Tim Lynch

Kimmie Tran

George Shepherd

Chris Mateer

ECON 590

Dr. Babii

Group 1 Research Paper

11/10/2020

Data Analysis for Predicting Game Outcomes for the Tampa Bay Rays

I. Background Information

Success and victory in games are the ultimate desire, goal, and purpose of every sports team in the United States and beyond. Team managers and coaches are constantly seeking out new tactics that increase their chances of victory by gaining an advantage relative to the opponent. This great desire for success has led to increased employment of data analysis and statistical inference procedures to improve game outcomes across a variety of sports. In particular, baseball, heightened by the sabermetrics movement, has been consistently at the forefront of sports analytics. Analysis in baseball is possible because of data collected from previously played games, based on players' different tactics in addition to modern technologies responsible for collecting advanced data.

This data analysis will be based on the Tampa Bay Rays, an American baseball team based in St. Petersburg, Florida. Although the team has been remarkably successful over the past

few years, the Rays have a long history of limited financial resources, which requires the team to employ innovative strategies to achieve success. Fresh off an appearance in the 2020 World Series, the Rays have established themselves as currently one of best teams in baseball. There is no doubt that the shift of success in their favor results from the Rays' adoption of innovative in-game tactics and player development strategies that have a strong basis in statistical foundations. The players' demographic composition and attributes based on age, ethnicity, and physical characteristics might also have contributed to their improved results.

Using data from the Rays' successful 2020 season, this analysis will seek to predict outcomes on the individual pitch level based on a variety of situational and pitch-tracking variables. The main aim of the analysis is to discern what factors make the Rays' pitchers and hit prevention tactics so successful and attempt to reveal which decisions relating to pitch selection and location most significantly affect the Rays' probabilities of winning games.

II. Research Targets

The primary objectives of this data analysis are:

- To determine which predictors are significant in predicting whether or not a given pitch will result in a ball in play based on a variety of pitch-level factors
- To predict whether a particular pitch will result in a hit or not based on the subset of balls put into play. We will also investigate whether or not predicting on other outcome-related variables, such as exit velocity, will be more useful in terms of devising more effective strategies
- To gain insight into the factors that contribute to the Rays' success and explore whether any of their decision-making on the pitching side of baseball can be improved

III. Data Information

Our dataset consisted of pitch-level data for every single pitch made by the Tampa Bay Rays in the 2020 MLB season. In order to properly investigate whether the individual pitcher had a significant impact on the outcome of the pitch, only Rays pitchers who pitched at least 10 innings throughout the course of the season were included in the final dataset. The pitch-level data includes information about the game situation prior to the pitch in addition to advanced details about the pitch and the outcome of the pitch. For pitches that resulted in a ball put into play, the data includes additional variables, such as exit velocity, launch angle, information about whether the pitch resulted in a hit, etc. The pitch-level data was obtained with the help of the baseball r package in R and the scrape_statcast_savant functions within baseballr, which obtained the pitch-level data from the baseballsavant.mlb.com website. Our dataset contained over 7000 observations, equivalent to the number of pitches thrown by the Rays during the shorented 2020 MLB season. For each pitch, information about the following relevant variables was provided:

- Pitch_name: name of pitch thrown for a given pitch (i.e fastball, slider, changeup, etc.)
- Release_speed: the speed at which the pitcher threw the given pitch
- Release_spin_rate: the spin rate of a given pitch defined as the rate of spin in revolutions per minute after the ball is released
- Release_pos_x & release_pos_z: defined as the horizontal and vertical release points of a given pitch from the catcher's perspective
- Release_extension: the release extension of a given pitch defined as how close a pitcher's release point is to home plate

