods

All analyses are performed in R.[2] Data is scraped using the *RCurl* package.[3] Data manipulation utilizes the *data.table, lubridate,* and *Matrix* packages.[4] [5] [6]

The current model uses both Trackman and situational data as inputs for each pitch (See *Table 1*), and outputs the expected change in xwOBA that each pitch generates. These pitch expectations are naturally centered around 0. Positive values increase the expected xwOBA for the batter and negative values decrease the expected wOBA. The values are count and batter handedness dependent but ignore any other batter specific information. Effectively we are

[2] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL
https://www.R-project.org/

[3] Duncan Temple Lang (2020). RCurl: General Network (HTTP/FTP/...) Client Interface for R. R package version 1.98-1.2.
https://CRAN.R-project.org/package=RCurl

[4] Matt Dowle and Arun Srinivasan (2019). data.table: Extension of `data.frame`. R package version 1.12.8.
https://CRAN.R-project.org/package=data.table

[5] Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL
http://www.jstatsoft.org/v40/i03/

[6] Douglas Bates and Martin Maechler (2019). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.2-18.
https://CRAN.R-project.org/package=Matrix

measuring how each pitch would play in an average situation in that specific count, against a batter of that specific handedness.

Initially we set out to create our model as a sort of multiclass classification problem. We were attempting to predict 9 probabilities for each pitch, one for each of the 9 potential outcomes (see *Data Description*). The form of our end result would have been relatively similar to what you see now. We intended to use the linear combination of these probabilities multiplied by their respective wOBA weights to find the expected change in wOBA created by each pitch. We reasoned that breaking down the model to its probabilistic roots would paint the most complete picture. However, because our dataset is heavily imbalanced, we struggled to find a metric with which to best train our model. Using error rate biased the model toward identifying balls and strikes over everything else, as they dominate the results. More rare events, like homeruns, were effectively ignored. Thus, we pivoted our approach to what you see now, a linear approach. The current model also differs in that we measure xwOBA rather than wOBA in our dependent variable, doubling down on our process-centric methodology.

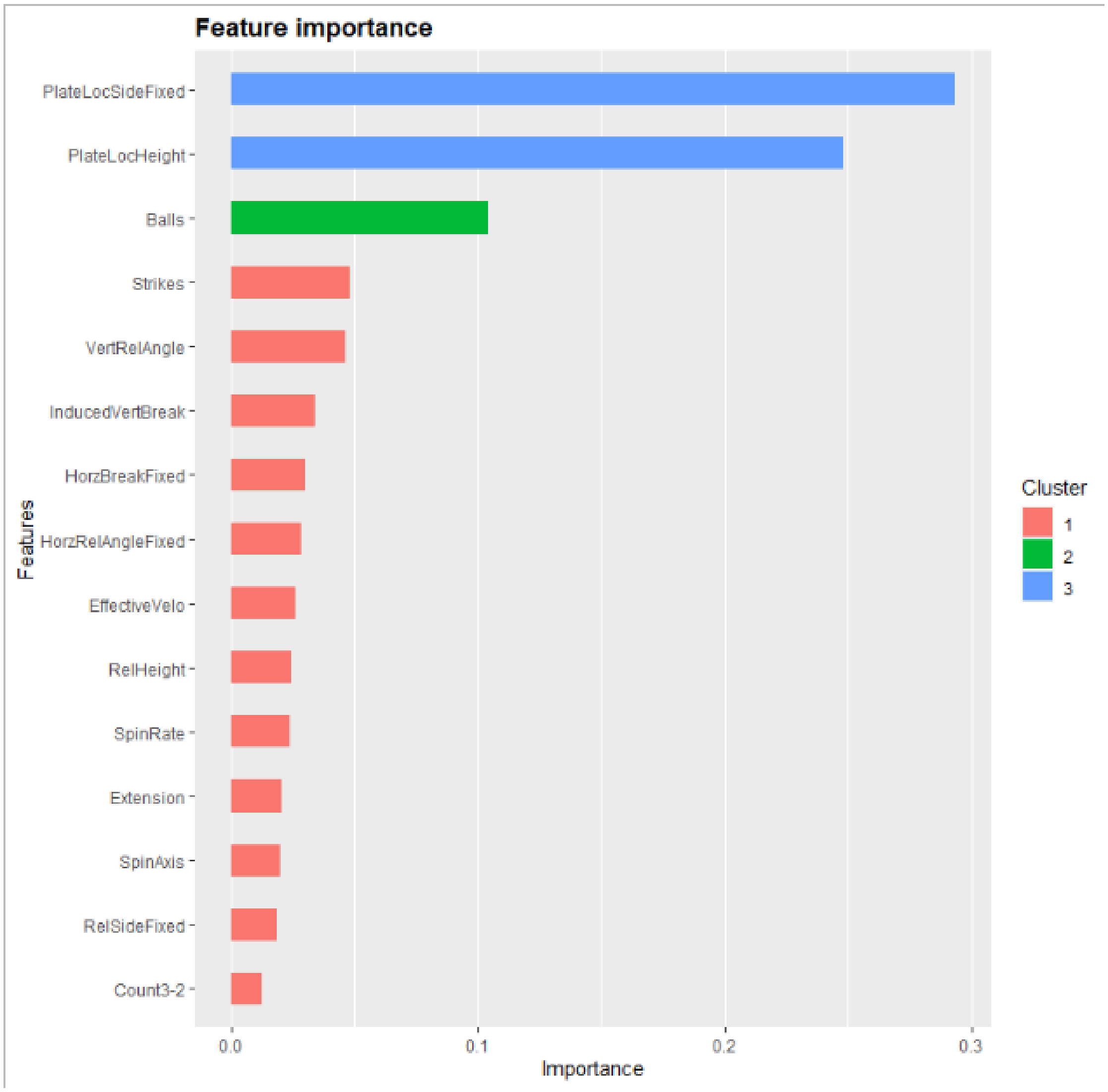
We use gradient boosting via the *xgboost* package in R to create our model.[7] This operates by creating an ensemble of decision trees where each subsequent tree fits to the residuals of the previous trees in the model. The learning rate, tree depth, and number of trees in our model were tuned by conducting a grid search across eta values ranging from .01-.20 (increments of .01) and depths 3-10 (increments of 1), with the number of trees being increased until improvement was not seen for 1000 rounds. Our final model consists of 906 trees of depth 9, with a learning rate of .01. In tuning our model, we opted to minimize the root mean squared error (RMSE). Because RMSE is mathematically sensitive to outliers, our hope is that by using it we will mitigate some of the aforementioned ball and strike dominance, thus allowing balls in play to carry increased weight in our model fit.