- Player_name: the name of the Rays pitcher who threw a given pitch
- Zone: indicates the zone location of the ball at the moment it crosses the plate from the catcher's perspective, numbered from 1 to 16
- Events: description of the outcome of the pitch if it marks the end of PA; possible values include single, double, triple, home run, strikeout, field_out, force_out, field_error, etc.
- If_fielding_alignment: indicates fielding alignment of infielders for a given pitch (i.e Standard, Strategic, Infield Shift)
- Of_fielding_alignment: indicates fielding alignment of outfielders for a given pitch
- Launch_speed: refers to the exit velocity or the speed at which the ball was hit for pitches that resulted in a ball in play
- Launch_angle: refers to the vertical angles at which the ball leaves the bat for pitches that resulted in a ball in play

In addition to the advanced metrics described above, the dataset also included several variables that provided information relating to the situation of the game prior to a given pitch. These variables included the handedness of the pitcher, the handedness of the batter, the number of outs at the time of the pitch, the number of balls, the number of strikes, the opponent, and indicator variables indicating whether or not there was a runner on each of the three bases. These situational variables were also used to create additional variables relating to the scoring margin, whether or not there were runners in scoring position, and whether or not the bases were empty to investigate how different situations of a baseball game may impact individual pitch outcomes. More detailed information on all the variables included in the Statcast data can be found at https://baseballsavant.mlb.com/csv-docs. Finally, before the data could be used to create any

models, some additional data cleaning was performed to convert character variables into factor variables and remove any observations with unintended missing values.

IV. Data Overview

Prior to making any predictions, it is important to comprehend how the key variables in our analysis compare to each other, especially for those who may not be familiar with baseball analytics.

Figure 1

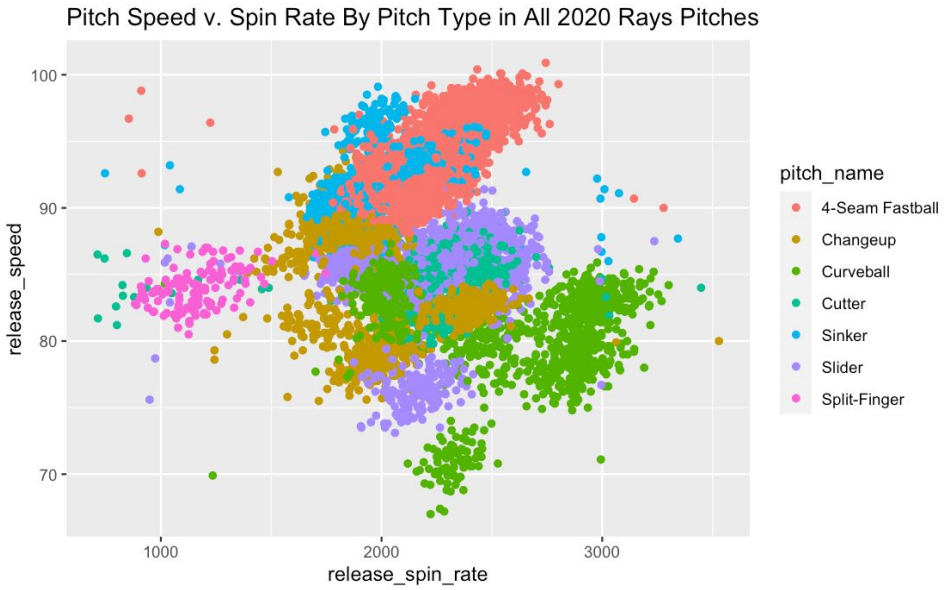


Figure 1 shows how the release speed of a pitch relates to the corresponding spin rate for each different pitch type. As expected, fastballs and sinkers consistently have the fastest velocities, while split-fingers and changeups generally have lower spin rates. Curveballs generally have lower velocities and higher spin rates. The clustering of the different pitch types on the plot

suggests that the interaction between these three variables may be useful to include in a set of possible predictors in our models.

Figure 2

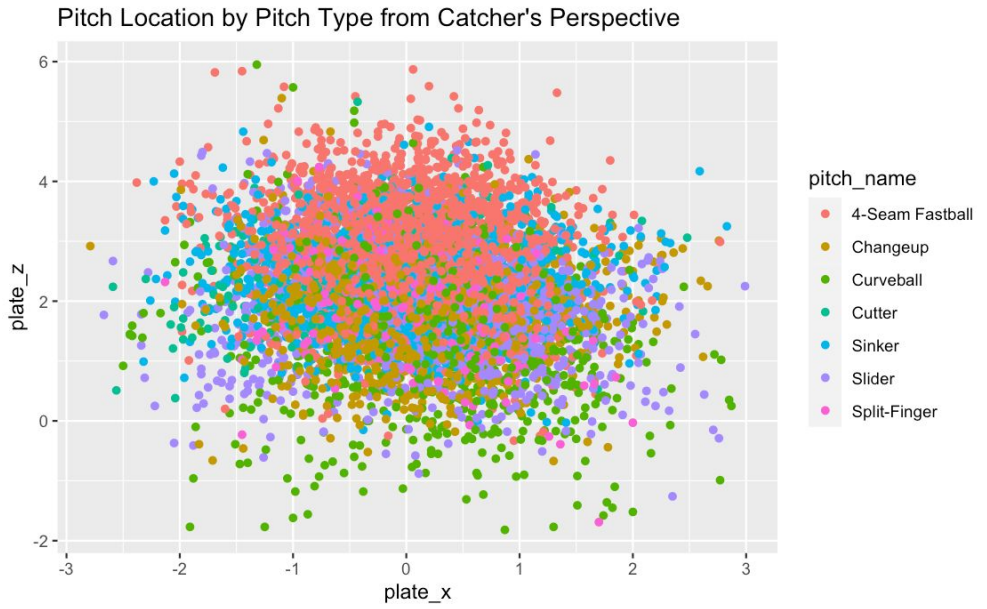


Figure 2 focuses on how the eventual pitch location differs for each different pitch type as seen by the catcher's perspective. While Figure 2 does not have the same evident clustering of pitch types that was seen in the previous figure, it does still reveal insight into how location choice differs for different pitches. Fastballs tend to be located higher in the zone, while many curveballs are located lower in the zone. Since the x and z coordinates of the pitch are extremely precise and cannot be completely controlled by the pitcher, the zone variable as described in the previous section will be primarily used to investigate how the location of the pitch affects the different pitch outcomes as we produce our models.

V. Analysis & Methodology

A. Predicting If A Pitch Will Result in a Ball in Play

To build our prediction of whether a ball will be put in play, we ran multiple kinds of regressions, trying to see if any method had a distinct advantage in accuracy. We used stepwise

logistic regression, linear discriminant analysis, and tree methods, specifically random forests, to try to find the most accurate prediction model. Before any regression was done, we added variables to determine the opponent and whether the Rays were home or away. We also created three variables for whether the bases are empty; if there are runners in scoring position, i.e. runners on second or third base; and whether a ball is put into play. Finally, we removed pitch location outliers such as when a wild pitch or a pitch out occurs.

After we cleaned up the data and added a few variables to help with the regression, we did the stepwise logistic regression. The logistic regression allowed us to make a prediction of a binary variable using the data that we are given. We first created a training and test data set and used these to find what variables we should use include in our logistic regression. From the stepAIC() function, we found the best predictors to use. This function began with all of the variables in the data set plus a few interaction terms in case they would be useful. The function then took away predictors in order to improve the logistic regression. We used the glm() function in R to build the regression, using Release_poz_z, Zone, Balls, Strikes, and Pitch_name, which we had found to be the relevant variables for this regression. At the 5% significance level, Release_poz_z, Zone, Balls, Strikes, Pitch_nameChangeup, Pitch_nameCutter, and Pitch_nameSinker were all statistically significant in predicting whether a ball will be put in play. This model correctly predicted whether a pitch was put into play 82.34% of the time. This high classification rate appears to come from over-predicting that pitches would not be put into play. Of the 3805 pitches in the test set, the logistic regression predicted that 3760 would not be put into play. Of the 649 pitches put into play, it only predicted 11 of them. Despite the classification rate being reasonably high, this regression is not as useful as we would have liked because it overly predicts pitches not being put in play so much.