[7] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan
Xie, Min Lin, Yifeng Geng and Yutian Li (2020). xgboost: Extreme Gradient Boosting. R package version 1.1.1.1. https://CRAN.R-project.org/package=xgboost

Results

Once our hyper-parameters are tuned, a model is fit on our full dataset. We use this 'full' model to glance at the ordered importance of the features included in our model. This importance metric measures the influence that each covariate has on improving the model's predictive performance, looking at the impact on the dependent variable of the splits the covariate is involved in. The top 15 features by importance can be seen below, produced via the *ggplot2*:[8]

Figure 1



As expected, pitch location is the most important piece of our model as it best delineates between balls and strikes. For the rest of the variables, it is a bit harder to reach any meaningful conclusions. One of the main drawbacks of this sort of importance metric is that it does not tell

[8] H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

us the direction in which these variables push a pitch's expectation. To better evaluate their effects, we will delve deeper into a few of these variables (specifically those that are often said to characterize a pitcher's 'stuff').

To investigate the directional effects of specific variables we hold all other variables constant at their average (can be adjusted to the average for a specific pitch type) and set the handedness and count factor variables to null. We then create two predictions from our model, one with the variable of interest held at its average and one with that variable altered some fixed amount from its average. The resulting difference between these two predictions will give us a general idea of how this covariate impacts the dependent variable. Here we will focus on such results for effective velocity, spin rate, horizontal break and induced vertical break (common measures of 'stuff').

Because different pitch types have different archetypes, we look at 'stuff' dependent on the Trackman auto-tagged pitch type. We use the automatic tags to eliminate human tagging errors, which appear to be much more prevalent at the college level. This is something we shied away from when creating the model; however discrete pitch types are more interpretable when diagnosing results. We choose to focus our attention here on fastballs and sliders, as they are two pitch types with relatively well-defined goals. High velocity, high spin fastballs are considered desirable, as are high spin, large break sliders. There are certainly exceptions to these rules, such as sinkers and frisbee sliders. Nevertheless, these are two of the more well understood pitch types. The tables below outline some of the more interesting finds for both fastballs and sliders (remember that negative values behoove the pitcher):