The next regression we used was linear discriminant analysis, a generalization of Fisher's linear discriminant. For the first LDA regression, we tried using the same variables as the logistic regression. Using the lda() function in R, we found that the LDA regression's classification rate was 82.26%, just below the accuracy of the logistic regression. To improve the accuracy, we tried using a different set of predictors. We used Release_pos_z, Zone, Balls, Strikes, Release_speed, Release_spin_rate, and Stand (the handedness of the batter). These predictors gave a slightly better classification rate, 82.55%, but it heavily over-predicted pitches not being put in play. Of the 3805 pitches, the LDA predicted that 3750 would not be put in play. In reality, 649 pitches were put into play. This model, although it appears to have a good classification rate, potentially over-predicts pitches not being put into play too often for it to be useful. Below is the model output for this LDA model.

```
Call:
lda(in_play ~ release_pos_z + zone + balls + strikes + release_speed +
    release_spin_rate + stand, data = TB2020.inplay.train)

Prior probabilities of groups:
        0         1
0.8302632 0.1697368

Group means:
  release_pos_z     zone     balls   strikes release_speed release_spin_rate    standR
0      5.888561 9.445642 0.8110935 0.8656101      88.56238          2230.364 0.6269414
1      5.705597 7.021705 1.0713178 1.0945736      88.42713          2178.626 0.6325581

Coefficients of linear discriminants:
                             LD1
release_pos_z     -0.3683934861
zone              -0.2013281280
balls              0.2212705785
strikes            0.5418525991
release_speed     -0.0084187661
release_spin_rate -0.0005646958
standR             0.0721021762
```

The third regression technique we used was random forests, a tree based method. Random forest splits the data into random subsets and uses decision trees in each of those subsets to create a regression. It then gives the average prediction of each of the created trees. We used

the randomForest() function in R to construct our regression. Random Trees performed worse than either of the other two regressions. Its classification rate was 81.65% was lower than the logistic and LGA regressions. Like the LGA regression, the random forests over predicted that a pitch would not be put in play. Of the 3805 pitches, random forests predicted that 3682 would not be put in play.

In trying to find a regression that could accurately and usefully predict whether or not a ball would be put in play, we found that the most accurate regressions will resort to predicting that almost all pitches will not be put in play. Although that kind of prediction led to classification rates of roughly 80%, predicting that almost everything will not be put into play cannot help a team to determine how they can reduce the amount of balls the opposing team puts into play. Hitting a major league baseball is sometimes referred to as the hardest thing in all of sports. This seems to be supported by the inability of our regressions to find an accurate useful model, which also reveals the inherent randomness of whether or not a given pitch will be put into play. It appears that the majority of pitches put into play are inexplicable according to our regressions. Because major league pitchers are able to pitch so hard and with so much movement, our regressions seems to suggest that a pitcher will not be able to use a model to reduce balls put in play since most of the time it will be almost by chance.

B. Predicting Whether or Not a Ball in Play Will Result in a Hit

Since we were unable to find a useful regression to determine whether a pitch will be put in play, we decided to see if we could predict whether a ball put in play will result in a hit. We decided to use the same regression methods as we used in the last section. We first filtered the data to find all of the pitches that were put into play against the Rays in 2020. We then created

an indicator variable to indicate whether a ball in play resulted in a hit. After creating test and training sets, we used the same method as in the previous logistic regression to find which predictors to include. The stepwise logistic regression suggested that we use Outs_when_up, Release_extension, and Score_margin. Release_extension, and Score_margin are both statistically significant at the 10% significance. This model did not predict that any ball in play had a probability of being a hit higher than 50%, but its classification rate was 67.54%. We decided to check if lowering the threshold below 50% would help the predictions. At the 45% threshold, it predicted 8 balls in play to be hits, but its classification rate was still 67.54%. At the 40% and 35% thresholds, they predicted 51 and 159 balls to be put in play, but their classification rates were 65.53% and 58.42% respectively. This regression gave so few balls a reasonable chance to be a hit that we cannot accurately and usefully employ this regression in predicting if balls in play will result in a hit, although it did reveal some insight into which predictors are important for determining the likelihood of a hit. Below is the model output for the model produce via stepwise logistic regression predicting a hit.

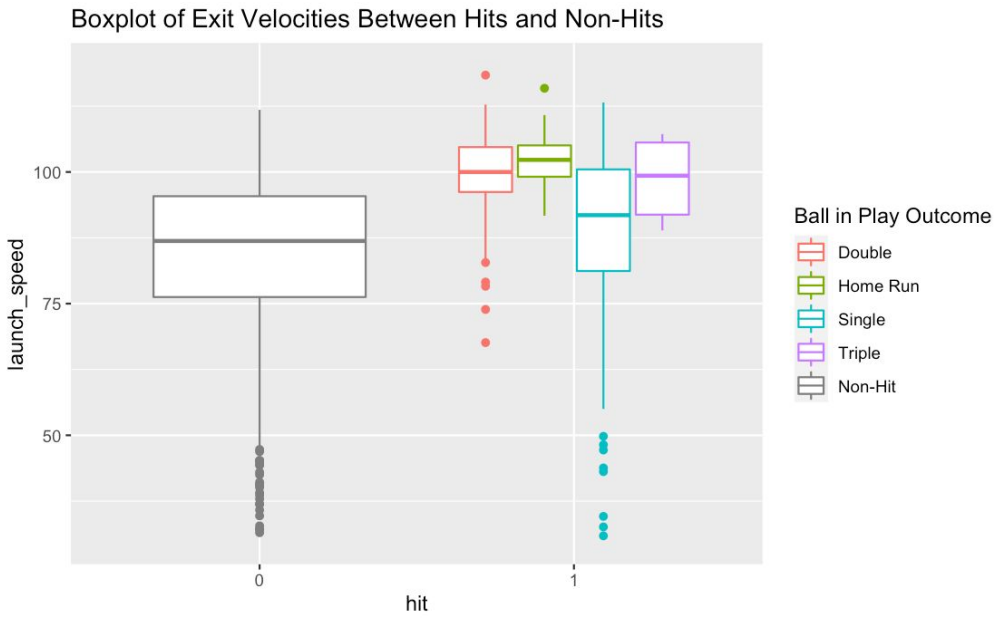
```
Call:
glm(formula = hit ~ outs_when_up + release_extension + score_margin,
    family = binomial, data = TB2020play.train.1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1743  -0.8924  -0.8074   1.4074   1.8012

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -3.22998    1.21259  -2.664  0.00773 **
outs_when_up       0.15895    0.10506   1.513  0.13028
release_extension  0.33607    0.17718   1.897  0.05786 .
score_margin       0.06064    0.03282   1.848  0.06463 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 811.11  on 646  degrees of freedom
Residual deviance: 802.17  on 643  degrees of freedom
AIC: 810.17

Number of Fisher Scoring iterations: 4
```

After the logistic regression, we used the linear discriminant analysis again. For this regression, we used Pitch_name, Release_speed, Release_spin_rate, Outs_when_up, Release_extension, Balls, and Strikes. We predicted whether a ball in play would be a hit, using the same method that we used to predict pitches being put into play using LDA. The LDA regression predicted all but four balls in play would not result in a hit, and the four that it did predict would result in a hit did not. This leaves the classification rate at 66.92%, meaning this regression is both inaccurate and useless. Finally, we tried using the random trees method again. Random trees predicted 77 of the instances would result in a hit, but it was only correct for 18 of those instances. Overall, the random trees method had a classification rate of 61.2%, lower than predicting that all instances would not result in a hit. Because of the seemingly random nature of getting a hit in baseball, we were unable to find a model that could accurately or usefully predict whether a ball put in play would result in a hit.