Table 3

Feature	Pitch Type	Change in Feature	Change in Pitch Expectation
Effective Velocity	Fastball	+1 MPH	-.000063
Effective Velocity	Fastball	+5 MPH	-.010841
Effective Velocity	Fastball	-1 MPH	0
Effective Velocity	Fastball	-5 MPH	-.000001
Spin Rate	Fastball	+100 RPM	-.000742
Spin Rate	Fastball	+200 RPM	-.002449
Spin Rate	Fastball	+300 RPM	-.005491
Spin Rate	Fastball	-100 RPM	+.000236
Spin Rate	Fastball	-200 RPM	+.000236
Spin Rate	Fastball	-300 RPM	+.000572

Table 4

Feature	Pitch Type	Change in Feature	Change in Pitch Expectation
Effective Velocity	Slider	+1 MPH	+.017822
Effective Velocity	Slider	+5 MPH	+.000930
Effective Velocity	Slider	-1 MPH	+.001519
Effective Velocity	Slider	-5 MPH	+.004911
Spin Rate	Slider	+100 RPM	-.003085
Spin Rate	Slider	+200 RPM	-.003137
Spin Rate	Slider	+300 RPM	-.003155
Spin Rate	Slider	-100 RPM	+.000911
Spin Rate	Slider	-200 RPM	+.000911
Spin Rate	Slider	-300 RPM	+.000408
Horz Break Fixed	Slider	+1 in.	0
Horz Break Fixed	Slider	-1 in.	0
Horz Break Fixed	Slider	-3 in.	+.000713
Horz Break Fixed	Slider	-5 in.	+.000713
Induced Vert Break	Slider	+ 1 in.	0
Induced Vert Break	Slider	+3 in.	0
Induced Vert Break	Slider	+ 5 in.	+.004052
Induced Vert Break	Slider	-1 in.	0
Induced Vert Break	Slider	-3 in.	0
Induced Vert Break	Slider	-5 in.	+.000112

* For Horz Break Fixed, negative values mean the ball breaks away from the batter. The average horz. break on sliders in our dataset is -1.45 inches, so we only show changes that stay below 0 (the expected break direction of a slider).

For the most part these directions and magnitudes support our a priori hypotheses about fastballs and sliders. It is important to note that we cannot interpret the changes we see in the same way we would unit increases in a linear model; these are not the ‘true’ values that we are seeing. There are cutoffs and interactions occurring in our model that make such a concrete interpretation relatively tricky. Instead, these results are a rough sketch telling us that our model is behaving in a way that appears to make some ‘baseball sense.’ However, there are some idiosyncrasies. These can possibly be explained by either: A) there being sweet spots for ‘stuff,’ higher is only better to a point or B) these values need to interact with something other than the average of another variable for their effectiveness to be recognized by our model. In *Figure 5*, the numbers for horizontal break on sliders stick out the most. It appears as though sliders that break more result in worse results for the pitcher…which is not intuitive. Our postulation is that, throughout our dataset, these pitches often end up breaking out of the zone. When this happens with an otherwise average pitch, the batter is likely to lay off the pitch resulting in a ball, a negative outcome for the pitcher. This is an example of explanation B), our approach here may be missing the correlation to other positive characteristics that would usually come with this increase in horizontal break. Overall, what we find is encouraging. Our model seems to be approximating what our eyes tell us is happening on the field.

After performing these sanity checks, we now look to measure the overall performance of our model. As we did when tuning our model (see *Methods* section), we will focus on RMSE as our primary performance metric. Because RMSE measures how far our model's predictions are from the actual values, we will want some null models to measure how well we have done. Predicting the result of a pitch is no easy task, even less so when we are throwing out information regarding the skill of the batter. Therefore, we are not aiming for a 'perfect fit,' rather we are looking for improvement upon more naïve models. Our model records an in-sample RMSE of about .1689. A good place to start the comparisons is with the standard deviation of the dependent variable, which is about .1831 When comparing the RMSE of our model to the standard deviation of the dependent variable, we can think of this standard deviation as the performance of a model that always spits out its prediction as the sample mean of our dependent variable. This is the simplest of models, and we see that we have beat it out by about 7.8%. While this may not sound like much, it is a marked improvement given the relative randomness of an individual pitch. For comparison's sake, the following simple linear models that we tested barely budged the RMSE from the standard deviation. None of them dropped the RMSE below .183.

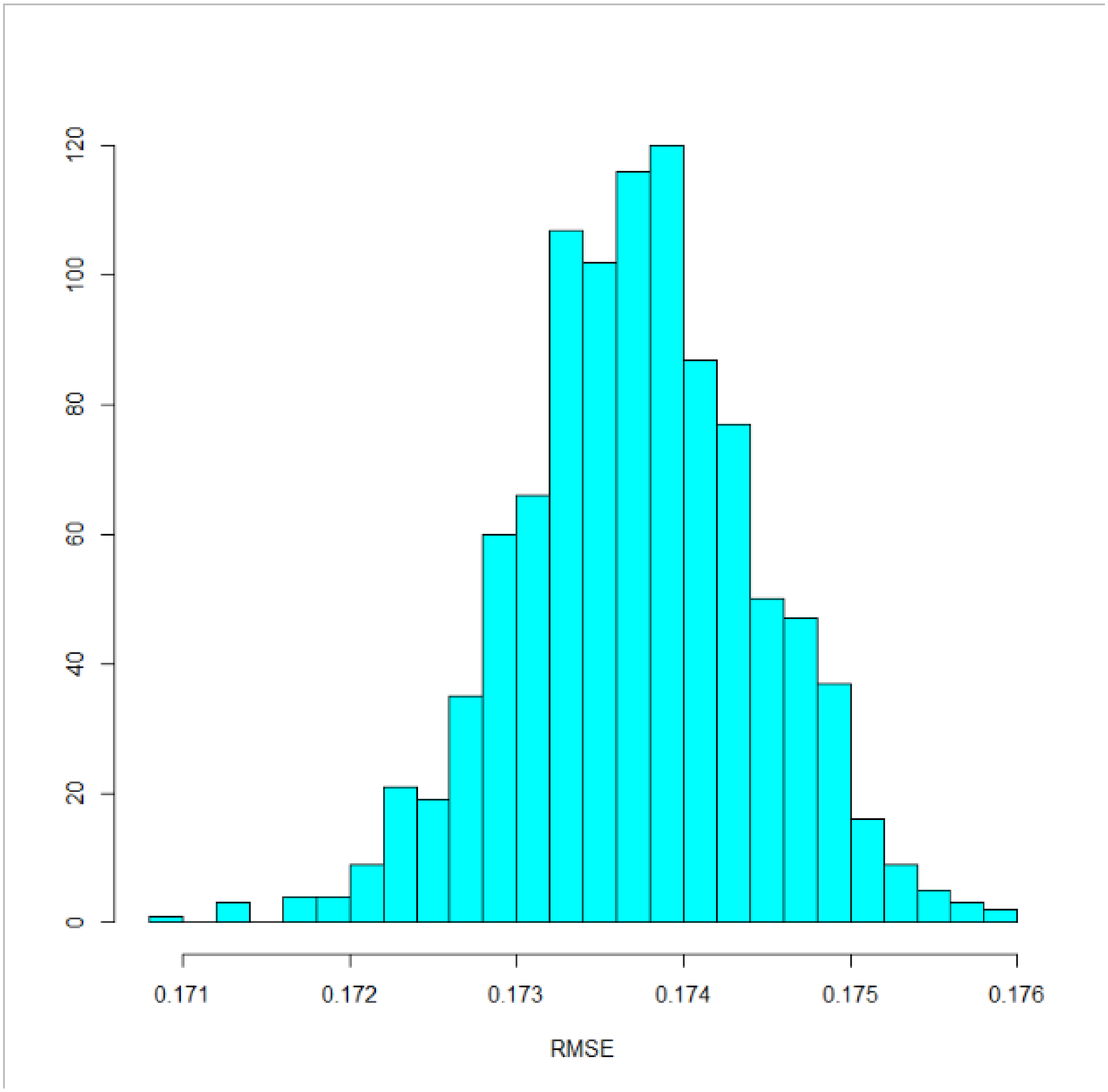
Table 5

Model	RMSE
Standard Deviation	.183099
$Y \sim b_0 + PlateLocHeight + PlateLocSideFixed + PlateLocHeight * PlateLocSideFixed$	.183098
$Y \sim b_0 + SpinRate + SpinRate^2$	.183068
$Y \sim b_0 + EffectiveVelo + HorzBreakFixed + InducedVertBreak + EffectiveVelo * HorzBreakFixed$	.183077
xgboost	.168889

These results are not completely surprising; interactions between the characteristics of a pitch are liable to be both unpredictable and highly non-linear. This is why we employed machine learning for this problem, and why our results hold such potential value. Simply looking at a pitcher's velocity, spin rate, and break is not going to paint a complete picture of that pitcher. The truth lies in how these attributes work together.