C. Predicting Exit Velocity

Since our model for attempting to predict whether or not a ball in play will result in hit was relatively unsuccessful due to the inherent randomness that comes with determining hits, our focus shifted to predicting other variables that may be useful for maximizing a given pitcher's abilities to prevent the other team from scoring in a given situation. Of these variables, we determined that exit velocity would be a useful predictor to predict on, since hits have a higher average exit velocity than balls in play that do not result in a hit. Additionally, hits that have higher exit velocities tend to be more consequential on the outcome of the game as seen in the below figure. As seen by the respective boxplots for non-hits, defined as outs, error, sacrifice flies, etc., and singles, there is not too much of a difference in the distribution of exit velocities

for non-hits and singles. However, extra base hits, including doubles, triples, and home runs all have much higher exit velocities on average.



Because balls in play that have higher exit velocities tend to be more likely to impact the game outcome, creating a model predicting exit velocities based on pitch-level factors could be incredibly useful to devising strategies for pitchers to generate weaker contact with lower exit velocities. Since exit velocity is a numerical variable, the model-creation methods considered differed slightly from the previous sections of the paper. Prior to model creation, any observations with exit velocities below 45 mph were removed since many of these outliers likely represented bunts, check swings, or broken bat swings.

First, after splitting the data into the same training and test sets used to predict whether or not a ball in play resulted in a hit, we considered a model produced via stepwise linear regression. The initial model prior to applying the stepAIC function included all possible pitch-level and situational predictors in addition to the interaction effects between a number of

key variables such as ball and strikes, release speed and spin rate, etc. In the final model produced via stepwise regression, 5 predictors were included. These predictors were release_speed, zone, p_throws (defined as the handedness of the pitcher), balls (the number of balls in the at bat at the time of the pitch), and scoring_position (indicator variable of whether or not the hitting team had at least one runner on 2nd or 3rd base). All five predictors were significant at a 5% significance level. When applied to the test data, this model had a mean squared error value of 182.9328, meaning that on average, our model missed the actual exit velocity by about 13.5 mph. Below is the model output for the stepwise regression model.

```
Call:
glm(formula = launch_speed ~ release_speed + zone + p_throws +
    balls + scoring_position, family = gaussian, data = TB2020play.train.2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-45.824   -7.617    1.544    9.535   32.455

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     67.75470    8.07000   8.396 3.11e-16 ***
release_speed    0.26227    0.08858   2.961  0.00319 **
zone            -0.73325    0.14481  -5.063 5.43e-07 ***
p_throwsR        2.65739    1.06916   2.485  0.01320 *
balls            1.28990    0.50883   2.535  0.01149 *
scoring_position -3.51581    1.21682  -2.889  0.00399 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 165.9221)

    Null deviance: 115677  on 629  degrees of freedom
Residual deviance: 103535  on 624  degrees of freedom
AIC: 5016.1

Number of Fisher Scoring iterations: 2
```

In addition to the stepwise regression approach, models with more regularized approaches were also considered to predict exit velocity. Using the cv.glmnet() function in R on the set of all possible predictors of exit velocity plus a few interactions between key variables, a number of models with a range of alpha values from 0 to 1 were created. For each of the 5 cv.glmnet models with different alpha values, we determined the lambda value at which the cross validation error on the test data was minimized. Ultimately, the lasso model with the alpha value

of 1 had the model with the MSE at 184.1375. The lambda value for this lasso model was 0.2595. Since lasso is a shrinkage method, not all of the possible predictors were included in the final model. However, 24 different predictors and dummy variables derived from some of the categorical variables were included in the final model, including release_speed, zone, balls, strikes, and a few indicator variables about the individual pitcher. In addition to having a higher MSE value than the previous model produced via stepwise regression, this regularized model also has issues with overpredicting lower exit velocity values, suggesting it may not be a suitable model to use to predict future exit velocities.