Because the RMSEs in *Table 5* are calculated in-sample, it is fair to be concerned that our results may be the product of overfitting. Therefore, we perform 1000 rounds of cross validation to estimate our model's out-of-sample performance, as well as to observe the variability of this performance. In each round, we randomly designate 75% of our data as our training set and leave the remaining 25% as our testing set. We fit a model using our desired parameters on this training set and use this newly created model to predict on the test set. Each round, we record the out-of-sample RMSE that results from comparing our predictions to the actual values seen in the test set. We then move to the next round where we resample new training and test sets, rinsing and repeating this process. In *Figure 2* below, we present a histogram of the RMSEs resulting from this cross-validation, produced via the *MASS* package:[9]

Figure 2



Although we see the RMSE rise from .1689 in-sample to (on average) .1737 out-of-sample, these results are encouraging. The variability observed between these out-of-sample RMSEs shows that our model conclusively outperforms both the standard deviation and our null models. Not

[9] Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

once does our out-of-sample RMSE come particularly close to rising above the RMSEs seen in *Table 5*…and those RMSEs were calculated in-sample. Thus, we can say definitively that our model brings value to the table in terms of predicting the result of a pitch.

Applications and Next Steps

Because college players cannot be traded, this model is mainly useful for finding undervalued pitchers/pitches on one's own staff and/or for valuing college players for pure interest's sake. However, because our model operates on the wOBA scale it is simply a matter of arithmetic to turn our pitch value metric into an ERA estimator. This can be done using the following formula (consider trackmanERA a placeholder for some fun acronym):

$$trackmanERA = LeagueAverageERA + mean\left(\frac{PitchExpectation}{.87}\right) * (9 * PitchesPerInning)$$

In the above formula, .87 is what we have calculated as the Pac-12 wOBA scale factor. Pitches per inning should be calculated at the pitcher-specific level as to credit (or debit) a pitcher for their efficiency.

Before recommending use of this metric over other common ERA estimators, we want to investigate how well current trackmanERA predicts future actual ERA as compared to these other estimators. This is the logical next step for this project, and one that will likely be completed soon. If this metric does in fact hold substantive predictive power, it can be used to identify pitchers who are under/overperforming their 'true talent' ERA.

Another interesting application presents itself when a high school player's Trackman data is available from showcase events held at a team's park. Inputting pitch data from such events into the model will grade these pitches at the college level, unlocking a scouting metric that does not rely on the lower, and highly variable level of competition that high school pitchers face. The model would effectively measure how we expect these pitches to perform when thrown to college hitters. Sample sizes at these showcases will be exceedingly small, but this caveat is true for both our model and for the eyes of recruiters/scouts at these events. However, our model as constructed will not be able to predict growth in these pitchers as a scout might.

Additionally, the model could be adjusted to inform pitch sequencing decisions. Including a previous pitch covariate would allow us to see how pitches play off of one another. Even without changing the model, work can be done to look at how the pitch values of similar pitches change depending on the previous pitches in an at bat. Pitch sequencing is something that a college team can use to improve the pitchers that they already have, which potentially makes this a more useful exercise than using the metric for pure pitch valuation.

One last possible application is as a biometric adjustment tool. Taking predictions at a pitcher's current values and then adjusting them, holding all but one variable constant (as we did in the *Results* sections), could provide insight on tweaks to release point, extension, and even where a pitcher stands on the rubber. Obviously, such mechanical changes would likely have a direct influence on other characteristics of a pitch (velocity, break, spin rate, ability to locate, etc.), but the model could give a nice overview on what might be worth tinkering with. While probably a pipe dream at the college level, theoretically if pitchers and coaches could receive live results from our model during bullpens, they could use the model to workshop both their mechanics, as well as new pitches.

As far as more general model improvements go, there is certainly an argument to be made that we could take things a step further and remove count dependency from our model. Is there really a separate skill for throwing a great slider in an 0-2 count as opposed to an 0-0 count? Or does demonstrating the skill in either count imply the pitcher is just as likely to be able to do it in a different count. There are arguments to be made on both sides, but for our purposes we were not comfortable enough to make this leap in logic…however, that is not to say that it is not worth further consideration. There are certainly many more such refinements that might be made, but as the model stands now it is a useful tool in a college baseball game where analytics are lightyears behind what we see in the big leagues.