Ultimately, especially with the model produced via stepwise regression, we were able to produce better success for predicting exit velocities than when we previously attempted to predict whether or not a given ball in play will result in a hit. Especially given the fact that extra base hits, which are more consequential on the outcome of the game than singles, tend to result in higher exit velocities, it is in the best interest of teams to implement pitch-by-pitch decision making that would minimize exit velocities. The models described above provide some insight into which situations or pitch decisions correspond to higher exit velocities. For example, hitters tend to produce higher exit velocities when there are more balls in the count, reiterating the importance of being able to throw strikes early in the count. Further research into how pitchers can attempt to minimize exit velocities could be very useful to allow innovative teams like the Rays to yield better results on the field.

VI. Results & Conclusion

Overall, the search for models designed to reliably predict the outcome of a given pitch proved to be quite difficult even after considering a wide range of machine learning techniques.

The process of predicting whether or not a given pitch will result in a ball in play considered models with a cross-validated approach to stepwise logistic regression, linear discriminant analysis, and random forests. Ultimately, the model produced via stepwise logistic regression produced the best classification rate. However, each of the models considered tended to underpredict the number of balls that are actually put into play, which is problematic for a few reasons. The tendency of the model to vastly underpredict the number of pitches put into play means that the predictions of the model are not representative of the actual population of pitches, even though it is more likely that a given pitch will not be put into play. Additionally, by predicting that an overwhelming majority of pitches will not be into play, we cannot draw any discernible conclusions that may improve pitch selection from these models in an effort to find strategies for pitch selection that would limit contact. Looking more closely at the predicted probabilities of a ball in play could produce some insight into how certain pitch selections could influence the probability that a given pitch is put into play.

Then, we limited the scope of the data to only balls that were put into play to try to apply machine learning methods to predict whether or not a ball in play would result in a hit solely based on pitch-level and situational factors. All models considered had similar issues to the models to predict whether or not a given pitch would be put in play as an overwhelming majority of the balls in play were predicted not to be hits. In fact, the model that had the highest classification rate, produced via stepwise logistic regression, actually predicted that every single ball in play would not result in a hit. By essentially predicting that every ball in play will not be a hit, which is not representative of the actual population of balls in play, this model faces many of the same problems that were faced in the previous section. Additionally, the results of this model speak to the inherent randomness that determines whether or not a given ball in play results in a

hit based solely on pitch-level and situational factors because there are so many factors outside a pitcher's control that determine whether or not the ball in play will result in a hit, such as fielders' skill level, fielding positioning, etc. Therefore, we determined that searching for ways for a pitcher to limit exit velocities would be much more effective than searching for ways to limit hits on balls in play because balls in play with higher exit velocities are more likely to result in extra-base hits, which are more damaging to the defensive team's chance of success. Considering a range of models, including stepwise models and regularized models, we found that the stepwise regression models resulted in the lowest MSE and produced insight into which pitch-level and situational factors have a significant impact on resulting exit velocity of a pitch that results in a ball in play.

Overall, this paper seeks to discover how machine learning principles can be applied to the sport of baseball to attempt to predict individual pitch outcomes using only pitch-level factors, such as pitch type, speed, spin rate, and location, as well as situational factors, such as runners on base, the number of outs, etc. Although better success was found with attempting to predict the exit velocity of a pitch that results in a ball in play, these individual pitch-level and situational factors and the machine learning methods outlined in the paper struggle to produce reliable models that can leaned upon to accurately predict the results of a given pitch due to the unbalanced nature of the classes, the inherent randomness of baseball itself, and the notion that the Rays are already successfully pursuing various ways to limit contact and higher exit velocities as much as they can. For future works, it would be interesting to see if this sort of analysis would have more accuracy with teams that are not as analytically-inclined as the Rays. Also, since pitchers are focused on throwing hitters off their rhythm by throwing unexpected pitch types or locations, it would be interesting how the previous sequence of pitches affects the

outcome of a subsequent pitch. This is just an example of how more baseball-centric ideas can applied to improve the preliminary application of machine learning methods to pitch outcomes outlined in this paper.